ABSTRACT

Title of Dissertation: SECURITY ENHANCEMENT AND BIAS MITIGATION FOR EMERGING SENSING AND LEARNING SYSTEMS

Mingliang Chen
Doctor of Philosophy, 2021

Dissertation Directed by: Professor Min Wu
Department of Electrical and Computer Engineering

Artificial intelligence (AI) is being used across various practical tasks in recent years, facilitating many aspects of our daily life. With AI-based sensing and learning systems, we can enjoy the services of automated decision making, computer-assisted medical diagnosis, and health monitoring. Since these algorithms have entered human society and are influencing our daily life, such important issues as intellectual property protection, access control, privacy protection, and fairness/equity, should be considered when we are developing the algorithms, in addition to their successful performance. In this dissertation, we improve the design of emerging AI-based sensing and learning systems from security and fairness perspectives.

The first part is the security protection of deep neural networks (DNN). DNNs are becoming an emerging form of intellectual property for model owners and should be protected from unauthorized access and piracy to encourage healthy business investment and competition. Taking advantage of DNN's intrinsic mechanism, we

propose a novel framework to provide access control to the trained DNNs so that only authorized users can utilize them properly to prevent piracy and illicit usage.

The second part is privacy protection in facial videos. Remote Photoplethysmography (rPPG) can be used to collect a person's physiological signal when his/her face is captured by a video camera, which may raise privacy issues from two aspects. First, individual health conditions may be revealed from a facial recording unintentionally by a person without his/her explicit consent from a facial recording. To avoid the physiological privacy issue, we develop *PulseEdit*, a novel and efficient algorithm that can edit the physiological signals in facial videos without affecting visual appearance to protect the person's physiological signal from disclosure. On the other hand, R&D of rPPG technology also has a potential leakage of identity privacy. We usually require public benchmark facial datasets to develop rPPG algorithms, but facial videos are often very sensitive and have a high leakage risk in identity privacy. We develop an anonymization transform that removes sensitive visual information identifying an individual, but in the meantime, preserves the physiological information for rPPG analysis.

In the last part, we investigate fairness in machine learning inference. Various fairness definitions in prior art were proposed to ensure that decisions guided by the machine learning models are equitable. Unfortunately, the "fair" model trained with these fairness definitions is sensitive to threshold, i.e., the condition of fairness will no longer hold when tuning the decision threshold. To this end, we introduce the notion of threshold-invariant fairness, which enforces equitable performances across different groups independent of the decision threshold.

SECURITY ENHANCEMENT AND BIAS MITIGATION FOR EMERGING SENSING AND LEARNING SYSTEMS

by

Mingliang Chen

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor Min Wu, Chair/Advisor
Professor Dana Dachman-Soled
Professor Furong Huang
Professor Chau-Wai Wong
Professor Michelle Mazurek, Dean's Representative

© Copyright by
Mingliang Chen
2021

To my family.

Acknowledgments

I owe my gratitude to all the people who have made this dissertation possible and because of whom I will cherish my graduate experience forever.

First and foremost, I would like to express my sincere and greatest gratitude to my advisor, Prof. Min Wu, for her continuous support and guidance over the past five years of my graduate study. With her vision, creativity, and enthusiasm, she has taught me how to identify important research problems and how to solve problems with critical thinking and ingenuity. Her encouragement and constructive suggestions have led me through setbacks. Her endless pursuit of excellence inspired me to strive for achieving my potential in my study, research, and professional career. The five-year working relationship will have a lasting impact on my future professional and personal development.

I would like to thank Prof. K.J. Ray Liu, Prof. Richard J. La, and Prof. Prakash Narayan for their enlightening courses to help me establish a solid foundation of core knowledge. I want to thank Prof. Dana Dachman-Soled, Prof. Furong Huang, Prof. Chau-Wai Wong, and Prof. Michelle Mazurek for serving on my dissertation committee and for their valuable comments. It has been my pleasure and privilege to work with my wonderful colleagues in Maryland. I thank my previous and current labmates Dr. Qiang Zhu, Xin Tian, and Zachary Lazri for providing

help and guidance on my research, and I enjoyed our discussions and brainstorms on various topics on research and life. In addition, I would like to thank Dr. Quanzeng Wang from Food and Drug Administration, Dr. Hui Su from Google, and Dr. Qi Zeng from Facebook, who were my mentors during my internships. Their guidance and support gave me wonderful and memorable internship experiences along with my Ph.D career.

Last but not least, I would like to thank my parents who always give me unconditional love and support. My gratitude and love to them are beyond words. I dedicate this dissertation to them.

Table of Contents

Dedication ii

Acknowledgements iii

Table of Contents v

List of Tables viii

List of Figures ix

Chapter 1: Introduction 1
- 1.1 Intellectual Property Protection of Neural Network Models 3
- 1.2 Privacy Protection in Facial Videos 4
- 1.3 Fairness in Machine Learning Models 7
- 1.4 Dissertation Organization 8

Chapter 2: Intellectual Property Protection of Deep Neural Networks 10
- 2.1 Overview 10
- 2.2 Related Work 13
  - 2.2.1 Adversarial Examples 13
  - 2.2.2 IP protection of DNN models 14
- 2.3 Access-control Design 16
  - 2.3.1 Threat Modeling 18
  - 2.3.2 Training Formulation 19
  - 2.3.3 Protective Transform 21
- 2.4 Experimental Results 25
  - 2.4.1 Network Structures and Training Settings 25
  - 2.4.2 Performance of Access-control Design 27
  - 2.4.3 Visualization of Image Inputs 31
  - 2.4.4 Impact of Perturbation Budget 34
  - 2.4.5 Resilience Analysis against Attacks 37
  - 2.4.6 Ablation Study on Training Formulation 41
- 2.5 Discussions 43
  - 2.5.1 Relation Between the Ordinary and the Restrictive DNN . . . 44
  - 2.5.2 Privacy-aware Smart Camera Application 45
- 2.6 Chapter Summary 47

Chapter 3: Physiological Privacy Protection Against rPPG Technology 49
3.1 Overview . . . 49
3.2 Related Work . . . 52
3.2.1 rPPG Technology . . . 52
3.2.2 Biometric Privacy Protection . . . 53
3.2.3 rPPG Feature in Visual Security Tasks . . . 55
3.3 Conceal rPPG Signals in Facial Videos . . . 56
3.3.1 rPPG Extraction . . . 56
3.3.2 rPPG Editing . . . 58
3.3.3 Skin Pixel Adjustment . . . 60
3.4 Experimental Results . . . 61
3.4.1 Performance on PURE dataset . . . 62
3.4.2 rPPG Analysis on Multiple Facial Subregions . . . 68
3.4.3 PulseEdit Against Forensic Analysis . . . 70
3.4.4 Performance Summary of PulseEdit . . . 74
3.4.5 Comparison with Prior Art . . . 75
3.5 Analysis of Adversarial Scenarios . . . 78
3.5.1 Analysis Against rPPG-based Liveness Detection . . . 78
3.5.2 Analysis Against rPPG-based Deepfake Detection . . . 79
3.6 Discussion . . . 81
3.7 Chapter Summary . . . 83

Chapter 4: Identity Privacy Protection in rPPG Technology 85
4.1 Overview . . . 85
4.2 Related Work . . . 87
4.2.1 Public Datasets for Remote HR Estimation . . . 87
4.2.2 Identity Privacy Protection for Facial Images . . . 88
4.3 Anonymization Transform for Privacy Protection . . . 90
4.3.1 Standardization to a Generic Face Template . . . 92
4.3.2 Scale Space Decomposition . . . 93
4.4 Experimental Results . . . 95
4.4.1 Analysis of Physiological Signal in DoG Layers . . . 95
4.4.2 Performance of Anonymization Transform . . . 97
4.4.3 Facial Feature Visualization . . . 99
4.4.4 Recognizability of Privacy-Protecting Faces . . . 100
4.5 Chapter Summary . . . 102

Chapter 5: Distribution Parity for Group Fairness 105
5.1 Overview . . . 105
5.2 Background . . . 108
5.2.1 Classification Parity . . . 108
5.2.2 Limitations . . . 109
5.3 Distribution Parity for Threshold Invariant Fairness . . . 111
5.3.1 Notations . . . 111
5.3.2 Fairness Definition . . . 111

5.4 Equalization of Risk Distribution 115
5.4.1 Approach 1: Histogram Approximation (HA) 115
5.4.2 Approach 2: Gaussian Assumption (GA) 117
5.4.3 Classifier Formulation 118
5.5 Experimental Results 120
5.5.1 Dataset and Feature Processing 122
5.5.2 Tradeoff Between Accuracy and Fairness 123
5.5.3 Visualization of Risk Distribution 124
5.5.4 Parity Variation Versus Threshold 127
5.5.5 Discussion 129
5.6 Chapter Summary 131

Chapter 6: Conclusion and Future Perspectives 133

Bibliography 139

List of Tables

2.1 Structure of stacked CNN used in the experiments 26
2.2 Structure of Resnet-56 used in the experiments 26
2.3 Performance of the proposed framework on unauthorized and authorized access 30
2.4 Visual similarity between the image inputs in two proposed protective transform modules 33
2.5 The resilience of the restrictive DNN against various attacks 38
2.6 Estimated error of random patterns for the three attacks 41
2.7 Four modes of training formulation in ablation study 41

3.1 Result of PulseEdit with $\lambda = 0.5$ on HR error, perceptual distortion, and forensic analysis 73
3.2 Performance comparison of rPPG removal methods 77
3.3 Impact of PulseEdit on performance of rPPG-based visual security algorithms 80

4.1 Summary of the public benchmark datasets for rPPG analysis 88

5.1 Preprocessed Features Used in COMPAS Data 123
5.2 Performance Under DP Constraint 128
5.3 Performance Under EO Constraint 129

List of Figures

1.1 Three social issues in AI-based sensing and learning systems studied in the dissertation. 2

2.1 Illustration of adversarial examples to a DNN by perturbing image pixels in a negligible amount. 13
2.2 The proposed framework for access control protection. 17
2.3 The structures of two proposed protective transform modules. 22
2.4 The structure of RandNN used in the experiments. 27
2.5 Performance of the restrictive DNN with different β. 28
2.6 Examples of the raw input and the authorized input for the restrictive DNNs. 32
2.7 Examples of two perturbation modes with different perturbation budget s from three datasets. 34
2.8 Performance of the restrictive DNN with different s. 36
2.9 Ablation study on attack resistance of the restrictive DNN. 42
2.10 Illustration of the relationship between the ordinary DNN and the restrictive DNN. 45
2.11 Illustration of the privacy-aware smart camera application. 46

3.1 Illustration of the proposed PulseEdit. 50
3.2 Pipeline of PulseEdit system. 57
3.3 Exemplary face crop from the videos and spectrograms of the rPPG extracted from the uncompressed videos. 63
3.4 Exemplary face crop from the videos and spectrograms of the rPPG extracted from the lossy compressed videos. 64
3.5 Performance of PulseEdit on the PURE dataset 66
3.6 Illustration of the three facial subregions. 67
3.7 HR estimation of PulseEdit videos on multiple face subregions. 69
3.8 Performance of PulseEdit against forensic analysis in MPEG-4 videos. 72
3.9 Performance comparison between PulseEdit and Chen's method . . . 76
3.10 ROC curves of rPPG-based vision security algorithms 81

4.1 Exemplary frames in several public benchmark datasets for rPPG analysis. 89
4.2 Pipeline of anonymization transform to protect identity privacy in rPPG data collection. 91

4.3 Generic face template. 92
4.4 Spectral similarity of the physiological signal between the DoG layers and the original facial videos. 96
4.5 Identity-protecting faces with different privacy strength levels. 97
4.6 Correlation of the rPPG signals extracted from the original videos and the processed videos. 98
4.7 Visualization of facial feature embedding in different levels of reconstructed face. (a) Only D step, (b) S+D steps. 101
4.8 Recognition accuracy of identity-protecting faces. 103

5.1 Working pipeline of machine learning models. 108
5.2 Hypothetical risk distributions of two groups. 110
5.3 Tradeoff between model accuracy and classification parity index. . . . 121
5.4 Risk distributions of LR classifiers with different levels of DP fairness regularizers. 125
5.5 Risk distributions of LR classifiers with different levels of EO fairness regularizers. 126
5.6 Variations of classification parity versus decision threshold. 130

Chapter 1

Introduction

With the development and popularity of Artificial intelligence (AI)-based sensing and learning systems over in recent years, we have witnessed their success in different applications, such as object recognition, health monitoring, and automatic decision making. These systems are influencing many aspects of our daily life. Many companies have deployed sensing and learning systems as backends to provide services to the customers or embedded them into commercial products. For example, we can also monitor our health status with a portable cellphone camera, facilitating physiological monitor during daily exercises or patient care. Banks turn to machine learning-based decision-making systems to evaluate the prospective customers' credits and help decide the approval of loan applications.

In the new AI era following the revolution of the Internet, the emerging AI-based sensing and learning systems take an active part in our social activities. Every day, large quantities of data are being uploaded and shared via the Internet, and these systems are processing them automatically for the customers' needs. The communication and share of data in social activities may raise privacy and security concerns. News media has reported several incidents about information security

leakage using these sensing and learning systems. Also, people have found that many off-the-shelf learning systems produce concerning outcomes or achieve different performances when processing data across the groups of the population of interest, such as race and gender. Therefore, it is necessary to provide solutions to address the security and fairness concerns of these systems beyond the commonly evaluated learning performance.

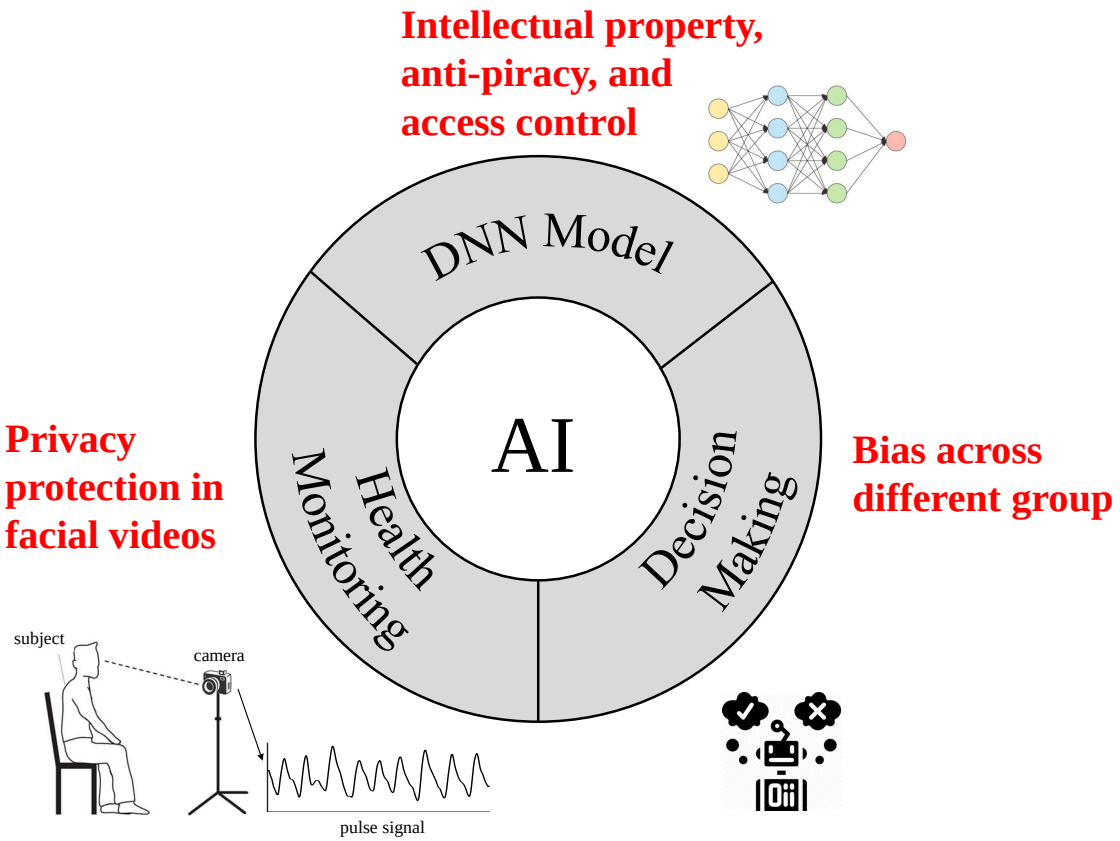


Figure 1.1: Three social issues in AI-based sensing and learning systems studied in the dissertation.

In this dissertation, we study three kinds of AI-based sensing and learning systems from security and fairness perspectives, making them better tools for usage in human society, shown in Figure 1.1.

1.1 Intellectual Property Protection of Neural Network Models

Deep neural networks (DNNs) have gained a growing amount of attention due to the significant performance leap in many learning problems ranging from computer vision to information forensics and biomedical analytics [1–8]. For instance, many companies have deployed well-trained DNN models as backends to provide service to customers. Building a DNN model with good performance is generally a substantial task, usually requiring massive human-labeled training data, powerful computing hardware, and researchers' skills and efforts. Although several DNN models [2, 9–11] are open to the public with permission of non-commercial usage, many model owners in commercial applications expect to keep the trained DNN models private due to business considerations and security issues. Hence, the DNN models are becoming an emerging form of valuable intellectual property (IP) for the model owners, similar to the digital media in the late 1990s. In order to encourage healthy business investment and competition, the IP protection of these trained DNNs in commercial use is becoming increasingly important.

Similar to watermarking in digital media, the prior art of IP protection for DNNs tries to embed specific verification information into the model during model training. The ownership of the DNN model can be identified in a post-piracy investigation by extracting the verification information from the model and comparing it with the owner's. Nevertheless, they do not actively address the problem of unauthorized access and piracy/theft. For example, an adversary may clone the whole DNN model and make a fortune from it.

Unlike the prior art, ensuring access control is an essential aspect of IP protection, which can actively prohibit unauthorized access and piracy/theft. Inspired by DNN's intrinsic property – adversarial examples – we propose to employ this adversarial behavior to trigger the access control functionality, such that the DNN model works properly to the authorized access while it is dysfunctional to the unauthorized access or illicit use. With our access-control design, only authorized users can access the trained model and benefit from its high learning capabilities. In addition to piracy prevention, proper access control also supports privacy protection against the misuse of DNNs that produce decision results of a private or sensitive nature, especially in biomedical analytics [6–8].

To make our access-control design robust for practical use, we establish the threat model and analyze adversarial attacks. We consider threat modeling in the access-control design and enhance the DNN model's resistance to various attacks and unauthorized access. For a proof-of-concept demonstration, we also introduce a potential application of the access-control design in the camera system, where only pictures taken from the authorized cameras can obtain valid outputs from the corresponding DNN model.

1.2 Privacy Protection in Facial Videos

Along the line of AI-enabled smart sensing systems, we have seen that video-capturing devices have become ubiquitous in our daily life. These devices greatly facilitate us to share our life with friends and communicate online with others. Yet

have we realized that whenever a person appears in front of a camera, people can not only recognize his/her identity based on the facial appearance but also monitor some aspects of his/her physiological status such as cardiac activity?

Recent research has shown that contact-free measurement of human physiological signals from facial videos is feasible through computer vision algorithms [12–15]. For instance, remote photoplethysmography (rPPG) technology has attracted a growing amount of R&D interests, which is capable of capturing the subtle color changes of the skin caused by heartbeats in facial videos under ambient light. We can further infer heart rate (HR) [16–21], respiration rate (RR) [22, 23], and heart rate variability (HRV) [24] from extracted rPPG signals. This promising technology can be leveraged to build systems for remote monitoring stress and fatigue during computer tasks [25] and sports training [26].

Recalling the question we have raised at the beginning, we recognize that this emerging technology may cause concerns about physiological privacy. Video-capturing devices not only record a person's appearance but also his/her cardiac activity and physiological status simultaneously. This kind of physiological information intrinsically present in facial videos may be abused to secretly collect and analyze a person's physiological features with ulterior motives. For example, your opponents can read your physiological status and analyze your psychological activities to gain an advantage in mission-critical negotiation conferences. In daily life, one person's certain health conditions may be revealed unawares by a party without his/her explicit consent, leading to potential privacy concerns.

To this end, it is necessary to develop a privacy protection algorithm to conceal

the physiological signals against rPPG technology. We propose *PulseEdit* to edit the rPPG signals in facial videos for physiological privacy protection but minimize the distortion of the person's appearance. "Edit" in the name indicates that the proposed algorithm can either remove the rPPG signal or modify it to a target one by the user's choice. In addition to physiological privacy protection, PulseEdit can potentially circumvent rPPG-based visual security algorithms, such as liveness detection [27–29] and deepfake detection [30], a potential threat to invalidate these algorithms, providing a direction to revise them and improve the confidence of their output decisions.

Meanwhile, the advancement of rPPG technology also raises a serious concern of identity privacy leakage. rPPG technology usually requires facial videos as inputs, and using facial videos has become a sensitive issue as the videos may be misused and the privacy of the persons being recorded is put into jeopardy. To reduce the identity privacy risk, we develop an anonymization transform that removes sensitive visual information identifying an individual and preserves the physiological information for rPPG analysis. The transformed facial videos contain no identifiable features for facial recognition, but the rPPG technology can still be applied to extract reliable physiological results from them. In the proposed anonymization transform, we provide several options to meet different privacy strength needs. In addition, establishing public benchmark datasets have been playing an essential role in the R&D of rPPG technology. The proposed transform can conceal the subjects' identities and encourage data sharing by the R&D community with few concerns on privacy leakage to foster the advancement of this area.

1.3 Fairness in Machine Learning Models

Machine learning is being used across a wide variety of practical tasks in recent years, influencing many aspects of our daily life. Many companies and government departments have deployed machine learning-based decision-making systems to facilitate decision making in business operations. The U.S. courts use software known as Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) to measure the risk for a defendant to recommit another crime, which helps judges make parole decisions. Banks turn to machine learning models to evaluate the prospective customers' credits and to help decide the approval of loan applications. Like humans, machine learning algorithms can suffer from bias and make their decisions "unfair" [31, 32]. A number of investigations have shown that the existing machine learning systems have fairness issues in a certain sense. For instance, the works by [33, 34] studied the COMPAS software and found a bias against African-Americans: it is more likely to assign a higher risk score to an African-American defendant than to a Caucasian with a similar profile.

In the context of decision making, fairness typically means the equivalent (non-discriminatory) decision performances across different groups of people based on their inherent or acquired characteristics. From this point of view, a number of recent studies introduced *demographic parity* (DP) [35, 36] and *equalized odds* (EO) [37] to characterize and evaluate the fairness level in the machine learning models. To take the COMPAS software as an example, DP means that parole granting rates should be equal across the race groups; and EO enforces that among

defendants who would or would not have gone on to commit a violent crime if released, the rates are equal across the race groups.

The prior work on fairness definition can help machine learning models achieve equitable decision performances across the groups from a probability point of view, such as positive proportion of decisions, true positive rate, false positive rate, and other statistics alike. However, such parity of the group-wise performances is generally not retained when we tune the decision threshold. In other words, the classifiers trained under these parity constraints only "delicately" achieve the fairness requirement in the default decision thresholds and are sensitive and vulnerable to the change of decision thresholds.

To alleviate the limitation of the prior fairness definition, we introduce a new fairness notion that is independent of the classifier's threshold. The new notion makes the classifier a consistent fairness level of classification results against the change of decision threshold, benefiting the threshold tuning of the classifier when we require specific output constraints on the classifier.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we discuss the mechanism to protect the intellectual property of DNN models. Utilizing the adversarial behavior of DNN, we can design a DNN that is functional to the authorized access while dysfunctional to the unauthorized access or illicit use. We conduct experiments to demonstrate the effectiveness of the framework design and evaluate its resistance to several attacks. We also discuss a potential application

of the proposed framework that fully takes advantage of the access-control design against unauthorized usage and attacks.

In Chapter 3, we consider the emerging class of rPPG technologies, which can extract a person's physiological information with a portable color camera under ambient light. We focus on physiological privacy protection and propose *PulseEdit*, a novel and efficient algorithm to prevent physiological information from disclosure. We also validate that it is possible to utilize PulseEdit in adversarial scenarios against some rPPG-based algorithms, such as rPPG-based liveness detection and rPPG-based deepfake detection.

In Chapter 4, we continue the research on these remote physiological sensing technologies and investigate the identity privacy protection for the facial videos used in the rPPG technologies. We propose geometric warping and spatial subband decomposition/reconstruction to remove the identifiable appearance features of a person to enhance identity privacy protection. The experimental results show that it is hard to recognize the person's identity from the processed faces and the rPPG signal is still maintained. This anonymization transform can facilitate the sharing of facial videos for rPPG analysis with fewer concerns of identity privacy leakage.

In Chapter 5, we investigate the bias and fairness issue in machine learning models and propose to alleviate this issue using risk distribution parity. We demonstrate that risk distribution parity is independent of the classifier's threshold, which is a significant difference from the prior definitions of group fairness.

Finally, in Chapter 6, we conclude this dissertation and outline research issues that can be explored in the future.

Chapter 2

Intellectual Property Protection of Deep Neural Networks

2.1 Overview

Deep Neural Network (DNN) is a useful and promising vehicle to solve practical computer vision tasks. For the encouragement of healthy business investment and competition in DNN commercial application, ensuring access control is an important aspect of intellectual property (IP) protection to prevent DNN models from illicit usage and piracy. With this functionality, only authorized users can access the trained model and benefit from its high learning capabilities. This calls for the need to investigate the mechanisms to enable access control to prevent unauthorized users from illicit and improper use of the trained DNN models. In addition to piracy prevention, proper access control also supports privacy protection against the misuse of DNNs that produce decision results of a private or sensitive nature, especially in biomedical analytics [6–8].

So far, one of the widely used approaches is to deploy a conventional password-based login system for identity verification. Although the login system can block external unauthorized access, an insider might easily pass the identity verification

and get access to the whole DNN model. Another intuitive approach is to encrypt the weights of the DNN model with traditional data encryption methods such as RSA or advanced encryption standard (AES). However, this would mean that for the DNN model to be run properly, either the DNN model needs to be put into a trusted computing platform using trusted platform modules (TPMs) so that all the encrypted model parameters can be decrypted and then securely run to provide results; or the DNN computations directly on the encrypted parameters must be enabled through homomorphic encryption [38,39] or other types of encrypted-domain computation tools. Given the large number of parameters, on the order of millions or more, involved in many state-of-the-art DNNs and the overall high computational power required, both of these security strategies would be very expensive under today's secure computing paradigm. It is challenging to fully utilize the off-the-shelf optimized DNN architectures and hardware-software tools designed for learning in plaintext while simultaneously achieving security and efficiency.

We investigate how to tackle this challenge by dividing a DNN system into two parts. More specifically, we maintain the bulk part of the DNN that can be run in a similar way as today's ordinary DNN utilizing the available software-hardware platform and only produce state-of-the-art results for an "authorized" type of input, and in the meantime, we design a transformation module that can provide such "authorized" inputs at a reasonable level of computational complexity. By dividing the overall system into these two parts, we can concentrate the security resource to protect the transformation module, for example, through a trusted computing platform, rather than directly protecting the trained DNN with millions of parameters

that requires a large amount of overall computations.

The next question is what kind of transformation can be applied in the module. We are inspired by the adversarial examples whereby well-designed noise in a small strength can render a poor performance of a DNN model [40, 41]. If the function of the transformation module is finding an adversarial example of the raw input, would it be possible to design a DNN so that when an unauthorized user presents a raw input to the trained model, it would be as if it is contaminated by adversarial noise and thus lead to a poor outcome? But in contrast, an authorized user can obtain the "adversarial example" using the transformation module and the model performance approaches the state-of-the-art.

To this end, we propose a novel key-based framework and investigate the transformation module to make DNN models behave properly only with authorized users. Harnessing the adversarial behavior of deep neural networks, the trained model has differential learning performance between authorized access and unauthorized access. In the new AI era following the revolution of the Internet, people are interacting with mobile devices and Internet of Things (IoT) everyday. Increasingly, DNN models are deployed to equip these electric devices with advanced functionalities, prompting a strong need of strengthening the security and intellectual property protection of DNNs in mobile devices. Our work can address this need by incorporating into the mobile device a key-based key-based transformation module that is lightweight.

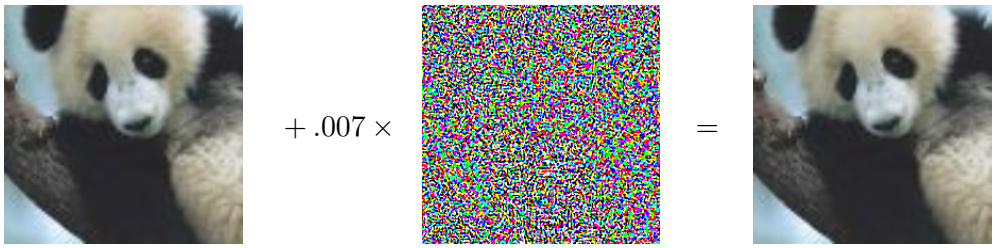
The rest of this chapter is organized as follows. In Section 2.2, we briefly review the related work in IP protection of DNN models. Section 2.3 introduces

the details of the proposed framework. In Section 3.4, we evaluate the performance of the proposed framework and its resistance to the attacks. Section 2.5 holds a related discussion and presents a practical application of the proposed framework and Section 3.7 summarizes the chapter.

2.2 Related Work

2.2.1 Adversarial Examples

Adversarial examples [40] are delicately designed malicious inputs which can fool a deep neural network and result in wrong inference from the network. For example, adding a small perturbation on a panda image which is correctly classified by a DNN can lead the DNN to misclassify the "panda" image as "gibbon".



$\boldsymbol{x}$
"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 2.1: Illustration of adversarial examples to a DNN by perturbing image pixels in a negligible amount.

Define a classifier $f(x) : x \rightarrow y$, mapping the input image x to the label y. In general, the problem of generating *untargeted* adversarial examples can be expressed

as the optimization

$$\begin{aligned} \arg\min_{x^*} \quad & L(x, x^*) \\ s.t. \quad & f(x^*) \neq y, \end{aligned} \tag{2.1}$$

where $L(\cdot)$ is a distance metric between sample x and the adversarial example x^* (*e.g.*, the L_2 norm). Similarly, generating *targeted* adversarial examples on a classifier can be expressed as

$$\begin{aligned} \arg\min_{x^*} \quad & L(x, x^*) \\ s.t. \quad & f(x^*) = y^*, \end{aligned} \tag{2.2}$$

where y^* is a specific target label.

Until now, multiple different approaches of generating adversarial examples have been proposed to attack DNN models, via additive perturbation noise [40, 42] and spatial perturbation [43]. Adversarial transformation networks [44] was proposed as an end-to-end framework to generate adversarial examples of the corresponding deep model. In this chapter, we will incorporate the idea of adversarial examples into our proposed access-control framework

2.2.2 IP protection of DNN models

Research in recent years has begun to address the IP issues of DNN models, mainly from three directions.

The first research direction lies in watermarking on model weights/layers. Inspired by the digital watermarking and fingerprinting techniques, recent studies in [45–47] embedded watermarks into DNN models to protect IP and claim owner-

ship. The authors added a regularization term in loss function during the training to enforce the watermark into the model weights [45, 46] and the model response layers [47], without influencing the performance of the models. The embedded watermarks can be pulled out from the models for verification purposes.

The second group of approaches is watermarking on input samples. The work in [48] poisoned training data by embedding a specific pattern and mislabeling, leaving backdoors in the models. By triggering the backdoors, the IP of the model can be verified. Instead of designing a fixed pattern on the training data, the studies in [49, 50] extract the inherent features of the model and generated a set of specially-crafted adversarial examples as the model fingerprints for IP verification. The generated adversarial examples carve out the decision boundaries of the DNN classifier and are considered as the individual signature of the given DNN model. If the reference model and the test model have matched responses to the adversarial examples, the IP of the test model is claimed.

The third proposed solution is obfuscation on model structure. Data obfuscation is a process of hiding original data with randomness, so as to protect some sensitive data. Layered obfuscation [51] hides the architecture information of the DNN model by replacing a structure block in the network with a shallow and sequential convolutional block. This obfuscation process prevents others from obtaining the "real" structure of the original DNN model. Hardware protected neural network [52] developed an obfuscation infrastructure on model weights with special hardware.

Different from the focus of the prior work on verifying the ownership of the

DNN model, we will address the problem of piracy prevention and privacy protection from the perspective of access control [53].

2.3 Access-control Design

In order to mitigate the impact of piracy and enable the need for access control, we propose a novel framework, presented in Figure 2.2, to train a DNN model with special requirements. Specifically, the DNN model is functional with high inference accuracy for authorized access, while it is dysfunctional for unauthorized users. We denote such a DNN model as *restrictive DNN* in this chapter.

We introduce a key-based *protective transform module* into our proposed framework, whose function is to embed specific signatures for the authorized access, waiting for the restrictive DNN to verify. For authorized users, they can obtain the authorized input using the protective transform module with the right key; for unauthorized users, they may obtain their input (*i.e.*, unauthorized input) with some transforms to emulate the effect of the protective transform module. As shown in the block diagram in Figure 2.2, the classification results have different accuracy rates depending on whether the access is from an authorized user or an unauthorized one. Note that for a general framework, it may be possible to make the restrictive DNN architecture known to the public, while only authorized users have the right key to the protective transform module so that the correct transformation can be made to the input images to enable the restrictive DNN to function properly.

In a nutshell, our proposed framework has two components: the protective

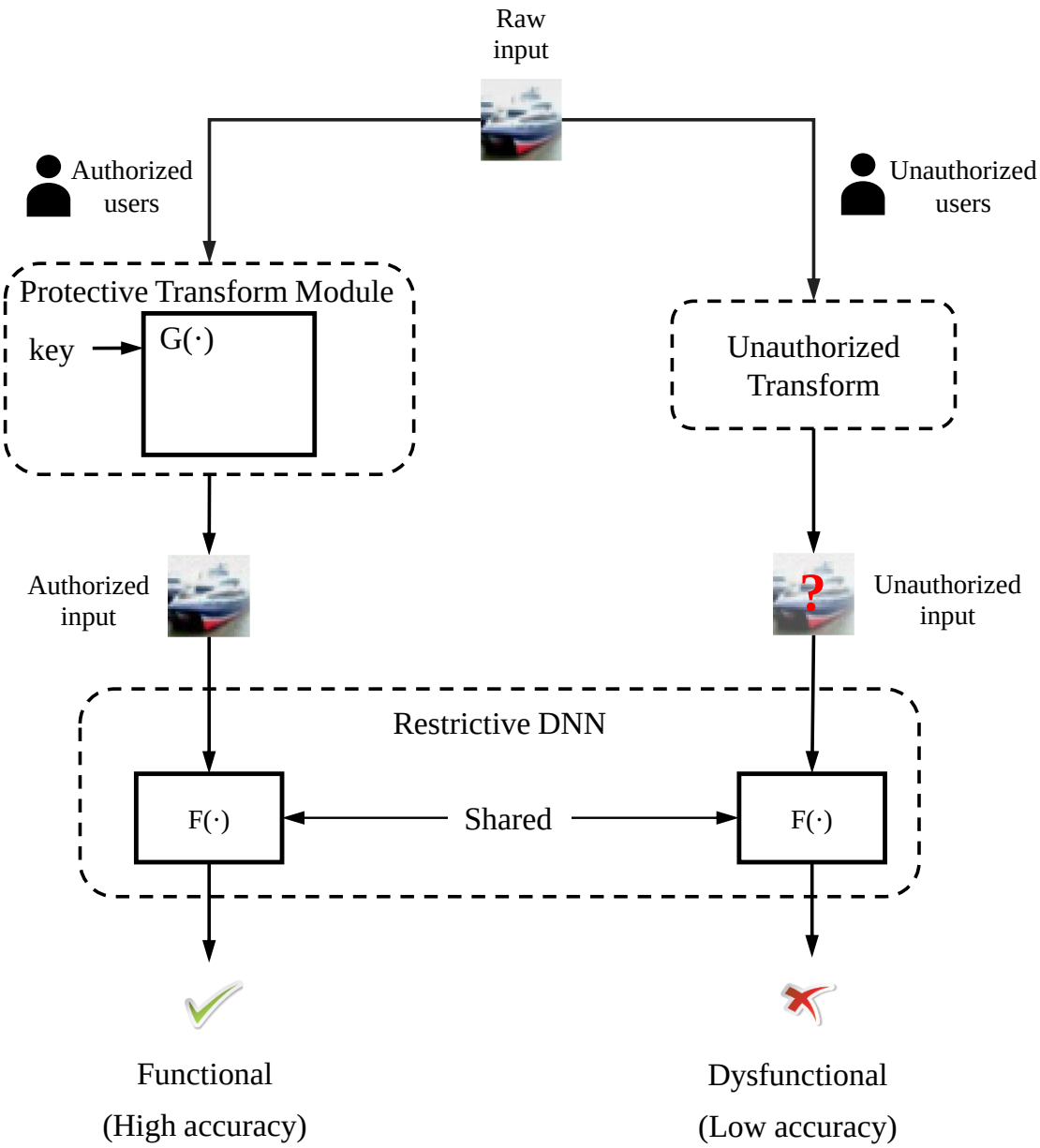


Figure 2.2: The proposed framework for access control protection. The restrictive DNN is the trained DNN model which requires access-control need and protection. The key-based protective transform module can apply the correct transformation on the input images with the right key provided by the authorized users. Only authorized inputs can guarantee the proper behavior of the restrictive DNN, while unauthorized inputs obtained from the unauthorized transform result in the model dysfunction.

transform module and the restrictive DNN that can recognize the authorized and unauthorized inputs and provide differential performance, respectively, thus enabling access control. Conceptually, the protective transform module mimics a "decryption" module, allowing the authorized users to generate the appropriate input for the DNN to function properly. The proposed framework can be tailored to address access control, piracy/theft prevention, and privacy protection in AI deployment. As stated in the introduction, the protective transform module can be implemented securely via TPM. With such focused protection on the protective transform built into our proposed framework, optimization and customization can be done in accordance to specific applications' needs.

2.3.1 Threat Modeling

Understanding the potential and practical threats is essential to help examine adversaries' potential actions and guide us to design, improve, and optimize the system against the threats. Considering the real-world scenarios of different resource availability, there are three types of adversaries that we consider in the design of the protective transform module and the corresponding restrictive DNN. For instance, a simple, *opportunistic* attack is to directly copy and steal the restrictive DNN model. The adversary can access the DNN model and use it as a blackbox for the corresponding classification tasks. This type of attack is common and feasible by even the least resourceful adversaries when no protection is applied to the trained DNN model. We refer to this kind of attack as "direct piracy" attack.

The second scenario considers more sophisticated adversaries with access to the transform module and can evaluate the model performance using different keys. The adversary guesses the key of the protective transform module to see whether one or more guessed keys result in good learning performances comparable to the performance of the right key. We refer to it as "brute force" attack, since the attackers generally need to exhaustively search for the keys to achieve a successful attack.

A more advanced adversary is assumed to have some input-output pairs of the protective transform module, which can be utilized to estimate and infer the transform module. The adversary can attempt to estimate the protective transform module by generating an adversarial transform module to emulate the effect of the genuine transformation. We name this attack as "pair attack".

2.3.2 Training Formulation

To facilitate the discussions, we first define some notations. Let x_r be the raw input and x_{au} the authorized input; $G(\cdot; k)$ denotes the protective transform module with key k, thus $x_{au} = G(x_r; k)$; $F(\cdot)$ denotes the restrictive DNN.

To ensure the restrictive DNN functional to the authorized input x_{au}, we define the loss for x_{au} as the cross-entropy loss:

$$E_{au} = -\sum_{i=1}^{N} p_i \log F(x_{au})_i, \tag{2.3}$$

where N is the number of class, the vector $(p_1, p_2, ...p_N)$ is the one-hot encoding of

ground truth label, and $F(\cdot)_i$ is the i-th entry of output probability vector of the restrictive DNN F.

To achieve the goal of making the model dysfunctional under the unauthorized input x_{un}, we define the unauthorized loss as

$$E_{un}(x_{un}) = -\frac{1}{N}\sum_{i=1}^{N} F(x_{un})_i. \tag{2.4}$$

When we minimize E_{un} in (2.4), the entries of the output probability vector incline to be closer to each other. Thus, the DNN has high probability to provide wrong prediction on the unauthorized input x_{un}.

We consider the three unauthorized cases described in Section 2.3.1 in the training formulation:

- Direct piracy: we consider the raw input x_r as unauthorized input to the restrictive DNN, *i.e.*, $x_{un} = x_r$.

- Brute force: an adversary may guess a key $\tilde{k} \neq k$ to attack the restrictive DNN. To mitigate this threat, we randomly select a key $\tilde{k} \neq k$ to generate unauthorized input in each training iteration, *i.e.*, $x_{un} = G(x_r; \tilde{k})$.

- Pair attack: an adversary may come up with an adversarial transform module $\tilde{G}$ to emulate the effect of the protective transform module. To defend this attack, we generate unauthorized input using $\tilde{G}$, *i.e.*, $x_{un} = \tilde{G}(x_r)$.

Combining the authorized loss and unauthorized loss, we formulate the overall

loss function E as

$$E = E_{au} + \beta\Big(E_{un}(x_r) + E_{un}\big(G(x_r;\tilde{k})\big) + E_{un}\big(\tilde{G}(x_r)\big)\Big), \tag{2.5}$$

where β governs the balance between the authorized input and the corresponding unauthorized input.

2.3.3 Protective Transform

The protective transform is an important component in our framework. Inspired by adversarial examples of DNN as motivated in the introduction, we design this module by applying perturbations on the raw input. Figure 2.3 illustrates the structures of two proposed protective transform modules with additive perturbation and spatial perturbation, respectively. We employ a random neural network (RandNN) [54] in the protective transform module, whose weights are generated from a pseudo-random number generator (PRNG) given the key k. RandNN is a neural network with random weights between layers, which has been used to design encryption algorithms [55, 56]. In our implementation, the weights are uniformly distributed on $\{-1, 0, 1\}$. Then, we quantize the output of the RandNN to $[-s, s]$ with $2N + 1$ quantization step to obtain the random perturbation pattern $P(\cdot; k)$, and apply it on the raw input x_r to provide the transformed input x_t. The hyperparameter s controls the perturbation intensity imposed onto the perturbation pattern. On the whole, we use (2.6) to fulfill the quantized random perturbation

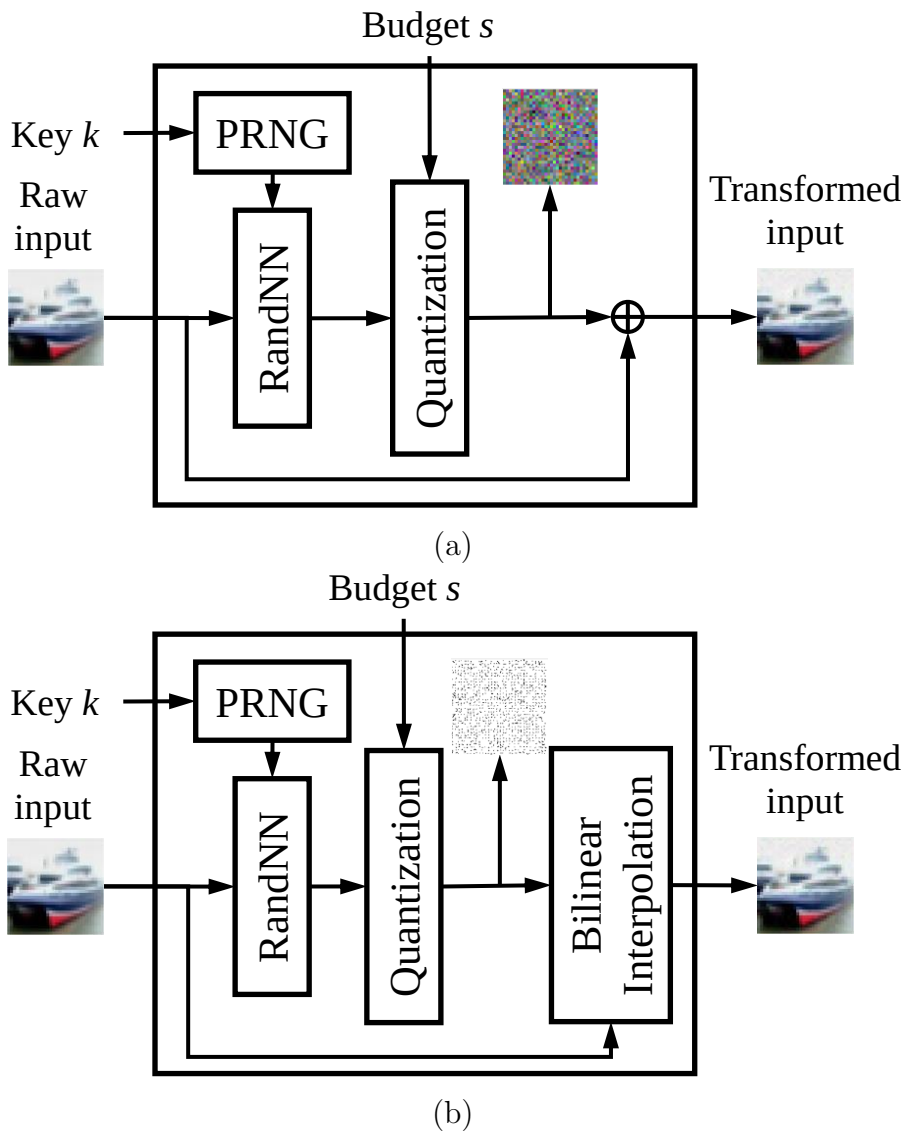


Figure 2.3: The structures of two proposed protective transform modules: (a) additive perturbation and (b) spatial perturbation. The weights in the RandNN are generated from the pseudo-random number generator with the key. The output of the RandNN is then quantized to range $[-s, s]$ to generate the perturbation pattern for (a) additive perturbation and (b) 2D flow field, respectively. Finally, the pattern is applied on the raw input to obtain the transformed input.

pattern:

$$\begin{aligned} y_{rd} &= R(x_r; k), \\ P(x_r; k) &= \text{round}\Big(N\big(\frac{2\big(y_{rd} - \min(y_{rd})\big)}{\max(y_{rd}) - \min(y_{rd})} - 1\big)\Big) \cdot \frac{s}{N}, \end{aligned} \tag{2.6}$$

where $R(\cdot)$ denotes the RandNN operation and round$(\cdot)$ denotes the rounding operation. The protective transform module can be done in a trusted computing environment with strong protection to prevent adversary from cracking it. Since the transform module is lightweight compared with the DNN model, we can focus the protection only on this module without spending expensive computation.

We discuss the detail of two perturbation modes as follows:

- Additive perturbation: In this mode, the generated random pattern $P(x_r; k)$ provides an additive perturbation of the image pixels. The transformed image x_t is synthesized as

$$x_t = x_r + P(x_r; k) \tag{2.7}$$

- Spatial perturbation: In this mode, the generated random pattern $P(x_r; k)$ is a 2D flow field, which represents the per-pixel displacement between the transformed image x_t and the raw input x_r [43]. We denote the displacement of the i-th pixel in the flow field as $(\Delta u^i, \Delta v^i)$. The relationship between the two images x_r and x_t can be expressed as $x_r(u_r^i, v_r^i) = x_t(u_t^i, v_t^i)$, where (u_r^i, v_r^i) and (u_t^i, v_t^i) denote the i-th pixel location of x_r and x_t, respectively, and $(u_r^i, v_r^i) = (u_t^i - \Delta u^i, v_t^i - \Delta v^i)$. We set (u_t^i, v_t^i) on an integer-valued image grid, then (u_r^i, v_r^i) can be fractional numbers and does not necessarily lie on an integer-valued grid. We use the differentiable bilinear interpolation [57] to transform the input image according to the generated flow field. Specifically,

the pixel x_t^i at the location (u_t^i, v_t^i) is calculated as

$$x_t^i = \sum_{x_r^j \in \mathcal{N}(x_r^i)} x_r^j (1 - |u_r^i - u_r^j|)(1 - |v_r^i - v_r^j|), \tag{2.8}$$

where $\mathcal{N}(x_r^i)$ are the 4-pixel neighbors of the pixel at (u_r^i, v_r^i) consisting of the top-left pixel at $(\lfloor u_r^i \rfloor, \lfloor v_r^i \rfloor)$, top-right pixel at $(\lfloor u_r^i \rfloor, \lfloor v_r^i \rfloor + 1)$, bottom-left pixel at $(\lfloor u_r^i \rfloor + 1, \lfloor v_r^i \rfloor)$, and bottom-right pixel at $(\lfloor u_r^i \rfloor + 1, \lfloor v_r^i \rfloor + 1)$. The transformed image x_t is obtained by calculating every pixel x_t^i using (2.8).

From adversary point of view, he can build up an adversarial transform module to replace the genuine protective transform module for attack purpose. We assume that the adversary knows the structure and hyperparameters s and N in the protective transform module, but has no knowledge of the key, *i.e.*, the weights in the RandNN. The adversary's goal is to recover these weights via some input-output pairs (x_r, x_{au}) obtained from the genuine protective transform module.

To obtain an adversarial transform module $\tilde{G}$, the adversary can recover the weights by

$$\min_{\tilde{G}} ||x_{au} - \tilde{G}(x_r)||_2^2. \tag{2.9}$$

The discretization in the RandNN weights and the perturbation pattern in the protective tranform module G makes (2.9) difficult to optimize. Thus, we employ the following relaxation in the adversarial transform module $\tilde{G}$:

- The RandNN weights in $\tilde{R}$ are real and the output is $\tilde{y}_{rd} = \tilde{R}(x_r)$.
- We relax the perturbation pattern in $\tilde{G}$ to real numbers within range $[-s, s]$,

i.e.,

$$\tilde{P}(x_r) = \Big(\frac{2\big(\tilde{y}_{rd} - \min(\tilde{y}_{rd})\big)}{\max(\tilde{y}_{rd}) - \min(\tilde{y}_{rd})} - 1\Big) \cdot s \tag{2.10}$$

We obtain the adversarial transform module $\tilde{G}$ using the whole training data and use it in the training of the restrictive DNN.

2.4 Experimental Results

In this section, we present the experimental results to demonstrate the effectiveness of our proposed framework. We use the MNIST [58], and the CIFAR10/100 [59] datasets in the experiments for proof of concept. The MNIST dataset consists of 28×28 grayscale fashion images in 10 classes, and have $60,000$ training images and $10,000$ test images. The CIFAR10 and CIFAR100 datasets contain 32×32 color images in 10 and 100 classes, respectively, and have $50,000$ training images and $10,000$ test images.

2.4.1 Network Structures and Training Settings

We use stacked convolutional neural network (CNN) and Resnet-56 [11] as the architectures of the restrictive DNN. The detailed structures are shown in Table 2.1 and 2.2, respectively. In the convolutional layer (conv), the parameters of the building block denote the filter size and the number of output channels; in the pooling layer (pool), the parameters denote the pooling method and the pooling size; the term "fc" denotes fully-connected layer. If not specified, rectified linear unit (ReLU) is used as the activation function in each layer. In the protective transform mod-

ule, we build the RandNN as a 4-layer fully-connected network (no bias term) with 256 hidden neurons in the intermediate layers, which is shown in Figure 2.4. The RandNN weights are generated from a discrete random number generator $\{-1, 0, 1\}$ with a key k.

Since all components in our proposed framework are differentiable, we use stochastic gradient descent (SGD) optimizer with momentum 0.9 to train the restrictive DNN. The weight decay is 1×10^{-4}, and the batch size is 64. For the stacked CNN model, we set the initial learning rate 0.01, divide it by 10 at 30k iterations, and terminate training at 60k iterations. For the Resnet-56 model, we set the initial learning rate 0.1, divide it by 10 at 40k and 64k iterations, and terminate training at 80k iterations.

Layer	Output size	Building block
conv1	28×28	$[3 \times 3, 32]$
pool1	14×14	max, 2×2
conv2	14×14	$[3 \times 3, 64]$
pool2	7×7	max, 2×2
fc1	1024	dropout: 0.5
fc2/output	10	softmax

Table 2.1: Structure of stacked CNN used in the experiments

Layer	Output size	Building block
conv1	32×32	$[3 \times 3, 16]$
conv2_x	32×32	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 9$
conv3_x	16×16	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 9$
conv4_x	8×8	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 9$
output	10	global avg-pool, fc, softmax

Table 2.2: Structure of Resnet-56 used in the experiments

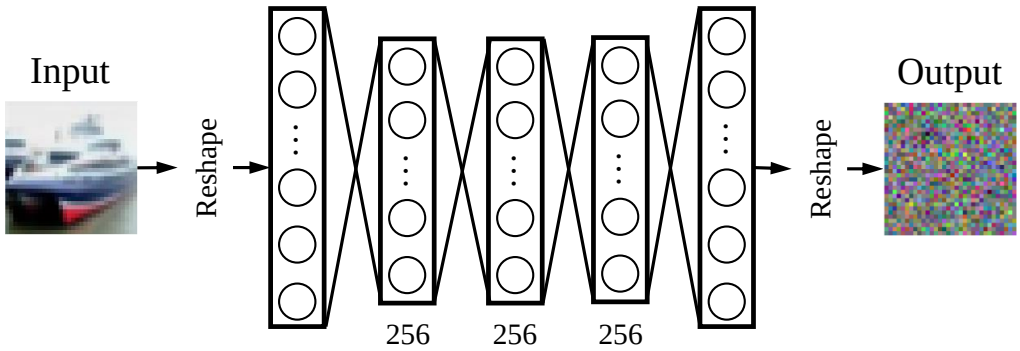


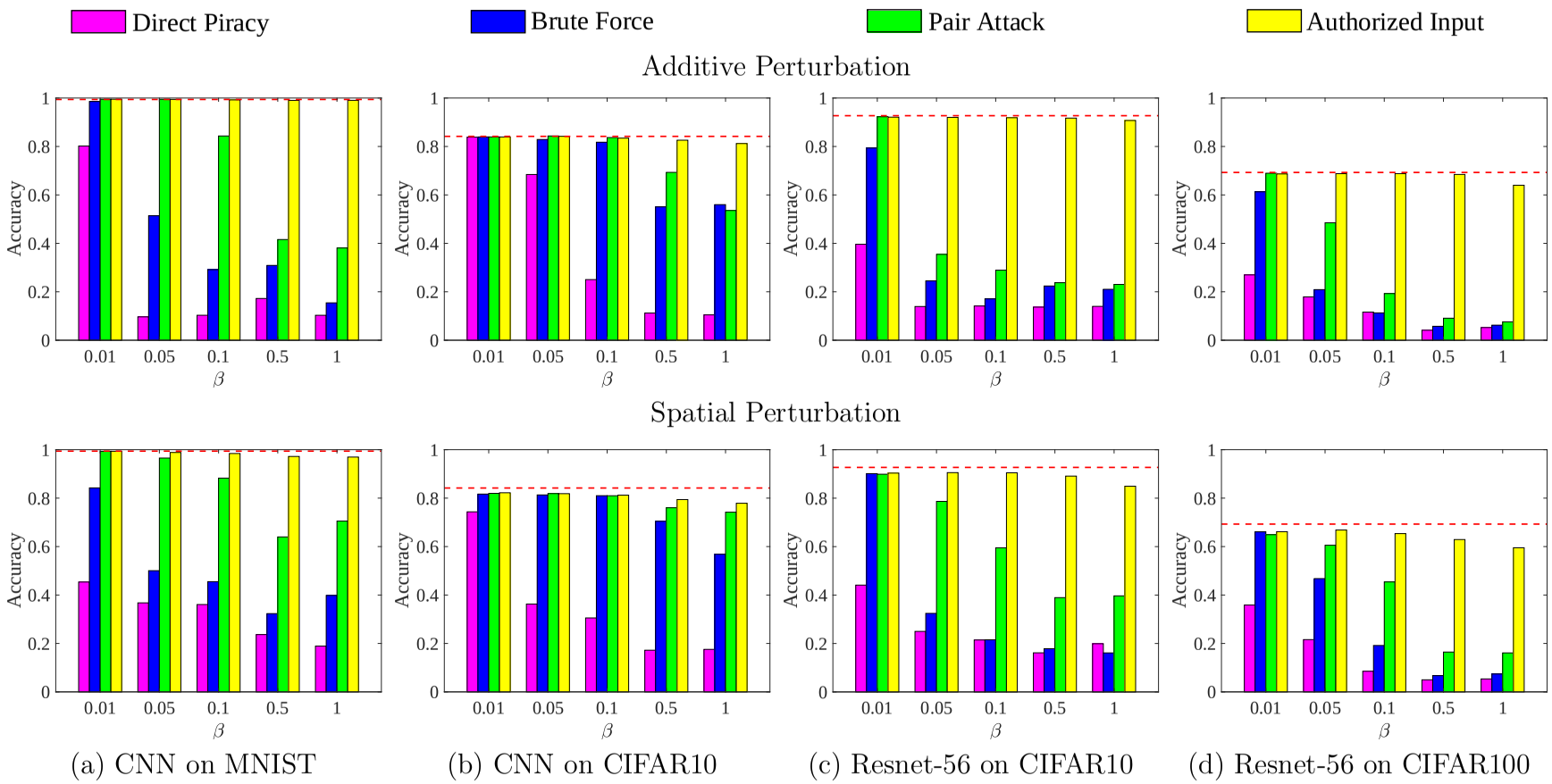
Figure 2.4: The structure of RandNN used in the experiments. For the input with size of $M \times M \times 3$, the output has size of $M \times M \times 3$ in additive perturbation mode and $M \times M \times 2$ in spatial perturbation mode.

We normalize the images to the range $[-1, 1]$ in the three datasets as data preprocessing. For the CIFAR10/100 dataset, we also follow a simple data augmentation described in [11] for training: an image is padded with 4 pixels on each side, and then, a 32×32 crop is randomly sampled from each padded image or its horizontally flipped version.

2.4.2 Performance of Access-control Design

We trained the restrictive DNN on the three datasets with two DNN architectures discussed in Section 2.4.1 using two proposed protective transform modules, *i.e.*, additive perturbation and spatial perturbation. We set the perturbation budge $s = 0.1$ in additive perturbation and $s = 1$ in spatial perturbation. We evaluated the performance of the proposed framework with different value of $\beta = \{0.01, 0.05, 0.1, 0.5, 1\}$. We also trained the ordinary DNNs for baseline references to analyze the cost paid for the design of access control.

Figure 2.5 presents the performance of the restrictive DNN with different β



(a) CNN on MNIST (b) CNN on CIFAR10 (c) Resnet-56 on CIFAR10 (d) Resnet-56 on CIFAR100

Figure 2.5: Performance of the restrictive DNN with different β using additive perturbation (first row) and spatial perturbation (second row) for the unauthorized input (*i.e.*, direct piracy, brute force, and pair attack) and authorized input. The red dash lines indicate the accuracy of the corresponding baseline models.

for the unauthorized access (*i.e.*, direct piracy, brute force, and pair attack) and authorized access on the MNIST and the CIFAR10/100 datasets. The red dash lines in the figures indicate the accuracy of the corresponding baseline models. The performance of the unauthorized access is preliminarily validated here with one guessed key in the brute force case and one adversarial transform module in the pair attack case, to facilitate us to establish a rough sense of the hyperparameter impact on the models and the model ability against the unauthorized input. We will have a more thorough resilience analysis against attacks in Section 2.4.5.

From Figure 2.5, we generally observe that increasing β can enlarge the performance gaps between the authorized input and the unauthorized input in both proposed protective transforms. For the three unauthorized cases, the difficulty of the model resilience against the unauthorized access is direct piracy $<$ brute force $<$ pair attack, whose order is consistent with the resource availability of adversaries.

If choosing a suitable β, we can see noticeable performance gaps between the authorized and unauthorized access, but a small degradation of the model performance compared with the baseline, which fulfills our design goal for the access control. This observation states that our proposed framework can mostly maintain the classification capability of the DNN architectures and distinguish between the authorized and unauthorized access. Comparing two protective transformations, the spatial perturbation is more sophisticated than the additive perturbation, which applies geometric warping rather than simple additive noise. Figure 2.5 reveals that the additive perturbation, in general, can achieve slightly closer performance to the baseline reference than the spatial perturbation does.

With the tradeoff between the performance on authorized and unauthorized access, we choose $\beta = 0.1$ for stacked CNN on MNIST, $\beta = 0.5$ for stacked CNN on CIFAR10, $\beta = 0.05$ for Resnet-56 on CIFAR10/100, whose statistics are shown in Table 2.3. We will use this setting in the subsequent experiments and evaluations. Table 2.3 shows that the models have no more than 5% performance drop to the authorized access compared with the baseline reference, while the models have much lower accuracy in performance to the attacks. Comparing the two models on the CIFAR10 dataset, we observe that if one network has a higher model capacity, it has a smaller performance penalty on the authorized access and has a larger performance difference between the authorized and unauthorized access. This suggests that higher model capacity is beneficial to the model capability of differentiation between authorized and unauthorized access in our proposed framework.

Dataset		MNIST	CIFAR10	CIFAR10	CIFAR100
Model		CNN	CNN	Resnet-56	Resnet-56
Baseline		99.41%	84.18%	92.71%	69.28%
Additive Perturbation					
Authorized		99.23%	82.59%	92.02%	68.77%
Unauthorized Access	Direct Piracy	10.31%	11.21%	13.92%	17.88%
	Brute Force	29.15%	55.07%	24.50%	20.90%
	Pair Attack	84.34%	69.31%	35.45%	48.50%
Spatial Perturbation					
Authorized		98.43%	79.41%	90.55%	66.83%
Unauthorized Access	Direct Piracy	36.07%	17.21%	24.99%	21.59%
	Brute Force	45.52%	70.44%	32.44%	46.78%
	Pair Attack	88.27%	76.02%	78.68%	60.55%

Table 2.3: Performance of the proposed framework on unauthorized and authorized access

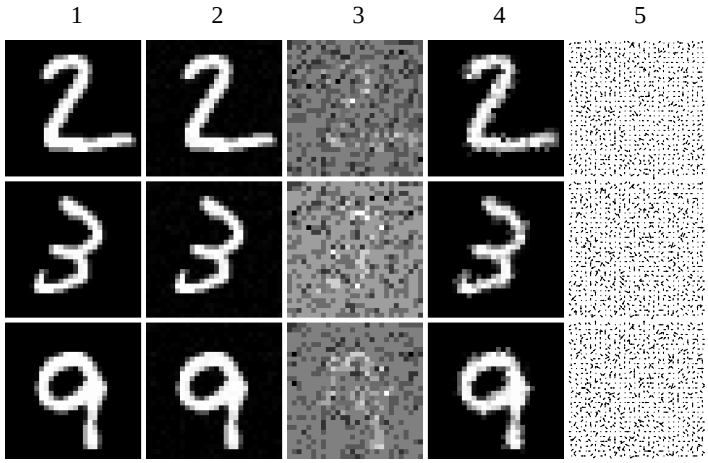
To sum up, the access-control consideration in the proposed framework is effective in defense against unauthorized access and only imposes a small penalty in the

model performance for authorized access compared with the baseline DNN models. Comparing two perturbation modes, we recommend additive perturbation in the protective transform module, which has a less performance penalty for authorized access.

2.4.3 Visualization of Image Inputs

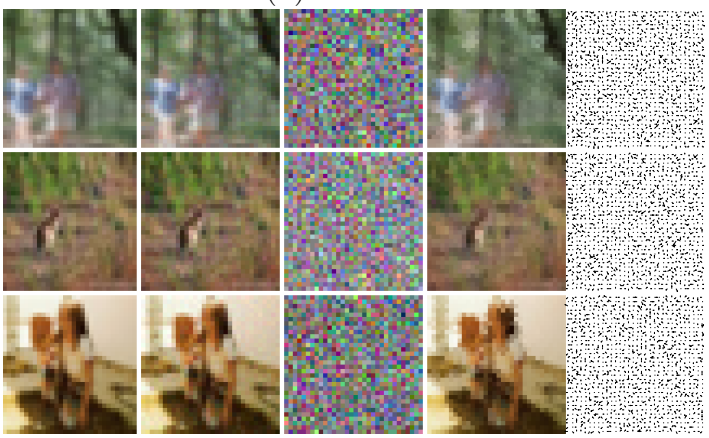
We investigate the effects of the proposed protective transform modules acting on the authorized inputs. Figure 2.6 visualizes the raw and the authorized inputs with different protective transform modules. Thanks to the moderate value of the perturbation budget s, the raw input and the corresponding authorized input, as indicated in Figure 2.6, appear to be very similar, and human eyes can hardly distinguish them. This observation suggests that the restrictive DNN models can differentiate the authorized and the unauthorized inputs, achieving a noticeable gap in classification accuracy, even though the perturbation is nearly imperceptible. Such differential learning performances are similar to the phenomenon of adversarial examples stated in the recent literature.

Table 2.4 quantitatively demonstrates the similarity between the raw and the authorized inputs in two perceptual criteria: peak signal to noise ratio (PSNR) and structural similarity index (SSIM) [60]. We tested all pairs of the inputs in the test set and computed the mean of the perceptual criteria for each model. From the statistics in Table 2.4, we can see that the additive perturbation achieves slightly better perceptual similarity than the spatial perturbation. Specifically, the additive



(a) MNIST

(b) CIFAR10



(c) CIFAR100

Figure 2.6: Examples of the raw input and the authorized input for the restrictive DNNs. Column 1 is the raw input. Column 2 and 3 are the authorized input with additive perturbation and the corresponding additive perturbation. Column 4 and 5 are the authorized input with spatial perturbation and the corresponding flow field.

perturbation has generally $8 - 16$ dB higher PSNR than the spatial perturbation. Since the PSNR criterion is good at accounting for the additive noise, it is relatively easy to maintain a high PSNR in additive perturbation. In SSIM criterion that is widely regarded as better reflecting perceptual quality than PSNR, the two perturbations show comparable performances in perceptual similarity and most of the SSIM indices are close to 1. Overall, we can see that both two types of perturbations are able to leave negligible perceptual changes on the images, but trigger the dysfunction of the restrictive DNN.

Dataset	Model	Additive Perturbation		Spatial Perturbation	
		PSNR	SSIM	PSNR	SSIM
MNIST	CNN	37.71 dB	0.825	21.91 dB	0.942
CIFAR10	CNN	36.44 dB	0.983	28.39 dB	0.955
CIFAR10	Resnet-56	36.44 dB	0.983	28.39 dB	0.955
CIFAR100	Resnet-56	36.52 dB	0.983	28.62 dB	0.958

Table 2.4: Visual similarity between the image inputs in two proposed protective transform modules

Due to the small difference between the raw and the authorized inputs, the proposed framework also has a side benefit: protecting data privacy. Since the authorized inputs retain most of the original physical meanings and appearance, they can replace the raw images in the real world and can be circulated and used as the raw images. Taking advantage of the perceptual similarity property, data privacy is protected by circulating the transformed copy of the image and avoiding the disclosure of the raw image itself, especially when we are dealing with images of sensitive nature, such as biometric identification.

2.4.4 Impact of Perturbation Budget

We study the impact of perturbation budget s in the protective transform module on the performance of the restrictive DNN. Figure 2.7 shows some examples of two perturbation modes with different s value from three datasets. We evaluate $s = \{0.02, 0.1, 1\}$ for additive perturbation and $s = \{0.2, 1, 5\}$ for spatial perturbation, which reflects small, medium, and large perturbation budget, respectively. The medium level of s was used in the experiments in Section 2.4.2. Small s makes the transformed image very close to the raw image, while large s destroys the original shapes and textures in the raw image to some extent.

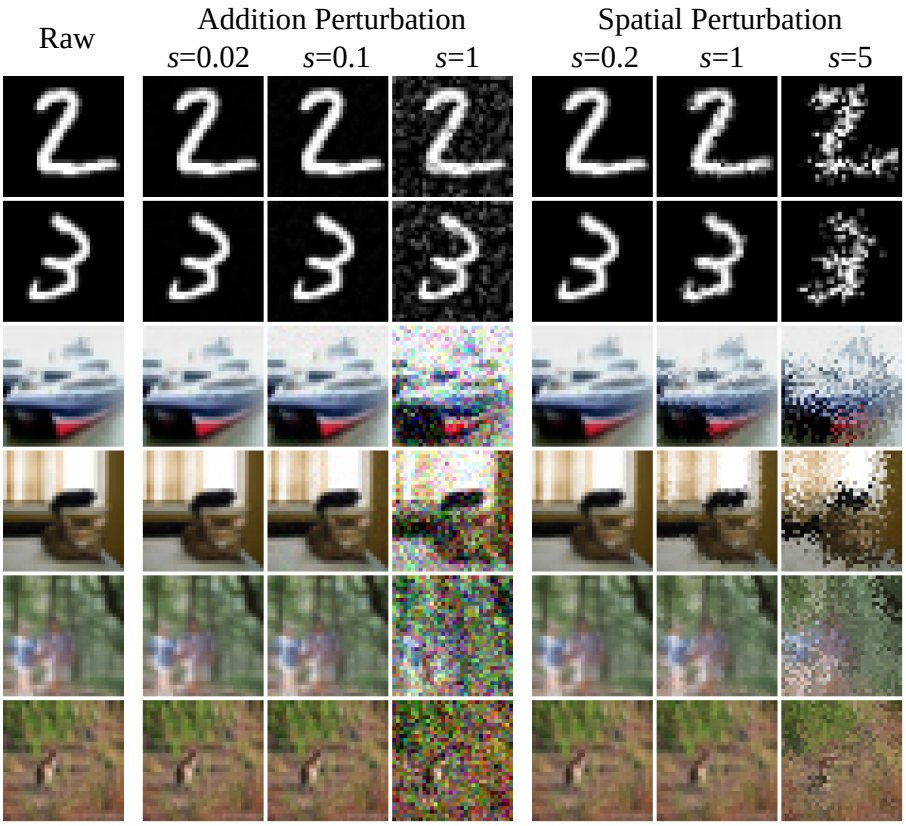
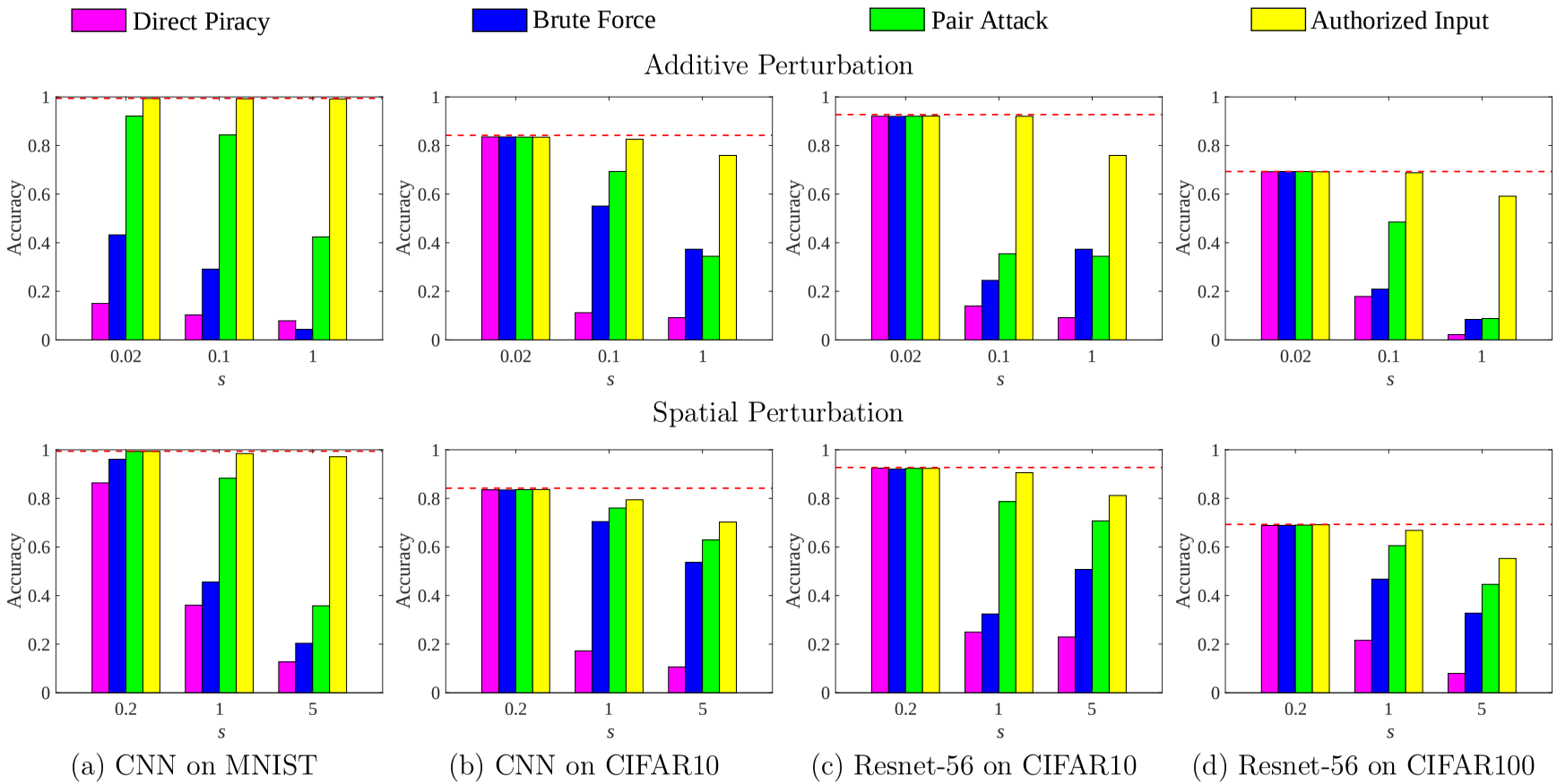


Figure 2.7: Examples of two perturbation modes with different perturbation budget s from three datasets. s controls the similarity of the raw image and the transformed image.

Figure 2.8 presents the performance of the restrictive DNN with different s for the unauthorized access (*i.e.*, direct piracy, brute force, and pair attack) and authorized access on the MNIST and the CIFAR10/100 datasets. The red dash lines in the figures indicate the accuracy of the corresponding baseline models. Similar to Section 2.4.2, the performance of the unauthorized access is preliminarily tested here with one guessed key in the brute force case and one adversarial transform module in pair attack case to establish a rough sense of the hyperparameter impact on the models.

From Figure 2.8, we observe that small s can maintain the model performance on authorized access, compared with the baseline models, but the models can hardly provide the differential learning performance between authorized access and unauthorized access. For example, the models in Fig 2.8(b), (c), and (d) have almost the same performance between authorized and unauthorized access. This is because small s makes the authorized input and unauthorized input very close to each other, and the model does not have enough high capacity to differentiate them. On the other hand, large s enlarges the image distance between authorized and unauthorized access, and we can see that the models are able to respond to authorized and unauthorized access with differential learning performance. Nevertheless, we have more performance penalties for the authorized access. For the models on the CIFAR10/100 datasets, there are more than 10% accuracy drops to the authorized access, compared with the corresponding baseline models. Large s introduces much random pattern on the images and makes the set of authorized images substantially different from the set of raw images. Since the random pattern is less learnable than



(a) CNN on MNIST (b) CNN on CIFAR10 (c) Resnet-56 on CIFAR10 (d) Resnet-56 on CIFAR100

Figure 2.8: Performance of the restrictive DNN with different s using additive perturbation (first row) and spatial perturbation (second row) for the unauthorized input (*i.e.*, direct piracy, brute force, and pair attack) and authorized input. The red dash lines indicate the accuracy of the corresponding baseline models.

the raw images, the learning performance of DNN is degraded for large s.

In all, the hyperparameter s is an important factor to balance the model performance on authorized access and the differential learning performance between authorized and unauthorized access.

2.4.5 Resilience Analysis against Attacks

The performance against attacks is an important aspect to examine the effectiveness of our proposed framework in the real world. We investigate the performances of the trained models against attacks and present the results in Table 2.5. As discussed earlier, "direct piracy" refers to directly copying the restrictive DNN model, equivalent to feeding raw images to the model; "brute force" refers to attacks by providing an arbitrary key, *i.e.*, trying trial and error exhaustively; and "pair attack" refers to attacks with the pairs of the raw and the corresponding authorized inputs. For "pair attack", we assume the adversary has the knowledge of the structure and hyperparameter in the protective transform module, except the random weights in the RandNN controlled by the key. We apply the same procedure described in Section 2.3.3 to obtain the adversarial transform module. We conduct "brute force" attacks 10000 times and "pair attacks" 10 times, and record the best attack performance in Table 2.5.

In the direct piracy scenario, we assume the restrictive DNN is stolen and copied by an adversary. Table 2.5 shows that the model accuracy is more than 45% lower than the authorized input, indicating that direct piracy of the restrictive DNN

Dataset			MNIST	CIFAR10	CIFAR10	CIFAR100
Model			CNN	CNN	Resnet-56	Resnet-56
Additive Perturbation						
Authorized			99.23%	82.59%	92.02%	68.77%
Unauthorized Access	Direct Piracy		10.31% (-88.92%)	11.21% (-71.38%)	13.92% (-78.10%)	17.88% (-50.89%)
Unauthorized Access	Brute Force		84.26% (-14.97%)	76.09% (-6.50%)	59.05% (-32.97%)	44.50% (-24.27%)
Unauthorized Access	Pair Attack	10%	78.30% (-20.93%)	72.76% (-9.83%)	42.07% (-49.95%)	43.50% (-25.27%)
Unauthorized Access	Pair Attack	25%	83.01% (-16.22%)	77.95% (-4.64%)	49.38% (-42.65%)	46.07% (-22.70%)
Unauthorized Access	Pair Attack	50%	86.83% (-12.40%)	79.31% (-3.28%)	60.82% (-31.20%)	49.77% (-19.00%)
Spatial Perturbation						
Authorized			98.43%	79.41%	90.55%	66.83%
Unauthorized Access	Direct Piracy		36.07% (-62.36%)	17.21% (-62.20%)	24.99% (-65.56%)	21.59% (-45.24%)
Unauthorized Access	Brute Force		66.63% (-31.80%)	73.43% (-5.98%)	48.46% (-42.09%)	57.08% (-9.75%)
Unauthorized Access	Pair Attack	10%	91.32% (-7.11%)	75.99% (-3.42%)	79.09% (-11.46%)	60.60% (-6.23%)
Unauthorized Access	Pair Attack	25%	91.53% (-6.90%)	76.46% (-2.95%)	82.16% (-8.39%)	61.09% (-5.74%)
Unauthorized Access	Pair Attack	50%	92.44% (-5.99%)	77.20% (-2.21%)	85.15% (-5.40%)	61.71% (-5.12%)

Table 2.5: The resilience of the restrictive DNN against various attacks

merely leads to invalid classification results and the proposed framework can protect the DNN model from theft and copy.

In the brute force scenario, an adversary expects to try the key of the protective transform module by trial and error, aiming to find a "good" key that can make the restrictive DNN provide satisfactory inference performance. We randomly chose 10000 keys different from the correct key k and tested the performance of the restrictive DNN models. From the column of "Brute Force" in Table 2.5, we can see that the accuracy is lower than the authorized input, and the performance gap can

be as large as 5% to 40%. In general, the brute force strategy fails to find a "good" key to have a comparable result to the correct key.

According to the experiments of the brute force attack, how low is the probability p to find a "good" key for each trial? Since we do not obtain a "good" key out of 10000 trials, the total probability of the successful attack should be less than $\frac{1}{10000}$. Thus, we have

$$1 - (1-p)^{10000} < \frac{1}{10000}. \tag{2.11}$$

Solve (2.11) and we have $p < 1 \times 10^{-8}$, indicating that "good" keys only take up no more than 1×10^{-8} of the total space of the keys. Thanks to the term $E_{un}\big(G(x_r; \tilde{k})\big)$ addressing the brute force attack in (2.5), the number of "good" keys is kept at such a low level that it ensures to enhance the difficulty for the brute force attack to succeed.

For the "pair attack", the task for an adversary is to estimate an adversarial transform module to emulate the genuine protective transform module, with the help of a number of input-output pairs. In our experiments, we provide the pairs generated from 10%, 25%, and 100% of the training data used to train the restrictive DNNs to estimate the module, respectively. From Table 2.5, we observe that as the number of pairs increases in the pair attack, the attack performance improves, shrinking the performance gap between the authorized access and the pair attack. Involving more pairs to estimate the adversarial transform module contributes to a better module to emulate the real protective transform module. Comparing the three investigated attack strategies, we can see that the pair attack, in general, is

the most effective. For example, the best performance is the pair attack on the CNN model for CIFAR10 using spatial perturbation mode, which can achieve accuracy of 77.20%, a performance drop of 2.21% compared with the authorized access.

Note that our "pair attack" scenario assumes that the adversary has the most prior knowledge of the access-control framework, including the structure and parameters in the protective transform module as well as the samples in the training set. Even in this most resourceful and advantageous condition, there is still a noticeable performance gap between the attack and the authorized access.

Overall, Table 2.5 demonstrates the resistance of the proposed framework to the various attacks, reflecting the noticeable performance gaps between the response of restrictive DNN models on the authorized and the unauthorized access.

To evaluate how accurate the attacks can achieve to estimate the random patterns imposed on the images, we compute the estimated error of random patterns for the three attacks and report the best results in Table 2.6. The estimated error is defined as the mean squared error of the random patterns between the authorized access and the attacks. The experiments were conducted with budge $s = 0.1$ in additive perturbation and $s = 1$ in spatial perturbation.

The estimated error can reflect how successful the attack is. The pair attack can achieve the minimum estimated error of the random pattern. The more input-output pairs used in the pair attack, the smaller estimated error we can achieve, which is consistent with the attack performance of the three unauthorized scenarios.

Dataset			MNIST	CIFAR10	CIFAR100
Additive	Direct Piracy		1.00	3.03	3.03
	Brute Force		1.76	5.63	5.61
	Pair Attack	10%	0.20	0.78	0.73
		25%	0.17	0.73	0.68
		100%	0.15	0.66	0.63
Spatial	Direct Piracy		145.34	203.80	203.42
	Brute Force		267.85	381.33	379.86
	Pair Attack	10%	58.58	59.90	55.44
		25%	50.38	54.82	51.21
		100%	40.24	49.10	46.09

Table 2.6: Estimated error of random patterns for the three attacks

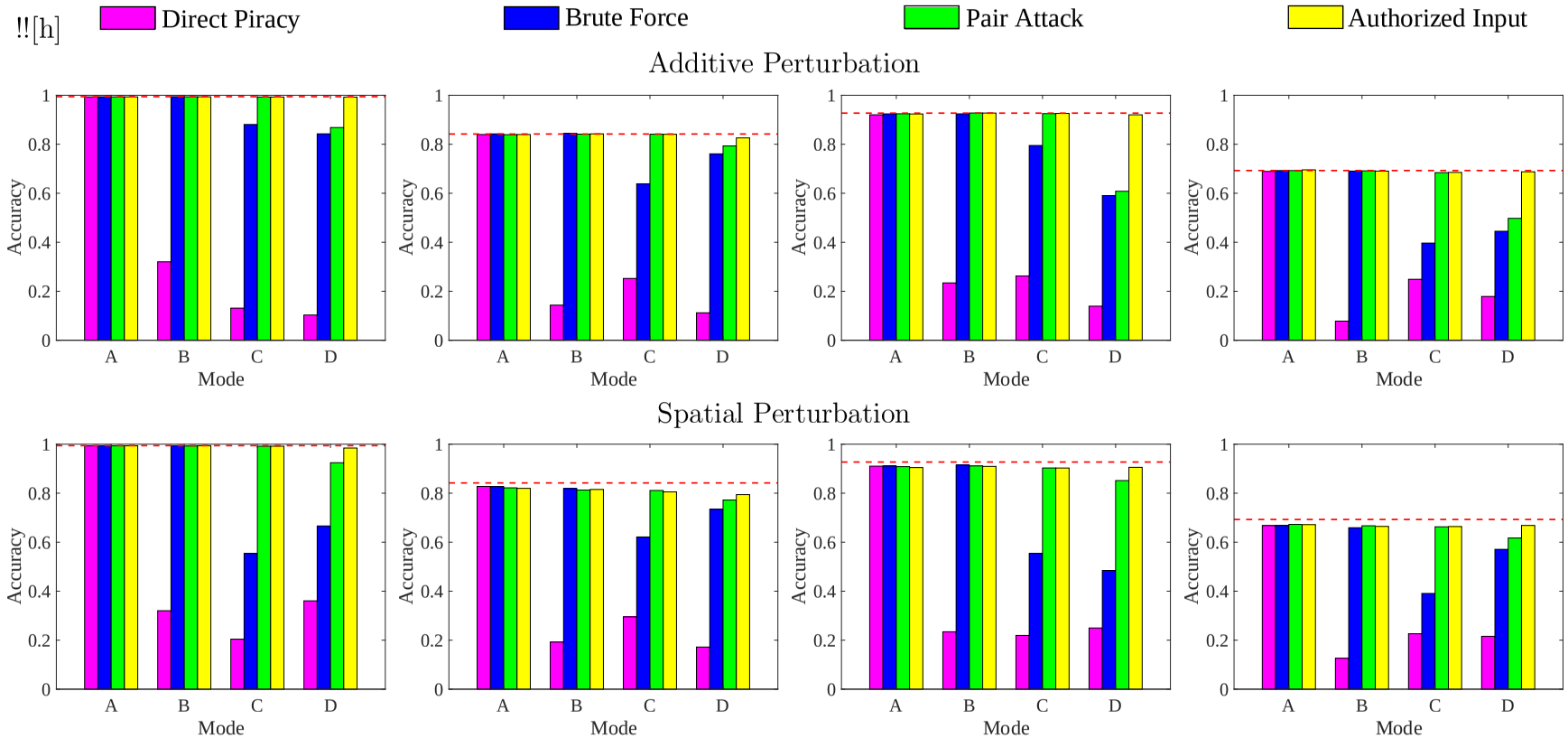
2.4.6 Ablation Study on Training Formulation

In the proposed training formulation, we incorporate four terms in (2.5), dealing with authorized input, direct piracy attack, brute force attack, and pair attack, respectively, the last three of which are unauthorized access. We now study how these unauthorized access terms influence the resistance of the restrictive DNN against attacks. In Table 2.7, we define four modes of training formulation in the ablation study, where "Mode D" is equivalent to our proposed training formulation. We apply the same attack processes to the models as stated in Section 2.4.5 and we utilize the whole training data to obtain the adversarial transform module in the "pair attack" case.

Mode	Authorized	Unauthorized Access		
		Direct Piracy	Brute Force	Pair Attack
A	✓	X	X	X
B	✓	✓	X	X
C	✓	✓	✓	X
D (Ours)	✓	✓	✓	✓

Table 2.7: Four modes of training formulation in ablation study

!![h]



(a) CNN on MNIST (b) CNN on CIFAR10 (c) Resnet-56 on CIFAR10 (d) Resnet-56 on CIFAR100

Figure 2.9: Ablation study on attack resistance of the restrictive DNN against unauthorized access with different modes of training formulation. "Mode D" is equivalent to our proposed formulation. The red dash lines indicate the accuracy of the corresponding baseline models.

Figure 2.9 shows the attack resistance of the restrictive DNN against unauthorized access with four modes of training formulation. "Mode A" does not consider authorized attack cases in the training and the restrictive models, compared with the authorized input, have no noticeable performance drop against attacks, suggesting that the "Mode A" models have no immunity to any type of unauthorized access defined in the threat model. When we incorporate the consideration of direct piracy, brute force, and pair attack in the training, the models gain the capability of attack resistance against the corresponding attack. Specifically, we observe that "Mode B" models reduce the performance for direct piracy; "Mode C" models maintain the poor performance for direct piracy and brute force; and the proposed "Mode D" models have noticeable performance gaps between the three types of unauthorized access and the authorized input, demonstrating good resistance of the restrictive DNN against attacks.

The ablation study reveals that incorporating the unauthorized access terms in the training formulation is effective in improving the defense ability against the corresponding attacks.

2.5 Discussions

Conceptually, the proposed work shares some connections with data obfuscation. The purpose of the data obfuscation is to prevent unauthorized access to sensitive data. In our framework, the obfuscated inputs trigger the mechanism of access control to the DNN model, and the main purpose is to protect the DNN

model from unauthorized access. Privacy protection of sensitive data is a secondary benefit in the proposed framework through data obfuscation.

2.5.1 Relation Between the Ordinary and the Restrictive DNN

Figure 2.10 illustrates the relation between the restrictive DNN and ordinary DNN. The conventional training of the DNN model often requires the DNN model to learn the features of the raw input. When we freeze the pretrained DNN model and add well-designed small perturbations to the raw inputs, these adversarial examples can fool the model by rendering the classification outcome deviated significantly from the ground truth. The ordinary DNN models are thus valid to the raw inputs but become ineffective to their adversarial examples.

In our proposed framework, we swap the roles of the raw input and adversarial examples in the DNN models. The restrictive DNN model learns the appropriate features of the perturbed inputs (*i.e.*, the authorized inputs), and the raw inputs effectively become the adversarial examples to the restrictive DNN model. The problem we need to solve here is given the adversarial examples, to find the equivalent "raw inputs" and their corresponding DNN model. Our proposed framework provides two feasible perturbation methods, *e.g.* additive perturbation and spatial perturbation, in the protective transform module to produce such an equivalent "raw input" and the restrictive DNN that can classify the perturbed input in high precision and the raw input in low accuracy.

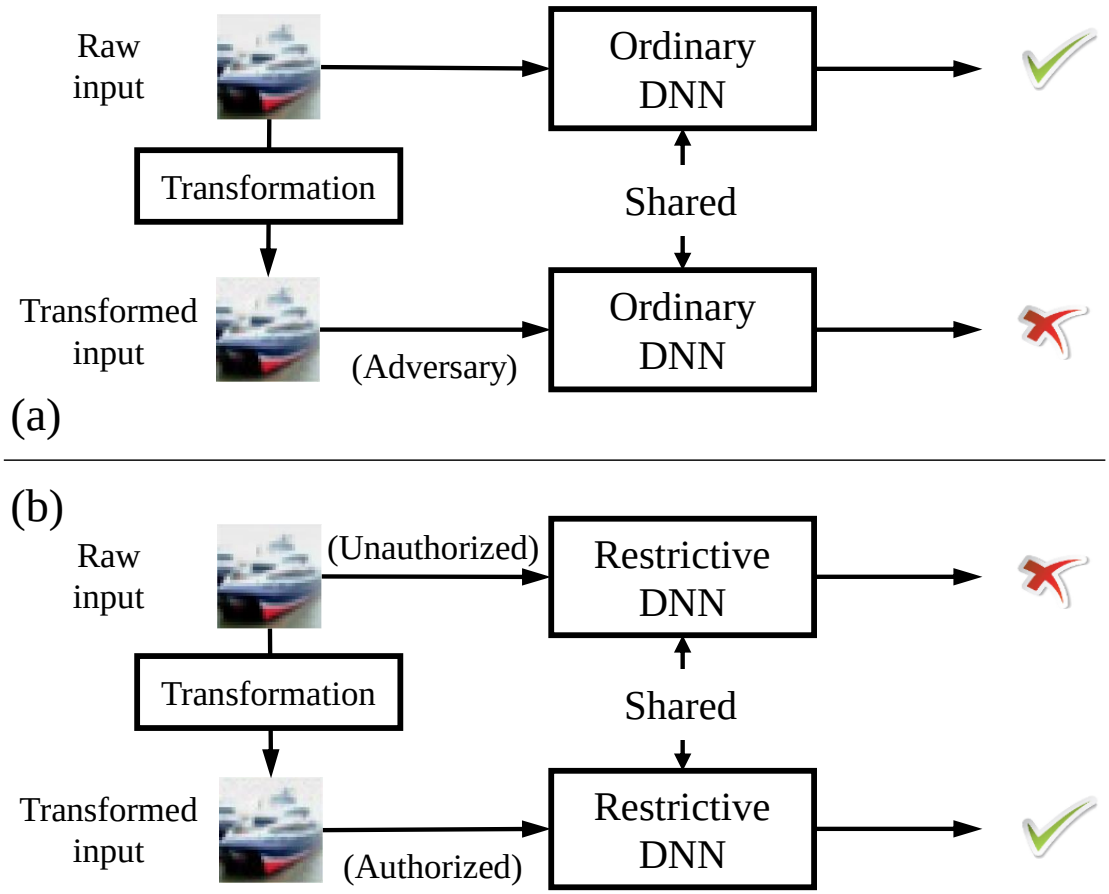


Figure 2.10: Illustration of the relationship between (a) the ordinary DNN and (b) the restrictive DNN in the proposed framework. The roles of the raw inputs and the transformed inputs are swapped in two DNN frameworks.

2.5.2 Privacy-aware Smart Camera Application

To provide a proof-of-concept demonstration of the proposed framework, we propose a practical application toward a privacy-aware smart camera. More specifically, we can incorporate the protective transform module into the camera, which becomes a smart camera. As illustrated in Figure 2.11, only the pictures taken by such the smart camera with the correct key k are valid inputs for the corresponding restrictive DNN and the result accuracy is high, while the photos from other

cameras or the smart camera with the wrong key $\tilde{k}$ will make the DNN work in a dysfunctional mode, so that no reliable prediction results would be produced.

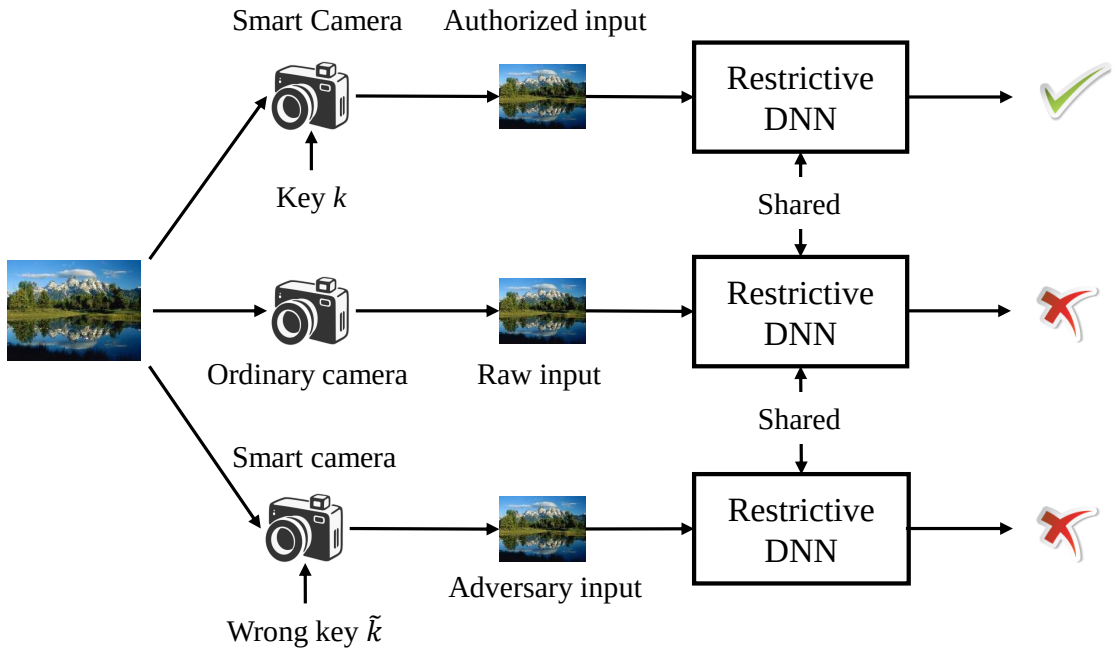


Figure 2.11: Illustration of the privacy-aware smart camera application where the protective transform module is embedded into a smart camera. Only when the correct key k is provided, the photos taken by a smart camera are proper inputs of the restrictive DNN. In contrast, the images taken by an ordinary camera or a smart camera with a wrong key $\tilde{k}$ result in the dysfunction of the restrictive DNN.

The advantage of this application is the difficulty for an adversary in the "pair attack" scenario to obtain the pairs of the raw input and the corresponding authorized input simultaneously, as the processing is encapsulated within the camera's embedded system and no intermediate signals are exported. As such, a user can only obtain either an authorized image from the smart camera with the correct key (which will lead to a prediction with high accuracy) or a regular image from other ordinary cameras or the smart camera with the incorrect key (which will lead to a prediction with low accuracy). Therefore, it enhances the difficulty in acquiring the pairs for the "pair attack" strategy, which is the most effective attack in

Section 2.4.5, and thus benefits the resistance to the privacy breach and attacks.

Due to the perceptual similarity of the raw and the authorized image in the proposed framework, a regular photo and the corresponding protective transformed photo have little distinguishable differences by the human vision system. Thus, the pictures taken from the smart camera retain their physical meanings and appearance, and can be considered, stored, circulated, and used as common images, similar to the digital media with digital watermarking.

2.6 Chapter Summary

In this chapter, we have introduced an access-control framework of utilizing the adversarial behavior of the DNN and turning it into a protection tool against illicit use and piracy. The access-control framework consists of a key-based protective transform module and the corresponding restrictive DNN. The restrictive DNN is only functional to the authorized users who can provide the correct key to the protective transform module. Any unauthorized access to the models will lead to classification results with low accuracy. We investigate additive perturbation and spatial perturbation as transformations to generate authorized inputs and evaluate their impacts on the access-control design. The experimental results show the effectiveness of the proposed framework to enable the access control of DNN models. Thanks to the adversarial terms in the training formulation, these models are resistant to several types of attacks. At last, we present a potential application towards building the system of privacy-aware smart cameras, which incorporates ordinary

commercial cameras and the proposed access-control framework. Such application will enhance the attack difficulty from an adversary perspective.

Chapter 3

Physiological Privacy Protection Against rPPG Technology

3.1 Overview

Remote photoplethysmography (rPPG) technology has attracted a growing amount of R&D interests, which is capable of capturing the subtle color changes of the skin caused by heartbeats in facial videos under ambient light. This emerging technology synergistically incorporates various computer vision algorithms to build up a sophisticated system to estimate a person's physiological status from a distance. Although rPPG technology facilitates daily health monitoring via the commercial color cameras in the phones, but it may cause concerns about physiological privacy in the meantime.

To address the above physiological privacy issue in rPPG technology, it is important to investigate how to effectively protect the physiological signals from disclosure in facial videos. To this end, we propose *PulseEdit* illustrated in Figure 3.1, a novel method that edits rPPG signals in facial videos by superimposing specifically designed perturbation of small amplitude onto the input videos. Our method outputs a video that is visually the same but has its rPPG signal either

removed completely or transformed to a target HR based on the user's choice. Processed by PulseEdit, the users' rPPG signals are protected from disclosure in the facial videos.

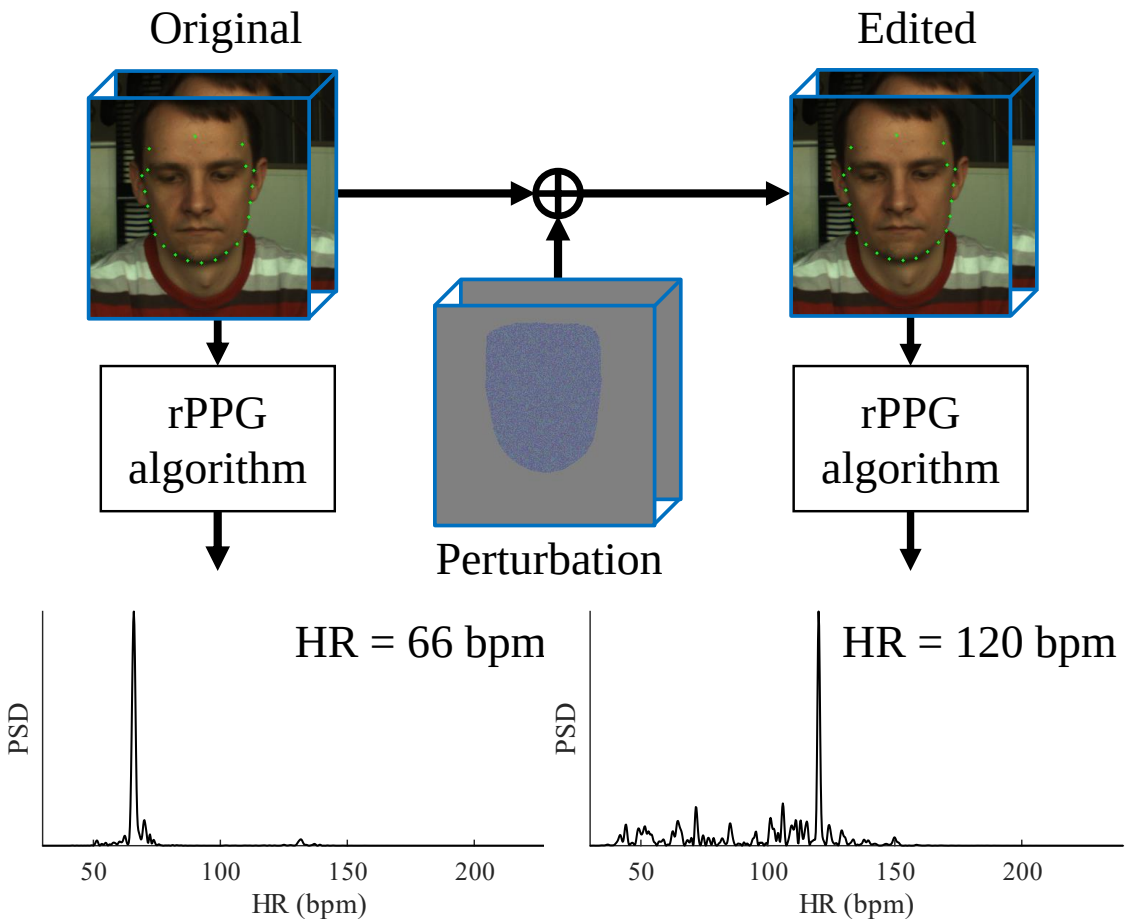


Figure 3.1: PulseEdit can edit rPPG signal in a facial video to conceal the person's true physiological status, without visual distortion of his/her appearance. We impose negligible additive perturbation onto the facial region in the video, and successfully modify the HR extracted by the rPPG algorithm. In this example, HR is edited from 66 to 120 beats per minute (bpm) to avoid the disclosure of the user's true heart rate in the video.

To make PulseEdit effective in practical use, we consider the following requirements when designing and evaluating the algorithm:

- **Invisibility:** the editing on the face should be negligible without obvious

appearance distortion.

- **Universality:** the protection should be valid on the face globally and locally. The processed video should no longer contain the user's true rPPG signal and the edited rPPG signal can be detected from the whole face, as well as local skin regions.

- **Generality:** the protection should be able to conceal a person's true rPPG signal against various rPPG algorithms in the literature. In other words, the edited rPPG signal can be measured by various rPPG algorithms.

- **Resistance:** an advanced requirement is that the editing on the face can be resistant to forensic analysis.

In addition to privacy protection, PulseEdit can impact other applications where rPPG is employed. More specifically, rPPG signal has been demonstrated as a useful and discriminative feature in various vision security tasks, such as liveness detection [27–29, 61] and deepfake detection [30], because real/live videos and fake/synthetic videos have different representations in rPPG signals extracted from the facial regions. Empowered by PulseEdit, we can edit the rPPG signals in facial videos and circumvent the above rPPG-based visual security algorithms. It is not difficult to see that PulseEdit is a potential threat to invalidate these algorithms, providing a direction to revise them and improve the confidence of their output decisions.

In the rest of the chapter, we first introduce the prior work related to rPPG technology and its application in security and privacy tasks in Section 3.2. Sec-

tion 3.3 describes the proposed *PulseEdit* to edit pulse signal in facial videos. We carry out comprehensive performance analysis on the PulseEdit algorithm for removing/modifying pulse signal in facial videos in Section 3.4, and explore the feasibility of the algorithm as a potential adversary against rPPG-based liveness detection and deepfake detection algorithms in Section 3.5. Finally, Section 3.6 holds a related discussion and Section 3.7 summarizes the chapter.

3.2 Related Work

3.2.1 rPPG Technology

Monitoring cardiac activity is essential for understanding a person's health status and is actively used in clinical practices and home care. Conventional methods require contact-based sensors attached to the human skin, such as electrocardiogram leads, a pulse oximeter, or a fitness tracker.

Recently, rPPG enables contact-free HR measurement using color cameras. The principle of rPPG is that the blood volume changes under the skin influence the intensity and color of the reflected light from the skin, whose pattern is consistent with heartbeat cycles. Although such subtle momentary changes in the reflected light from the facial skin are not detectable by the human eyes, they can be captured by a color camera [12]. The method of Eulerian video magnification [62] can amplify and visualize the subtle color changes in a facial video caused by the blood flow. Independent component analysis (ICA) [24], chrominance mapping (CHROM) [13], and plane-orthogonal-to-skin (POS) [15] were proposed to extract robust rPPG fea-

tures from three color channels. Li *et al.* [16] applied adaptive filtering to handle environmental illumination and voluntary motion issues in remote HR measurement. Tulyakov *et al.* [17] proposed self-adaptive matrix completion to denoise rPPG features and offer robust HR estimation. The challenging fitness scenario [63, 64] has also been studied to improve the robustness of the rPPG technology. End-to-end models [18, 20] employing deep learning were also introduced to estimate HR from videos.

3.2.2 Biometric Privacy Protection

Biometric privacy protection [65, 66] aims to conceal a person's privacy in biometric data and prevent possible thefts and misuses of this information. Traditional biometric privacy protection algorithms were proposed to de-identify person's identity from these biometric features, including face [67, 68], iris [69], and fingerprint [70]. Deep learning has been introduced to protect privacy in multimodal biometrics [71].

In spite of privacy protection at the image perception level, several researches studied the privacy protection approaches at the feature representation level. Several facial representation methods were proposed to eliminate facial expressions [72] or selected biometric attributes (*e.g.*, age and gender) [73] in facial feature level. SensitiveNets [74] generates a learned embedding space that eliminates specific sensitive biometric information from the existing representation subspace. Terhorst *et al.* [75] proposed a privacy-preserving solution to suppress biometric attributes in an

unsupervised manner. The privacy-preserving feature representations can improve the robustness of training models and benefit the fairness in model inference across biometric attributes.

Recently, as many methods have been proposed in the recent decade to extract physiological signals from facial videos, concerns are raised concurrently on the privacy issues of physiological information in videos. This information may be misused to collect and analyze a person's physiological features with ulterior motives. Chen *et al.* [76] applied motion elimination in facial videos to remove subtle motion induced by pulse on the subjects' faces to avoid the disclosure of the rPPG signal. The experimental results show that the rPPG signals are successfully removed without appearance distortion. Nevertheless, the work only studied the steady case in the research. When the subject performs voluntary motion (*e.g.,* talking, head translation, and rotation) in video recording, it is hard for Chen's method to remove pulse-induced subtle motion but maintain the subject's voluntary motion.

In this chapter, we propose to edit the rPPG signals that are intrinsically presented in facial videos by perturbing the skin pixels on the face and conduct experiments on motion case as well as steady case. Compared with the prior art, not only is our work capable of removing rPPG signal in a facial video, but also transforming it to a target one if desired by the user.

3.2.3 rPPG Feature in Visual Security Tasks

rPPG signal has been employed as a discriminative feature to tackle several visual security tasks involving face videos, such as liveness detection against spoofing and deepfake detection. Liveness detection is crucial to protect face recognition systems from spoofing attacks, including printing a face on paper, replaying a facial video on a digital device, wearing a 3D face mask, and other approaches by adversaries. Liu *et al.* [27, 28] used the cross-correlation of rPPG features in multiple facial regions to classify live faces vs. spoofed faces. Hernandez *et al.* [29] proposed to analyze the signal quality of rPPG extracted from faces to discriminate live faces and spoofed faces. Liu *et al.* [61] used a CNN-RNN architecture to learn a mapping from a facial video to the rPPG signal to improve the robustness of liveness detection.

"Deepfake" refers to the technologies for a computer to transform a person's face to another's in images and videos. Since deepfake videos circulated in social media have brought serious concerns such as through celebrity pornographic videos, fake news, hoaxes, and financial fraud, which largely impairs the integrity of social media, deepfake detection has attracted a lot of attention in the recent computer vision research. In terms of the roles of rPPG for deepfake detection, FakeCatcher [30] explored the discriminative features of rPPG signals extracted from facial videos and utilized them for deepfake detection.

3.3 Conceal rPPG Signals in Facial Videos

PulseEdit has three main steps as shown in Figure 3.2. We first detect the facial region in the video and extract skin intensity signals from multiple subregions on the face. We then obtain the perturbation signal via an optimization problem that transforms the rPPG signal in the video to a target signal. Finally, we manipulate the skin pixels in the video according to the perturbation signal, so that the PulseEdit video successfully removes the rPPG signal, or if desired, transforms the rPPG signal to a target rPPG signal. We refer to the two tasks as "removal" and "modification", respectively, for short. In the removal task, the target signal can be white Gaussian noise; and in the modification task, the target signal can be a simulated sinusoid with the frequency of a target HR or the rPPG signal extracted from a reference video of the user's choice.

3.3.1 rPPG Extraction

Similar to the prior art in rPPG research, we first track the subject's face in the video to extract rPPG signal. We apply the facial landmark detector by Dlib [77] to locate and track 68 facial landmarks, from which we define the facial region of interest (ROI) shown with the green dots in the video frame in Figure 3.2. To facilitate rPPG extraction from multiple subregions [17], the ROI is normalized to a rectangle using piecewise linear geometric transformation, and skin color pixels

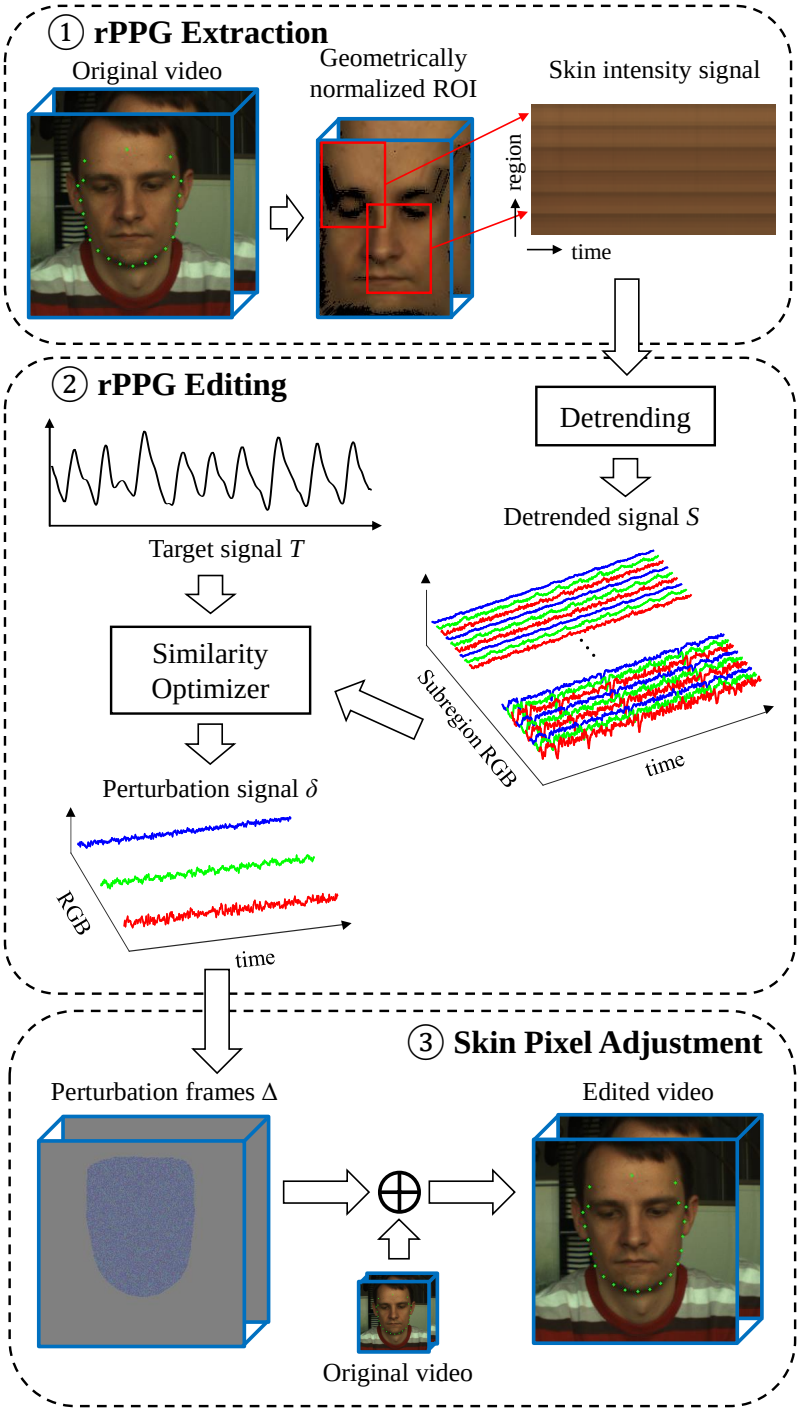


Figure 3.2: Pipeline of PulseEdit system. We first extract skin intensity signals from multiple facial subregions in the video. Then, we compute the perturbation signal that can change the rPPG signals in the video to the target rPPG signal. Finally, we edit the skin pixels in the video, and the rPPG signal extracted from the video processed by PulseEdit is successfully transformed to the target signal.

are masked by a Gaussian skin color model in chrominance space [78]:

$$p(x) = \exp\big(-\frac{1}{2}(x-m)^T\Sigma^{-1}(x-m)\big) \underset{\text{non-skin}}{\overset{\text{skin}}{\gtrless}} p_t, \tag{3.1}$$

where $x = [cb, cr]^T$, the empirically determined parameters of $m = [104.50, 156.93]^T$, and $\Sigma = [77.69, -51.47; -51.47, 81.93]$. Within the masked rectangle facial ROI, we use a rectangle of a quarter size to uniformly select M subregions (subregions can have overlap with their neighbors). We compute the spatial average of the skin pixels in each subregion to form the skin intensity signal $R \in \mathbb{R}^{M\times 3\times N}$, for M subregions, 3 color channels, and N frames in the video. In the subsequent discussions, we refer to the subscripts i and c as subregion i and color channel c, respectively. For example, $R_{i,c}$ denotes the skin intensity signal in subregion i and color channel c.

3.3.2 rPPG Editing

In this module, our goal is to find a suitable perturbation on the skin intensity signals to change the rPPG in videos to the target signal given by users. We first detrend the skin intensity signal $R_{i,c}, \forall i, c$, to eliminate the illumination interference in the environment. In the detrending process, we use l_1 trend filtering [79] to obtain the signal trend and subtract the trend from the skin intensity signal. The whole process can be described as

$$\min_{S_{i,c}} \quad \frac{1}{2}||S_{i,c}||_2^2 + \mu||D(R_{i,c} - S_{i,c})||_1, \forall i, c, \tag{3.2}$$

where $S \in \mathbb{R}^{M \times 3 \times N}$ denotes the corresponding detrended signal, the subscripts i and c denotes the subregion and the color channel, and $D \in \mathbb{R}^{(N-2) \times N}$ is the second-order difference matrix

$$D = \begin{bmatrix} -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & \text{\Large 0} & \\ & & \ddots & \ddots & \ddots & & \\ & \text{\Large 0} & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \end{bmatrix}. \tag{3.3}$$

We denote $\delta \in \mathbb{R}^{3 \times N}$ as the additive perturbation imposed onto the detrended skin intensity signal S, which gives rise to the edited signal $\tilde{S} \in \mathbb{R}^{M \times 3 \times N}$, *i.e.* $\tilde{S}_{i,c} = S_{i,c} + \delta_c, \forall i, c$, where δ_c denotes the perturbation in the color channel c.

Next, we set up the target rPPG signal $T \in \mathbb{R}^{3 \times N}$. To ensure the output video contains the target rPPG signal T, we maximize the similarity between the edited signals $\tilde{S}$ and the target signal T using the Pearson correlation coefficient:

$$\mathrm{P} = \frac{1}{M} \sum_{i,c} \rho(\tilde{S}_{i,c}, T_c) = \frac{1}{M} \sum_{i,c} \rho(S_{i,c} + \delta_c, T_c). \tag{3.4}$$

For an edited facial video, we require that the person in the video has negligible perceptual distortion. Thus, we regularize the perturbation signal δ with L_2 loss to control the perturbation budget in the facial video:

$$E = \frac{1}{N} ||\delta||_2^2. \tag{3.5}$$

Combining the above two terms, we obtain the perturbation signal δ by solving the optimization problem:

$$\min_{\delta} \quad -\frac{1}{M}\sum_{i,c}\rho(S_{i,c}+\delta_c, T_c) + \lambda\frac{1}{N}||\delta||_2^2. \tag{3.6}$$

We can use gradient-based solver (*e.g.*, Adam [80]) to solve the optimization problem in (3.6).

3.3.3 Skin Pixel Adjustment

The goal of this module is to map the perturbation signal $\delta \in \mathbb{R}^{3\times N}$ in time series to the spatial-temporal perturbation frames $\Delta \in \mathbb{R}^{h\times w\times 3\times N}$, where h and w refers to the height and width of the frames in pixel count. We denote $\delta_c(n)$ as the perturbation of the color channel c in the n-th frame. One simple and intuitive approach to edit the pixels on the face is to directly add $\delta(n)$ to every skin pixel on the facial region in the n-th frame of the input video. Due to the integer quantization of pixel values in video frames, the decimal part of $\delta(n)$ needs special consideration in order to ensure the pixel values collectively are changed by the expected amount.

We adopt randomized dithering to skin pixels to achieve decimal perturbation in a statistical sense. Specifically, for the color channel c in the n-th frame, we adjust the skin pixels in an amount of either $\lfloor\delta_c(n)\rfloor$ with probability p or $\lceil\delta_c(n)\rceil$ with probability $1-p$, where p should be chosen so that

$$\delta_c(n) = \lfloor\delta_c(n)\rfloor p + \lceil\delta_c(n)\rceil(1-p). \tag{3.7}$$

Algorithm 1: Skin Pixel Adjustment

Input: Original video $\mathcal{I}$ containing N frames, $\mathcal{I} = [\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N]$, (frame dimension $h \times w \times 3$); perturbation signal δ (dimension $3 \times N$).
Output: PulseEdit video $\tilde{\mathcal{I}}$.

```
Δ ← ZeroLike(I)                                  ▷ memory allocation
for n = 1 → N do
    R_face ← FaceSkinPixel(I_n)                  ▷ detect skin pixels
    for c = 1 → 3 do                             ▷ each color channel
        for all (x, y) ∈ R_face do               ▷ each skin pixel
            p ← Rand(0, 1)
            if p < ⌈δ_c(n)⌉ − δ_c(n) then
                Δ_n(x, y, c) ← ⌊δ_c(n)⌋
            else
                Δ_n(x, y, c) ← ⌈δ_c(n)⌉
            end if
        end for
    end for
end for
Ĩ ← I + Δ
```

Equation (3.7) yields $p = \lceil \delta_c(n) \rceil - \delta_c(n)$. Algorithm 3.3.3 presents the detailed procedure of skin pixel adjustment to generate the final PulseEdit video.

3.4 Experimental Results

In this section, we present experimental results on the PURE dataset [81] to demonstrate the effectiveness and robustness of PulseEdit in editing rPPG signals in facial videos. To further validate the forensic undetectability of PulseEdit when being used as a potential attack, we test the PulseEdit videos against digital forensic analysis. Lastly, we compare PulseEdit with the prior art of rPPG removal method [76] and study the influence of different subject motion settings in video recordings on the performance of rPPG removal.

In our experiments, we set $M = 6 \times 6 = 36$ and use Adam [80] to solve (3.6) with the learning rate 0.1 and the number of iterations 200.

3.4.1 Performance on PURE dataset

The PURE dataset [81] contains 60 facial video recordings of 640×480 pixel resolution and 30 frames per second (fps) in well-lit rooms from 10 subjects. Each subject was recorded in 6 different setups: steady, talking, slow translation, fast translation, small rotation, and medium rotation. The videos were stored without lossy compression. To validate the effectiveness of PulseEdit in editing rPPG signals in facial videos, we analyzed the PulseEdit outputs of the PURE videos with five representative rPPG algorithms: ICA [24], CHROM [13], POS [15], HR-CNN [18], and DeeprPPG [82]. The first three methods are classical signal processing methods and the last two are deep learning methods. We extracted rPPG signal from the whole facial region in this part of the experiments to estimate HR, and evaluated the performance using mean absolute error (MAE). For the rPPG removal task, we computed the error between the estimated HR from the video and the reference HR from pulse oximeter provided by the dataset. For the rPPG modification task, we computed the error between the estimated HR from the video and the target HR.

We applied PulseEdit on the PURE videos for both the removal task and the modification task. In the removal task, we generated white Gaussian noise as the target rPPG signal T to remove the intrinsic rPPG signal in the original video. In the modification task, we aimed at changing the rPPG signal to HR = 120 bpm

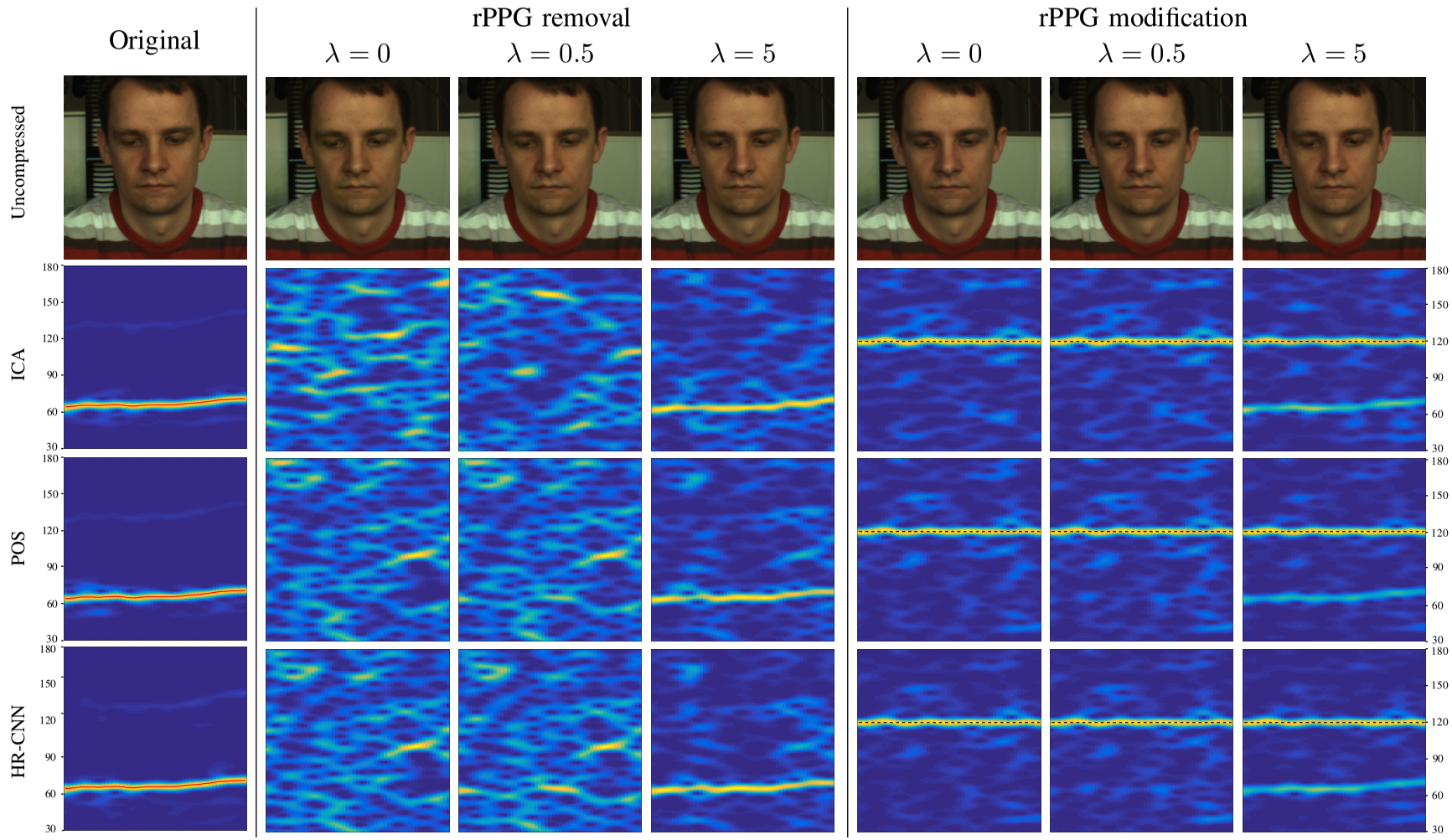


Figure 3.3: Exemplary face crop from the videos and spectrograms of the rPPG extracted from the uncompressed videos with two classical rPPG methods, ICA and POS, and one deep learning method, HR-CNN. We set the target HR = 120 bpm in rPPG modification task. The red lines in the spectrograms of the original video indicate the reference HR from pulse oximeter and the black dash lines in the spectrograms of the rPPG modified videos indicate the target HR = 120 bpm.

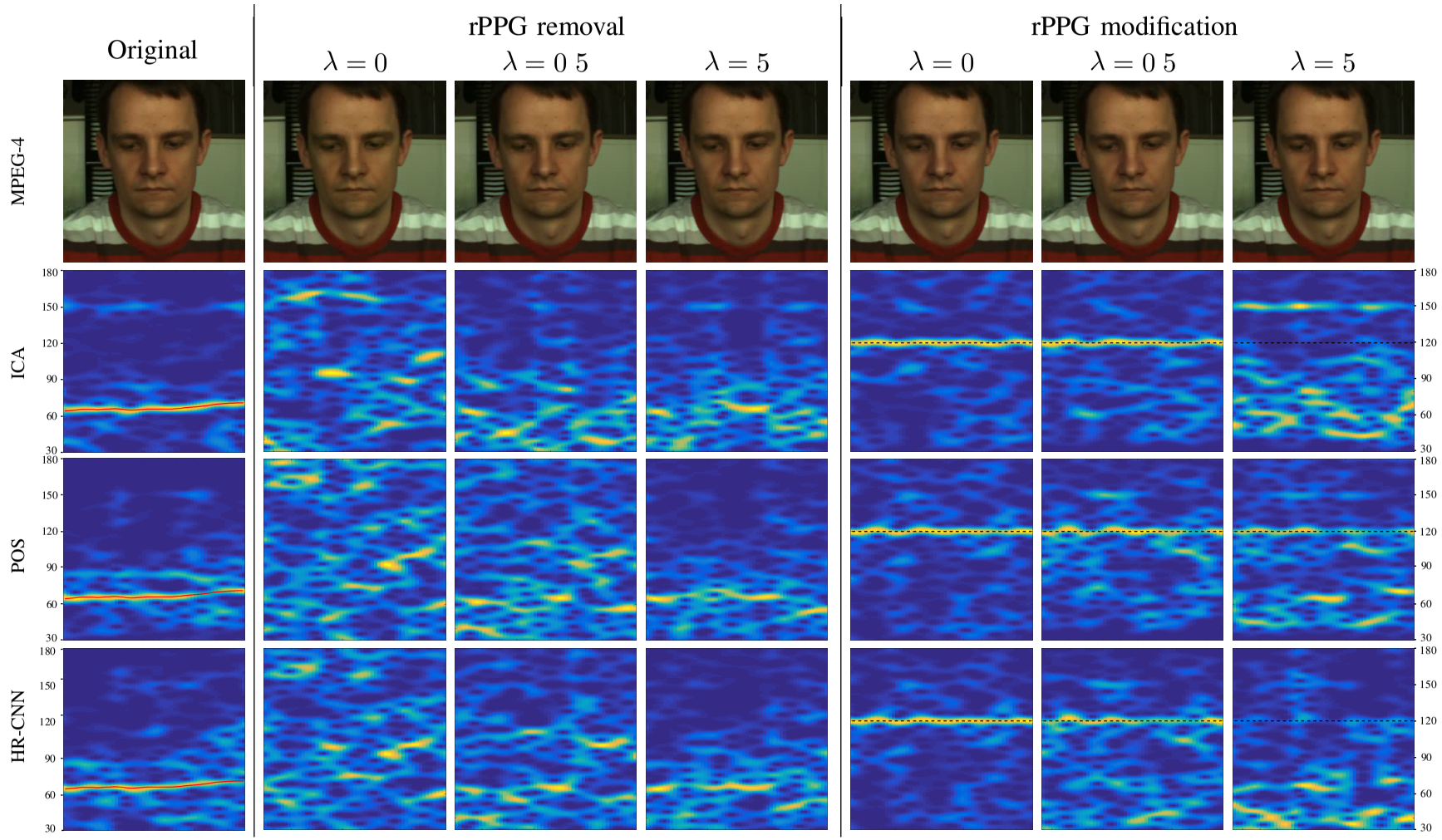
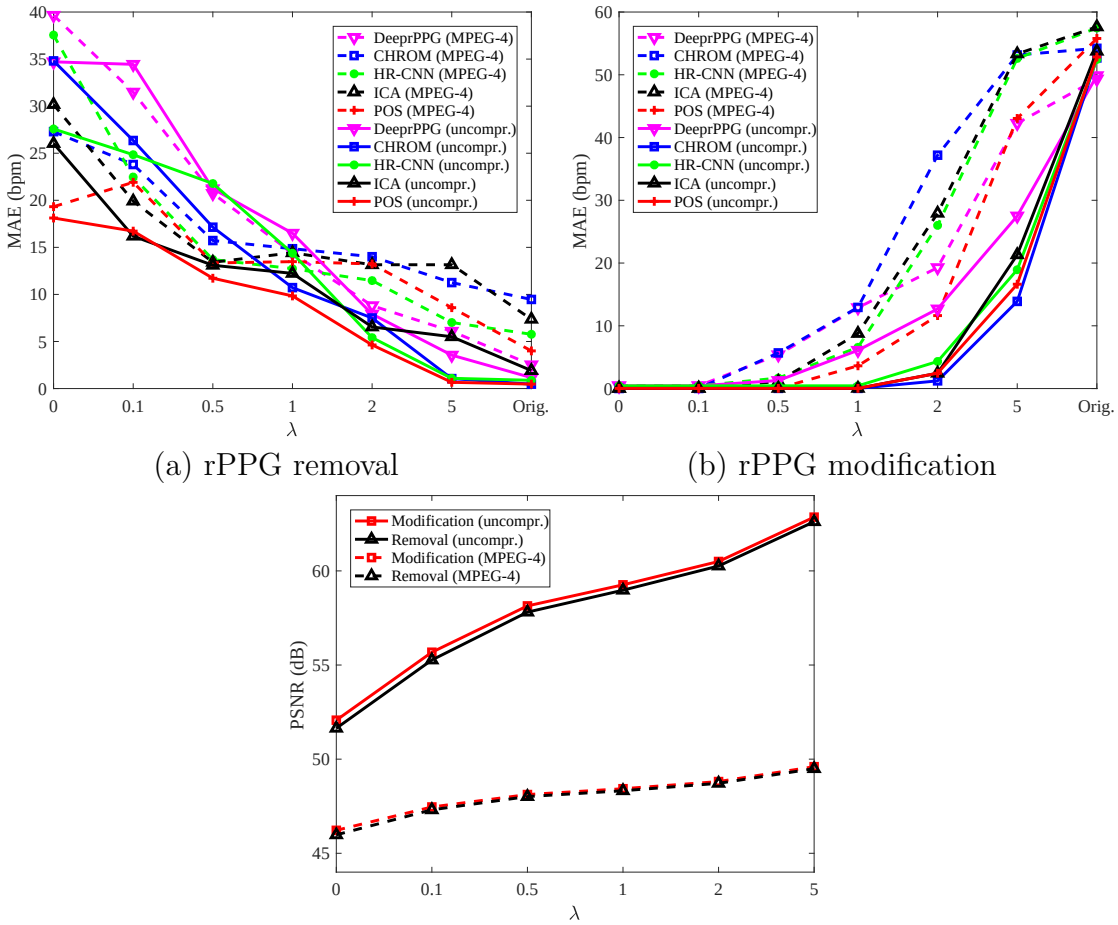


Figure 3.4: Exemplary face crop from the videos and spectrograms of the rPPG extracted from the lossy compressed videos with two classical rPPG methods, ICA and POS, and one deep learning method, HR-CNN. We set the target HR = 120 bpm in rPPG modification task. The red lines in the spectrograms of the original video indicate the reference HR from pulse oximeter and the black dash lines in the spectrograms of the rPPG modified videos indicate the target HR = 120 bpm.

as an example. We generated a sinusoid of frequency 120 bpm as the target rPPG signal T for all the color channels. To simulate the noise condition of rPPG signals, we added white Gaussian noise with -10 dB, -0 dB, and -10 dB in red, green, and blue channels, respectively, since the green channel generally contains the strongest level of PPG signal among all three channels [12]. We used the whole face region in rPPG analysis to estimate HR from facial videos.

We study the effect of different $\lambda = \{0, 0.1, 0.5, 1, 2, 5\}$ on the performance of PulseEdit, which governs the perturbation budget in the facial video. To investigate the robustness of PulseEdit against video lossy compression, we compressed the edited frames by MPEG-4 format at the average bitrate of around 500 kbps. Figure 3.3 and 3.4 show the qualitative comparison of the video frames and the corresponding rPPG spectrograms with different λ in lossy uncompressed and compressed mode. Figure 3.5(a) and (b) present the performance of HR estimation before and after PulseEdit in removal and modification task, respectively.

In the removal task, we aim to increase the error of HR estimation with respect to reference HR, and Figure 3.5(a) shows that the error increases as λ decreases. When λ is less than 0.5, the rPPG-removed videos have a very large estimation error (*i.e.*, $>$ 10 bpm), indicating the successful removal of the intrinsic rPPG signal by PulseEdit. In the modification task, our goal is to reduce the error of HR estimation with respect to target HR, and Figure 3.5(b) shows that the error is reduced as λ decreases. When λ is less than 0.5, the rPPG-modified videos have HR estimations very close to the target HR, with an error no more than 1 bpm for uncompressed videos and 10 bpm for MPEG-4 videos. This suggests that PulseEdit can effectively



(a) rPPG removal

(b) rPPG modification

(c) Frame-level PSNR

Figure 3.5: Performance of PulseEdit on the PURE dataset with different λ: (a) HR estimation error in the rPPG removal mode, (b) HR estimation error in the rPPG modification mode, and (c) average frame-level PSNR.

transform the rPPG signal in a video to a target HR. From Figure 3.3 and 3.4, we observe that when λ increases from 0 to 5, the original rPPG signals gradually appear in the spectrograms of the edited videos. This indicates that we need to spend enough editing expense (smaller λ) in the video to successfully conceal the original rPPG signal.

Since lossy compression may attenuate the rPPG signal on the face, it is expected that the HR error is larger in MPEG-4 videos than in uncompressed videos. Specifically, in the rPPG modification task, the HR error with respect to target HR

is larger in the MPEG-4 video than in the uncompressed one. Nevertheless, the modified rPPG signal of target HR can still be detected by the rPPG methods within an acceptable error range, when we choose $\lambda < 0.5$. In comparison, lossy compression has less impact on the rPPG removal task. On the whole, these results indicate that although lossy compression can weaken the manipulations introduced by PulseEdit, the privacy protection of the intrinsic rPPG signal remains effective when choosing a proper λ.

An important observation is that the five rPPG methods have similar HR estimation performance on the PulseEdit videos, indicating that PulseEdit is effective to various rPPG algorithms, including the classical signal processing methods and the deep learning methods as well. This satisfies the "generality" requirement.

Figure 3.6: Illustration of the three facial subregions: forehead (red), left cheek (green), and right cheek (blue). The regions are detected automatically according to the facial landmarks.

Figure 3.5(c) shows the objective image quality assessment for the PulseEdit videos within the facial ROI with a size of 300×300. Since λ governs the editing

strength in the video, frame-level PSNR increases when λ increases. By vision examination, we can hardly notice the distortion on the person's appearance shown in Figure 3.3 and 3.4.

Running time. Overall, PulseEdit runs efficiently. On average, the step of rPPG extraction runs at around 10 fps, the step of rPPG editing reaches 170 fps, and the step of skin pixel adjustment runs at around 100 fps. These running times were measured using a single-core Python implementation on a PC with an Intel Core i5-4440 processor. As the step of rPPG extraction is highly dependent on the speed of facial landmark detection, the running time can be further reduced if facial landmark detection is optimized.

3.4.2 rPPG Analysis on Multiple Facial Subregions

To examine the universality of PulseEdit, we analyze the presence of rPPG signals in three facial subregions: forehead, and left and right cheek (shown in Figure 3.6). The regions are detected automatically according to the facial landmarks. Figure 3.7 presents the performance of HR estimation from three facial subregions using five rPPG algorithms. For classical non-deep learning methods, we apply the algorithms within the selected subregions; for deep learning methods, we first warp the polygon regions to regular rectangles with the fitting input size, and then feed them into the models. Since a larger size of ROI generally gives a better average quality of rPPG extraction [83], we expect a reduced accuracy of HR estimation from facial subregions alone, compared with using the whole face region.

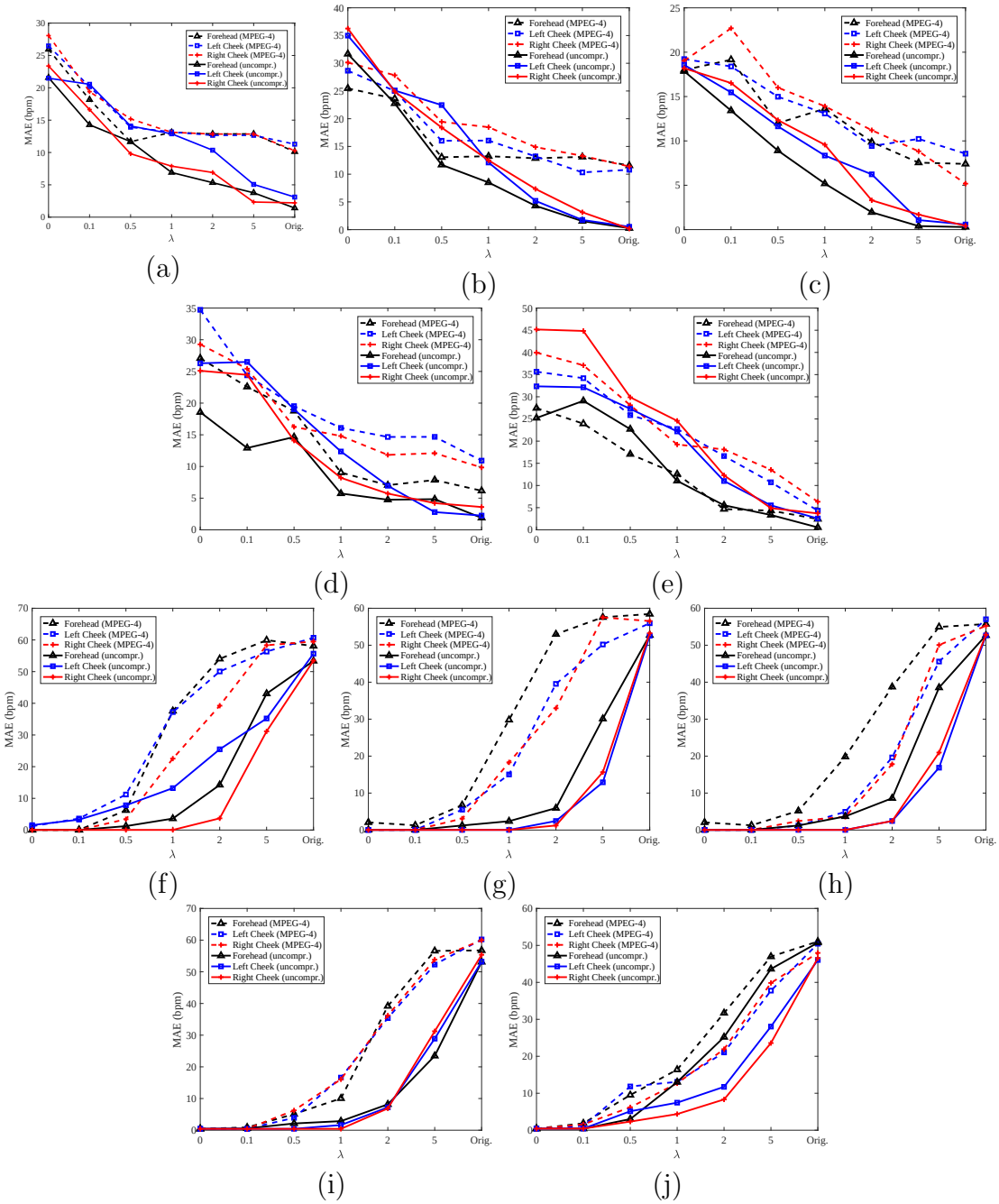


Figure 3.7: HR estimation of PulseEdit videos on multiple face subregions with different λ using five rPPG methods: (a) ICA, (b) CHROM, (c) POS, (d) HR-CNN, (e) DeeprPPG in the rPPG removal mode, and (f) ICA, (g) CHROM, (h) POS, (i) HR-CNN, (j) DeeprPPG in the rPPG modification mode.

From Figure 3.7, we observe that HR error from the three facial subregions has a similar trend as that from the whole face region under different λ values. For the

rPPG-removed videos, the error is much larger than the original videos, when λ is less than 0.5. This suggests that the intrinsic rPPG signals are completely erased in all three facial subregions. For the rPPG-modified videos, the HR error with respect to the target HR is in an acceptable range, when λ is less than 0.5. We can see that the rPPG signals in all three facial subregions are successfully transformed to the target HR. In summary, these results indicate that PulseEdit can effectively edit the rPPG signals not only in global facial region, but also in local facial subregions, which satisfies the "universality" requirement.

PulseEdit computes the original skin intensity variations $R \in \mathbb{R}^{M \times 3 \times N}$ (M denotes the number of subregions) from multiple facial subregions and finds the optimal perturbation in (3.6) that can change the heartbeat information in all the local regions. This design can help the perturbation universally change the heartbeat information in the global face and the local facial regions.

3.4.3 PulseEdit Against Forensic Analysis

From the previous performance analysis on PulseEdit, we can see that PulseEdit is effective in editing the intrinsic rPPG signals in facial videos for privacy protection. From this point of view, it is possible to utilize PulseEdit in adversarial scenarios by forgers. In this subsection, we examine the effect of forensic analysis tools to help us understand the strengths and limitations of PulseEdit.

PulseEdit perturbs the skin pixels by a small amount in the video frames to edit rPPG signals, which is similar to how steganography manipulates the images. Based

on this point of view, we examine the forensic detectability of PulseEdit against two representative steganalysis methods: spatio–color rich model (SCRM) [84] with ensemble training [85], and WISERNet [86] based on deep learning. Since PulseEdit only edits the facial regions, we cropped facial ROI with a size of 300×300. We set the original video frames as negative and the PulseEdit video frames as positive, and used 5-fold cross-validation to evaluate the performance. For deep models, we changed the size of feature maps in the intermediate layers accordingly to cater to the input size of 300×300.

We observe that the steganalysis models are most effective on uncompressed video frames as their detection performance has an area under curve (AUC) of 0.99+ for every λ value. They can almost perfectly differentiate the original video frames and the edited video frames by PulseEdit. Without incorporating additional constraints, the randomized pixel adjustment in Section 3.3.3 perturbs the skin pixels independently in the frame, introducing artificial changes among local neighboring pixels that are not presented in the direct output of video cameras. This kind of unconstrained distortion can be easily extracted by various image forensic models and discriminative to natural images and edited images [87–89]. Figure 3.8 presents the steganalysis results on the lossily compressed videos. Compared with the uncompressed videos, the steganalysis result of the MPEG-4 videos degrades in a noticeable amount. For the two steganalysis models, the deep model has a better ability to detect the manipulation trace in the lossily compressed videos than the classic model. We also find that the steganalysis performance on the lossily compressed videos decreases significantly in both forensic methods as λ increases.This

suggests that lossy compression can alleviate the detectability of the manipulation traces in videos introduced by PulseEdit.

In the current form, PulseEdit focuses on altering the rPPG information for privacy protection and has not explicitly concealed the traces of manipulation. As such, the presence of perturbation can be detected by forensic tools such as steganalysis from the uncompressed frames. Because of the limitation of such forensic analysis for lossy compressed frames and the small and random perturbation of PulseEdit by design, a lossy compression on PulseEdit videos can evade forensic steganalysis and remain effective in concealing/modifying the intrinsic rPPG information. As part of future work, we plan to investigate the inclusion of various forensic undetectability into the algorithm, to gain insights on the ability of PulseEdit as an antiforensic tool and the competing direction of detecting the manipulations made by PulseEdit.

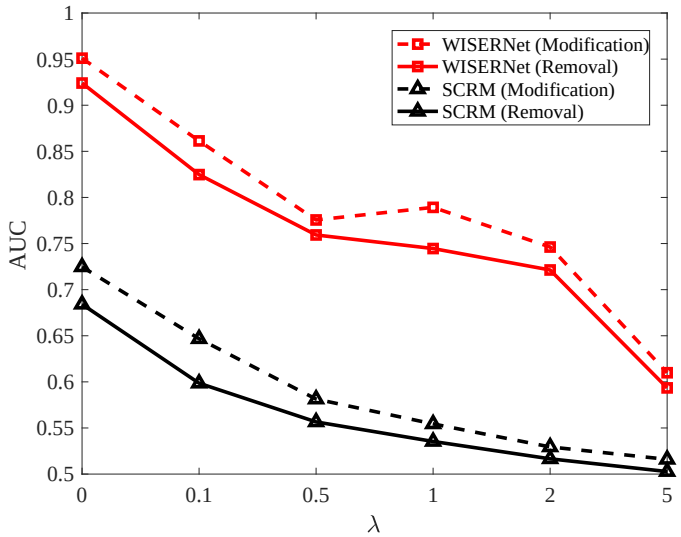


Figure 3.8: Performance of PulseEdit against forensic analysis in MPEG-4 videos.

			rPPG removal				rPPG modification			
			Uncompressed		MPEG-4		Uncompressed		MPEG-4	
			Original	Edited	Original	Edited	Original	Edited	Original	Edited
MAE of HR estimation (bpm)	ICA	Face	1.91	13.70	7.36	13.41	53.75	0.03	57.63	1.12
		Forehead	1.41	11.68	10.12	11.67	53.36	1.23	58.09	6.23
		Left cheek	3.11	14.09	11.28	13.91	55.67	7.81	60.70	11.21
		Right cheek	2.21	9.80	10.28	15.19	53.47	0.03	59.50	3.41
	CHROM	Face	0.47	17.14	9.47	15.72	52.68	0.03	54.22	5.68
		Forehead	0.26	11.69	11.55	13.05	52.71	1.22	58.44	6.75
		Left cheek	0.52	22.46	10.80	16.03	52.57	0.03	55.95	5.51
		Right cheek	0.18	18.40	11.39	19.44	53.24	0.03	56.53	3.09
	POS	Face	0.47	12.33	4.00	13.34	52.84	0.03	55.79	0.03
		Forehead	0.29	8.91	7.41	12.01	52.67	1.22	55.73	5.18
		Left cheek	0.58	11.64	8.56	14.98	52.64	0.03	57.02	1.17
		Right cheek	0.42	12.35	5.18	16.01	52.84	0.03	55.23	2.44
	HR-CNN	Face	0.88	21.78	5.77	13.70	57.46	0.44	57.46	1.68
		Forehead	1.89	14.65	6.17	18.76	53.11	5.09	56.76	6.75
		Left cheek	2.25	19.02	10.93	19.53	60.22	0.44	55.95	3.91
		Right cheek	3.60	14.09	9.84	16.25	55.38	0.43	59.99	6.25
	DeeprPPG	Face	1.16	21.23	2.52	20.73	49.86	1.28	49.33	5.38
		Forehead	0.55	22.70	2.44	17.06	50.74	3.01	51.06	9.52
		Left cheek	2.55	27.28	4.33	25.87	46.06	5.13	50.45	11.84
		Right cheek	3.70	29.88	6.33	28.09	46.57	2.33	47.96	6.19
PSNR (dB) (Orig. as ref.)			ref.	57.81	ref.	48.01	ref.	58.14	ref.	48.12
PSNR (dB) (Uncompr. orig. as ref.)			ref.	57.81	33.02	33.04	ref.	58.14	33.02	33.05
SCRM [84] (AUC)			n/a	1.00	n/a	0.58	n/a	1.00	n/a	0.56
WISERNet [86] (AUC)			n/a	1.00	n/a	0.78	n/a	1.00	n/a	0.76

Table 3.1: Result of PulseEdit with $\lambda = 0.5$ on HR error, perceptual distortion, and forensic analysis

3.4.4 Performance Summary of PulseEdit

Taking into consideration HR estimation error in PulseEdit videos, perceptual distortion, and resistance against forensics, we think that $\lambda = 0.5$ is a good choice for PulseEdit algorithm and we use it for the following experiments. We summarize the experimental results of PulseEdit with $\lambda = 0.5$ in Table 3.1. The first five macro-rows show MAE of HR estimation (unit: bpm), using different rPPG algorithms. Note that, we compute MAE between the estimated HR and the reference HR from pulse oximeter in rPPG removal task, and compute MAE between the estimated HR and the target HR= 120 bpm in rPPG modification task. The next row shows frame-level perceptual distortion analysis within the facial ROI between original videos and edited videos. The last two rows present the forensic analysis on PulseEdit.

Table 3.1 shows that the error of HR estimation with respect to reference HR from the facial video increases after PulseEdit in rPPG removal task; the error of HR estimation with respect to target HR decreases significantly after PulseEdit in rPPG modification task. This indicates that the proposed PulseEdit can effectively remove/modify rPPG information both in the whole face sense and in the local subregion sense, tested by various rPPG methods. High PSNR index suggests that PulseEdit hardly introduces perceptual distortion on the subject's appearance. Comparing the HR estimation error between the uncompressed and MPEG-4 videos, we can see that lossy compression can weaken the manipulation applied in the facial videos, but PulseEdit can still edit the rPPG signals to some extent. From the angle of antiforensics, the AUC index reduces more than 0.4 in the SCRM and more

than 0.2 in the WISERNet. This indicates that lossy compression can greatly help PulseEdit videos defend forensic analysis.

3.4.5 Comparison with Prior Art

We compare the proposed PulseEdit of removal task with the prior art Chen's method [76]. We report the estimated HR error from the facial videos using the five rPPG methods: ICA, CHROM, POS, HR-CNN, and DeeprPPG. The performance is evaluated on the uncompressed videos. Figure 3.9 presents bar plots of performance comparison between the proposed PulseEdit and Chen's method. We study the influence of 6 motion settings on the rPPG-removing methods: steady, talking, slow translation, fast translation, small rotation, and medium rotation.

From Table 3.2, we can see that the five rPPG methods can accurately estimate HR from the original videos. Given the fact that we can extract rPPG signals accurately from the original videos, Chen's methods and PulseEdit, in average performance, can both amplify the HR estimation error and our method can remove the intrinsic rPPG signal more completely, leading to larger amplification of HR error. PSNR index indicates that the proposed PulseEdit has less distortion than Chen's method on the video frames. When we analyze different motion settings in the video recordings from Figure 3.9, we can observe that Chen's method has similar performance to our method in steady case, but does not perform well in talking, head translation, and head rotation cases. This suggests that Chen's method is not effective when dealing with head motions, and voluntary motions can easily over-

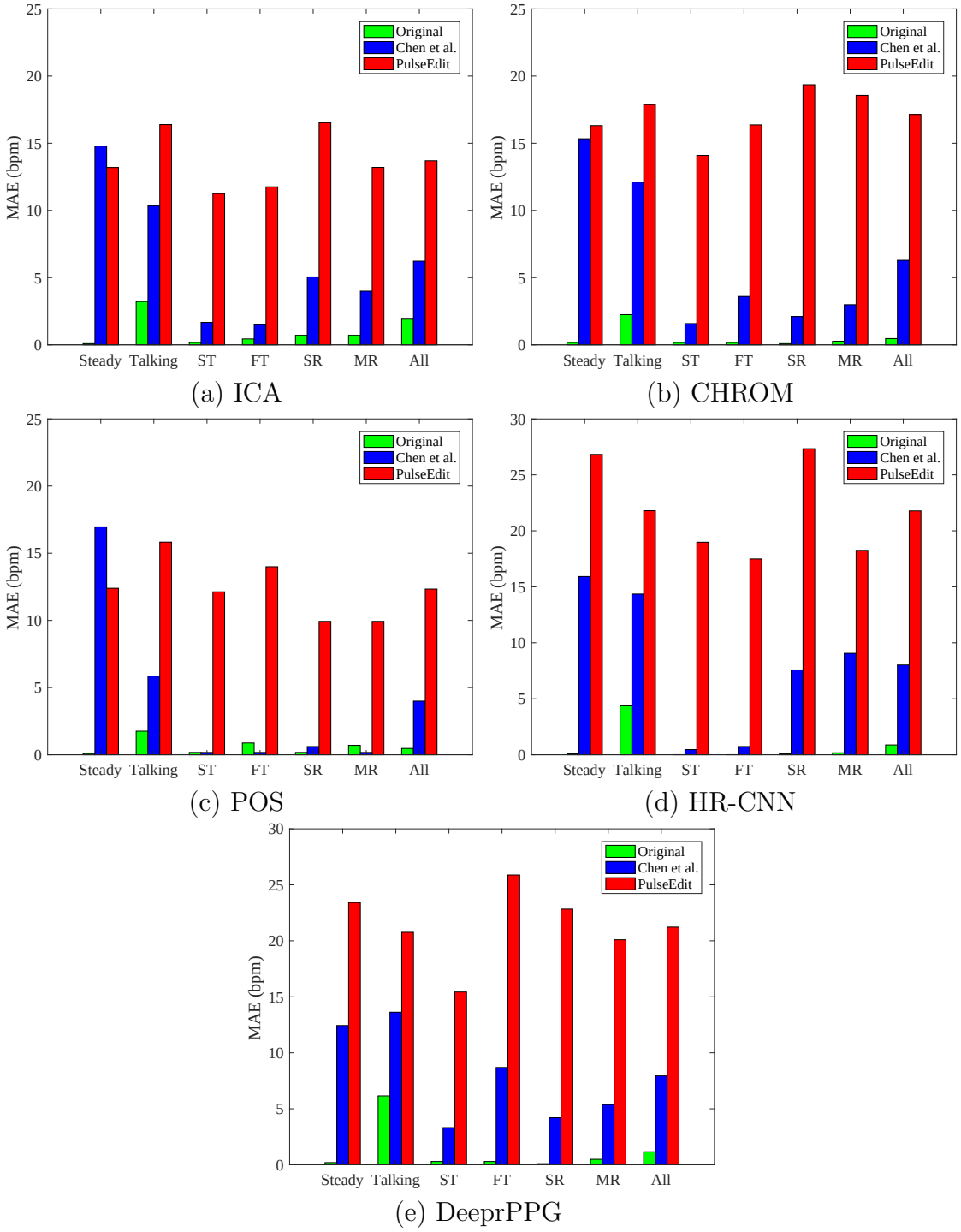


Figure 3.9: Performance comparison between PulseEdit and Chen's method in different subject motion settings with five rPPG methods: (a) ICA, (b) CHROM, (c) POS, (d) HR-CNN, and (e) DeeprPPG. The motion settings are steady, talking, slow translation (ST), fast translation (FT), small rotation (SR), and medium rotation (MR).

whelm the subtle HR-induced motion in the video. In comparison, PulseEdit has little variation in the performance among all 6 different motion settings, indicating that our proposed method is effective in a variety of motion settings.

	MAE of HR estimation (bpm)					PSNR (dB)
	ICA	CHROM	POS	HR-CNN	DeeprPPG	
Original	1.91	0.47	0.47	0.88	1.16	n/a
Chen *et al.* [76]	6.23	6.29	4.00	8.02	7.94	47.01
PulseEdit (Ours)	13.70	17.14	12.33	17.78	21.23	57.81

Table 3.2: Performance comparison of rPPG removal methods

Chen's method first estimates the color intensity variations from pixel level, and then subtracts the pixel-wise intensity variations from the original video to remove the physiological signals. The color intensity variation in each facial pixel is a combined consequence of pulse-induced color variation and voluntary motion. For head motion cases, the color intensity variation caused by voluntary motions can easily overwhelm the pulse-induced color variation in the video. This could explain the reason why Chen's method has ineffective editing performance for motion cases. In contrast, PulseEdit extracts the pulse-induced skin color variations from multiple facial subregions. The tracking of facial subregions can alleviate the interference of color variations caused by voluntary motions. Also, PulseEdit finds the optimal perturbation such that the original rPPG signals in multiple facial subregions can directly transfer to the target rPPG signal. Hence, PulseEdit can deal with both steady cases and motion cases.

3.5 Analysis of Adversarial Scenarios

Since PulseEdit can edit rPPG signals in videos, we expect that PulseEdit, as an adversarial operation, can circumvent rPPG-based liveness detection [27,29] and rPPG-based deepfake detection [30]. Thus, we conducted experiments on the HKBUMARsV1+ dataset [28] for liveness detection and the Celeb-DFv1 dataset [90] for deepfake detection to evaluate the effectiveness of PulseEdit on above two aspects, respectively.

3.5.1 Analysis Against rPPG-based Liveness Detection

Liveness detection aims at detecting whether a person seen by a camera is in his/her true live appearance or wearing a camouflaging mask with different facial appearances, a profile photo, or a video replay, to prevent face spoofing in identity authentication. Since live faces and many spoofed faces often have different characteristics in rPPG features extracted from the facial area, several prior publications have presented classifiers based on rPPG features. We test two rPPG-based liveness detection methods, namely, Hernandez's method [29] and Liu's method [27], as a proof-of-concept, to analyze the performance of PulseEdit on circumventing the rPPG-based methods.

We conducted experiments on the HKBUMARsV1+ dataset [28], which consists of video recordings of 12 subjects in flesh and wearing 3D face masks of different appearances. We set live facial videos as negative and 3D mask videos as positive. The classifier settings are the same as stated in [27,29]. PulseEdit was applied to the

3D mask videos, with the target rPPG signals generated using the same procedure as in the rPPG modification task in Section 3.4.1. We used subject-independent 5-fold cross-validation to evaluate the performance of the detector on the videos before and after PulseEdit.

We report equal error rate (EER) and AUC in Table 3.3 to show the impact of PulseEdit on the rPPG-based liveness detection algorithms. EER refers to the point where false positive rate and false negative rate are equal. AUC refers to the area under the receiver operating characteristic (ROC) curve. Smaller EER and larger AUC indicate better detection ability. We can see that PulseEdit increases the EER from 0.29 to 0.40 and decreases the AUC from 0.77 to 0.31 for Hernandez's method [29], and increases the EER from 0.10 to 0.26 and decreases the AUC from 0.94 to 0.73 for Liu's method [27]. On average, the 5-fold cross-validation shows that 96% of correctly classified 3D mask videos in Hernandez's method are classified as live videos after we apply PulseEdit to these videos and 64% in Liu's method. These results suggest that the current form of PulseEdit can already circumvent the rPPG-based liveness detection to some extent, and additional optimization may enhance such evasion by incorporating information from the existing research of liveness detection.

3.5.2 Analysis Against rPPG-based Deepfake Detection

The fast development of deep learning enabled computers to transform a person's face to another's in images and videos. These "deepfake" videos can spread

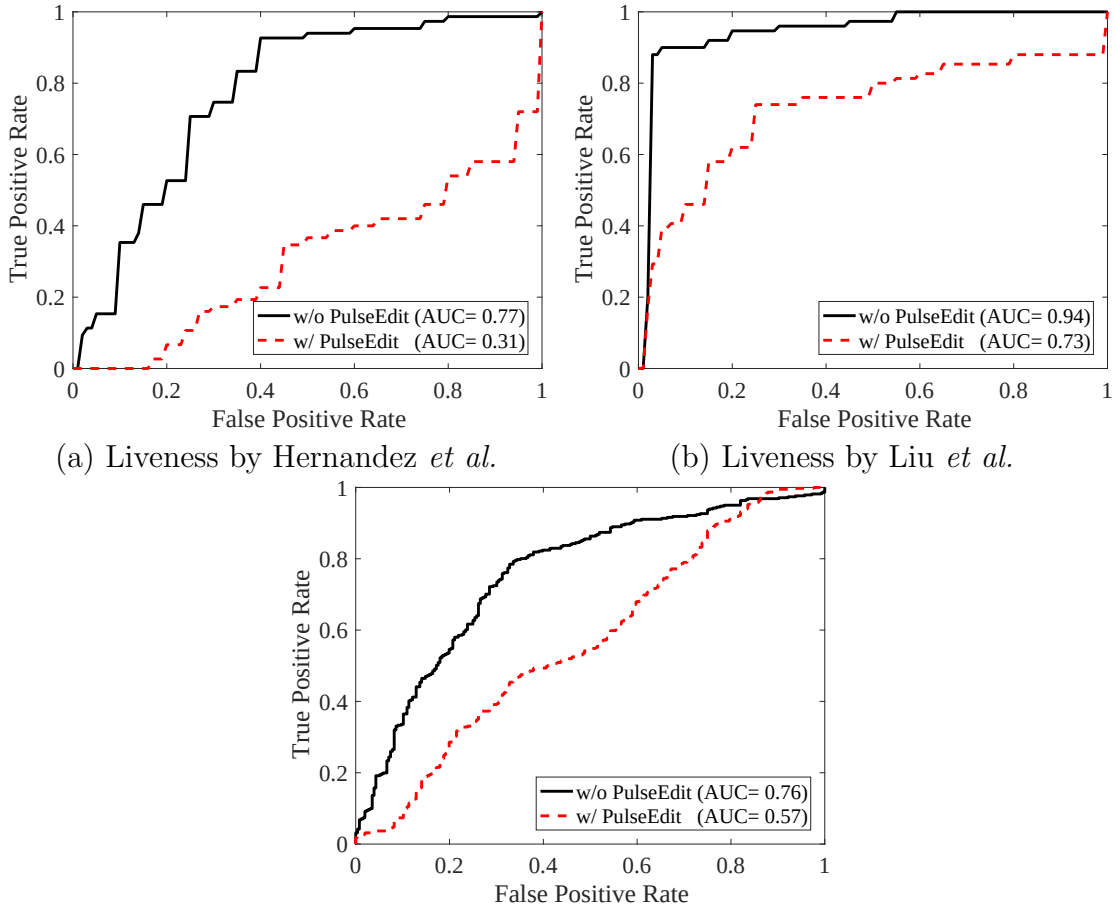
misinformation and fake news and impair the integrity of social media, prompting a strong and urgent need for investigating the detection of deepfake videos. Recently, FakeCatcher [30] was proposed to utilize rPPG signals from the video as features to detect whether the video is real or deepfake. To analyze the effectiveness of PulseEdit, we tested PulseEdit videos using the FakeCatcher CNN model.

Method	Dataset	w/o PulseEdit		w/ PulseEdit	
		EER	AUC	EER	AUC
Hernandez *et al.* [29]	HKBUMARsV1+	0.29	0.77	0.40	0.31
Liu *et al.* [27]	HKBUMARsV1+	0.10	0.94	0.26	0.73
FakeCatcher [30]	Celeb-DFv1	0.29	0.76	0.47	0.57

Table 3.3: Impact of PulseEdit on performance of rPPG-based visual security algorithms

We conducted experiments on the Celeb-DFv1 dataset [90], which consists of 370 real videos and 733 deepfake videos in the training set, and 38 real videos and 62 deepfake videos in the test set. We consider real videos as negative and fake videos as positive, and trained the FakeCatcher CNN model in the training set. The CNN architecture is the same as stated in [30]. As shown in Figure 3.10(c) and Table 3.3, FakeCatcher achieves an EER of 0.29 and an AUC of 0.76 in the test set.

We applied PulseEdit on the deepfake videos in the test set, with the rPPG signals extracted from the corresponding real videos as the target rPPG signals. In other words, we tried to restore the rPPG signal before deepfake was applied to the video. From the classification performance of the pretrained FakeCatcher on the test set with PulseEdit, we observe that the EER increases to 0.47 and the AUC reduces to 0.57, indicating that the rPPG signals inserted by PulseEdit can circumvent FakeCatcher, making FakeCatcher consider the deepfake videos as



(a) Liveness by Hernandez *et al.*

(b) Liveness by Liu *et al.*

(c) FakeCatcher

Figure 3.10: ROC curves of (a) Hernandez's and (b) Liu's methods of rPPG-based liveness detection, and (c) FakeCatcher of rPPG-based deepfake detection before and after PulseEdit

trustworthy. Among the deepfake videos in the test set that are correctly classified by FakeCatcher as forgery, we find that 49% of these videos become to be classified as unforged videos after PulseEdit.

3.6 Discussion

Considering the running time and the HR estimation error of PulseEdit, the proposed PulseEdit is an effective algorithm to edit rPPG signal in facial videos.

Compared with the prior art [76] that only focuses on eliminating the rPPG information, we have designed PulseEdit with two modes: rPPG removal and rPPG modification. The former mode can remove the rPPG information and the latter mode can change the rPPG information to a target HR designed by users. The proposed algorithm offers the users more options of editing operations on the physiological signal in facial videos regarding physiological privacy protection. PulseEdit also provides a better capability to remove the physiological signal from videos with head motions (*i.e.*, talking, translation, and rotation), more robust to deal with different practical recording cases.

Considering PulseEdit as an adversarial operation to rPPG technology, we have studied to what extent PulseEdit can circumvent rPPG-based visual security algorithms. As a proof-of-concept, we consider the rPPG-based liveness detection and deepfake detection algorithms. The experimental results demonstrate noticeable performance drops between the original videos and the PulseEdit videos, indicating that PulseEdit can successfully mitigate the rPPG-based visual security algorithms. From the perspective of threat modeling for these visual security algorithms, our PulseEdit research suggests that it is important to investigate this and other similar vulnerabilities and improve the rPPG-based visual security algorithms against adversarial operations.

Over the past decade, rPPG technology has been prospering and it has become feasible to monitor vital signs, such as HR, using commercial digital cameras in daily life. One common bottleneck in the R&D of rPPG technology is the lack of sufficient facial videos with known HR of a wide range [91]. PulseEdit in rPPG modification

mode may be used to synthesize facial videos with controllable HR to enlarge the dataset and facilitate the R&D of rPPG technology.

In the current form, a weakness of PulseEdit is that it focuses on altering the rPPG information in each frame and has not explicitly considered to conceal the manipulation traces introduced by itself. Forensic tools such as steganalysis can detect the presence of perturbation from the uncompressed frames if available. Nevertheless, we find that lossy video compression is a feasible approach to improve the resistance of the edited frames against forensic analysis and retain the edited rPPG signal in the video. In future work, the inclusion of various forensic undetectability into the framework of PulseEdit and the development of new detectors to detect these manipulations could be two intertwining research directions. In addition, the current form of PulseEdit perturbs the facial pixels independently and the algorithm, in the future, can take spatial and temporal correlations of facial pixels into consideration for the pixel perturbation to further minimize the perceptual distortion of facial videos.

3.7 Chapter Summary

In this chapter, we have proposed *PulseEdit*, a novel algorithm that can edit the rPPG signal in facial videos without visible distortion, to protect the physiological information from disclosure. We design a set of perturbation frames to add to the input video frames to change a person's intrinsic rPPG signal that is presented in the facial region. PulseEdit can either remove the rPPG signals on the face or

change them to a target heart rate. Extensive experimental results demonstrate the effectiveness and robustness of PulseEdit in different facial subregions, and various rPPG algorithms can no longer detect the accurate heart rate from facial videos after applying PulseEdit. We also show that PulseEdit can potentially circumvent rPPG-based liveness detection and deepfake detection, suggesting a direction for improvement in these areas. In the current form, the traces of PulseEdit can be detected by forensic steganalysis from the uncompressed video frames, but lossy video compression can significantly reduce the forensic performance. In future work, we plan to investigate the inclusion of various forensic undetectability criteria into our algorithm, to gain insights on the ability of PulseEdit as an antiforensic tool and the competing direction of detecting the manipulations made by PulseEdit.

Chapter 4

Identity Privacy Protection in rPPG Technology

4.1 Overview

Visual-based physiological monitoring, namely remote photoplethysmography (rPPG) technology, has gained more and more attention in the research field. rPPG technology can capture subtle color change from a person's skin and measure physiological information (e.g., heart rate [20, 21, 64], heart rate variability [24], and breathing rate [22, 23]) using a regular RGB camera. In the R&D of the rPPG technology, human facial skin is the most frequently used region of interest for vital signs extraction, as it is the most exposed skin area and exhibits a relatively strong pulsatility response than other body parts. Collaborative facial video data collection and benchmarking play an essential role in the performance evaluation and comparison of the rPPG algorithms. However, the facial video data are often very sensitive regarding the privacy of the persons being recorded. They have a high concern of being misused, which discourages healthy sharing of data by the community to foster the advancement of this area. A few public benchmark datasets [20, 81, 92–94] for evaluating remote HR measurement methods have been established to propel the

R&D of the rPPG technology, but none of them remove the identifiable appearance features of the subjects before publishing. Although these datasets have strict user agreements on data usage (*e.g.*, academic and prototyping purpose), there is still a potential risk of identity privacy leakage. This has been a major hurdle to foster inter-organizational collaborations and benchmarking to advance the field.

To address the identity privacy concerns in the rPPG technology, we develop an anonymization transform that removes sensitive visual information identifying an individual, but in the meantime, preserves the subtle color and movement changes that contain relevant physiological information. After the transform, we remove the person's identification characteristics from the facial video while retaining rPPG signals for physiological sensing applications. In this way, we can de-identify the subjects' facial videos in the benchmark datasets before we publish the datasets to the research community. Using these de-identified public datasets, we have little concerns about identity privacy leakage, and different rPPG algorithms can be comprehensively compared on the same ground over a large data collection contributed from different sources, which fosters the synergistic improvement in the R&D of the rPPG technology.

We consider the following aspects to evaluate the anonymization transform:

- **Physiological fidelity:** the distortion of rPPG signals extracted from the processed videos, compared to the original videos, should be small enough to preserve the fidelity of physiological information.

- **Recognizability:** the person's identity has a low probability of being recog-

nized from the processed videos.

In addition to enhancing the identity privacy protection for the public benchmark datasets, the anonymization transform can also be applied as a pre-processing before people upload the selfie facial videos for the services of rPPG analysis.

4.2 Related Work

4.2.1 Public Datasets for Remote HR Estimation

Due to facial videos' sensitivity and privacy issues, many existing rPPG algorithms reported their performance using private datasets, causing difficulties in performance comparison across different algorithms. Public benchmark datasets for evaluating remote HR measurement methods are very limited. Some public datasets were initially designed for emotion analysis but were tailored to evaluate the performance of rPPG algorithms. The MAHNOB-HCI dataset [92] consists of 527 videos of 27 subjects, recorded with small head movement and facial expression variation under laboratory illumination. The MMSE-HR [93] consists of 102 videos of 40 subjects involving various subjects' facial expressions and head movements.

There are also a few public-available datasets especially designed for the task of remote HR estimation. The PURE dataset [81] consists of 60 videos from 10 subjects, in which all the subjects were asked to perform 6 motion settings such as talking or head rotation. The PFF dataset [94] consists of 104 videos of 10 subjects, in which small head movements and illumination variations are involved. The VIPL-HR dataset [20] contains $3,130$ facial videos from 107 subjects, which were recorded

with various sensors, different illumination, and head poses.

We summarize the public benchmark datasets for remote HR estimation in Table 4.1 and show exemplary facial frames in these datasets in Figure 4.1. To perform a comprehensive evaluation for rPPG algorithms, the dataset contributors considered a variety of scenarios in data collection, such as different sensors, different illumination, and head motions. Especially, The VIPL-HR dataset [20] expands the number of recruited subjects and the number of videos in the collection. Nevertheless, these public datasets have potential risks to leak the subjects' identity privacy, and the dataset contributors have not concealed the subjects' identifiable appearance features before publishing the datasets.

Dataset	# of Subjects	Illumination	Motion	Sensor	# of Videos
MAHNOB-HCI	27	L	E	C	527
MMSE-HR	40	L	E	C	102
PURE	10	L	S/M/T	C	60
PFF	13	L/D	S/M	C	104
VIPL-HR	107	L/D/B	S/M/T	C/N	3,130

L = Lab environment, D = Dim environment, B = Bright environment,
E = Expression, S = Steady, M = Movement, T = Talking,
C = Color camera, N = NIR camera

Table 4.1: Summary of the public benchmark datasets for rPPG analysis

4.2.2 Identity Privacy Protection for Facial Images

To address the identity privacy issue in rPPG technology, a conceptual solution was proposed in [95] to use a single-element camera to record facial videos for rPPG extraction. Single-element camera refers to single-pixel photo-detector. Since it has no spatial resolution, this kind of camera does not record the subject's face,

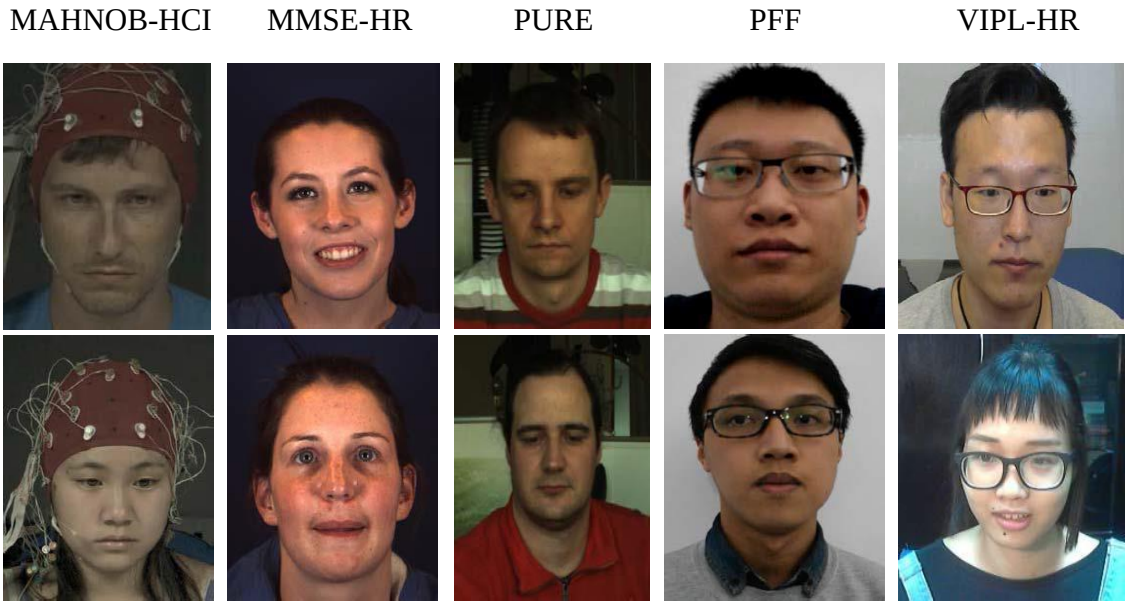


Figure 4.1: Exemplary frames in several public benchmark datasets for rPPG analysis. None of them apply some processes to conceal the subjects' appearance features.

and no identifiable appearance features are disclosed; thus, it fundamentally eliminates privacy concerns. In some sense, the single-element camera is similar to the conventional contact-based oximeter sensor that uses a photo-diode to measure the blood volume changes. However, a normal camera sensor for rPPG is susceptible to environmental noises, such as illumination and motion, let alone a single-element camera. Since the single-element camera throws away most facial color information, this aggressive concealment of identity can not guarantee the fidelity of the rPPG signal. Also, there will be little room and freedom for R&D to improve the performance of rPPG technology, given the recorded signals from a single-pixel camera sensor. In contrast, our proposed anonymization transform can provide options for different privacy strength needs and balance privacy protection and physiological fidelity.

In the image processing field, morphing-based privacy protection [96] was proposed to protect identity privacy in surveillance videos. Since the surveillance videos provide recordings of public places to the authorities for social security, the morphed faces should be recoverable if needed, given the parameters of the morphing process that can be encrypted with a secret key. In contrast, our anonymization transform does not necessarily require reversibility of the transformed faces since our goal is to conceal the person's appearance from anyone. Hence, our algorithm can be more aggressive to throw away a person's appearance information as long as retaining physiological information.

4.3 Anonymization Transform for Privacy Protection

Empirically, two types of facial features can help recognize a person's identity. The first is facial contour and facial proportion features, such as a round face, an oval face, and big eyes. These features leave a general and profound impression of how the person looks like. The second one is the facial surface features, such as protruding forehead, deep eye socket, and a high-bridged nose. These features have been successfully applied to identify a person on electric devices, for example, Face ID in iPhones. When a color camera captures a person's face, the facial surface can be inferred from the facial color variation in the picture. For example, a protruding forehead makes the color of the forehead region brighter than other regions in the video.

Considering the above two types of features, we design an anonymization trans-

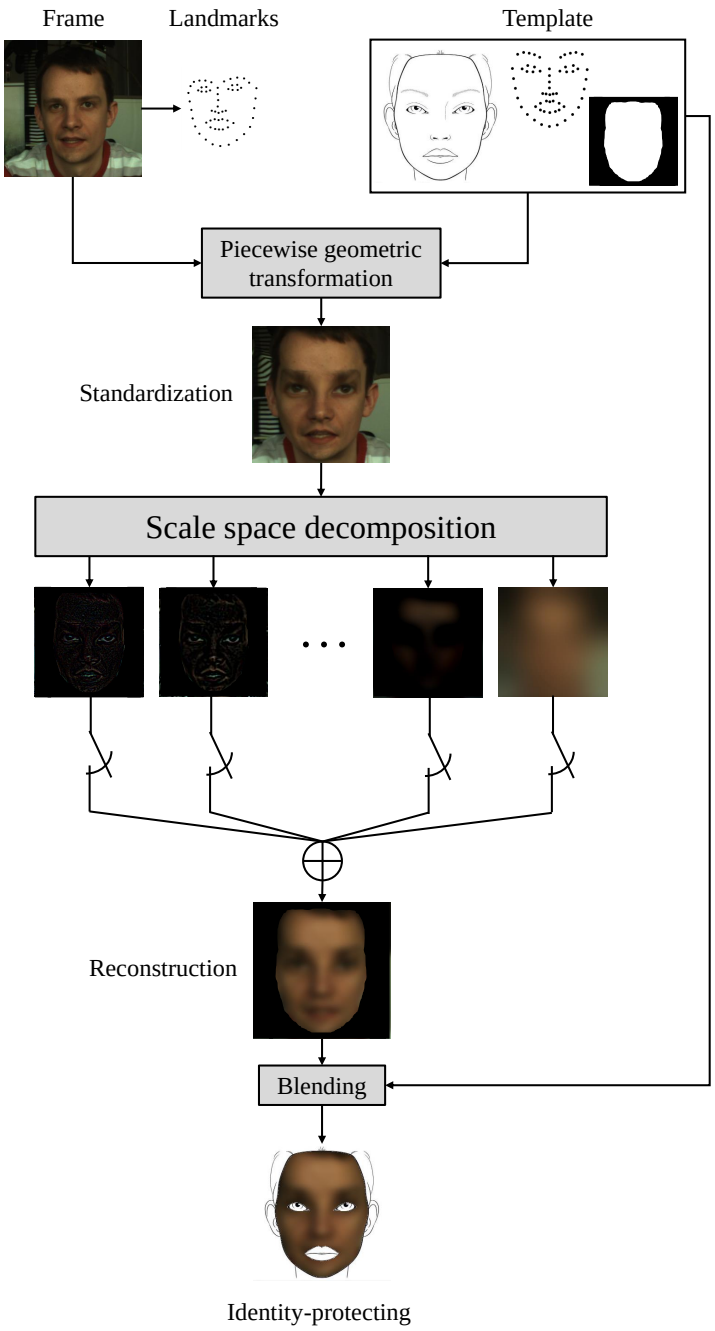
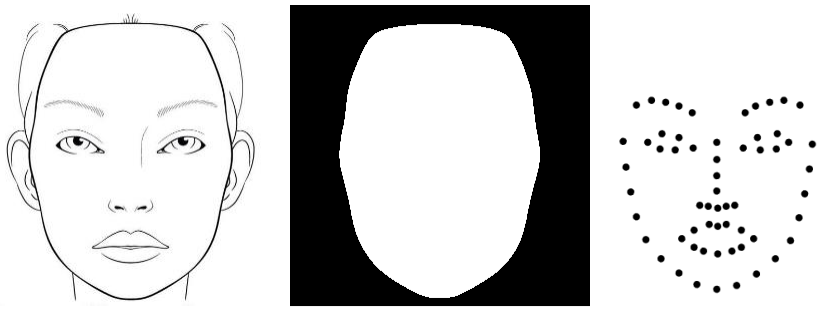


Figure 4.2: Pipeline of anonymization transform to protect identity privacy in rPPG data collection. Piecewise geometric transformation first standardizes the original face to a generic template face according to the facial landmarks. Then, spatial decomposition decomposes the transformed face into multiple levels. Finally, the identity-protecting face is reconstructed using parts of decomposition levels to accommodate different privacy strength needs.

form to remove the identifiable appearance feature, consisting of two main steps – standardization and spatial frequency decomposition. Standardization aims to map the original face onto a generic template face, removing the person's facial contour and facial proportion features. The goal of spatial decomposition is to remove the local color variation information that may indicate the facial surface features. We present the pipeline in Figure 4.2.

4.3.1 Standardization to a Generic Face Template

We provide a generic face template beforehand, shown in Figure 4.3, and transform the original face to the given template to eliminate facial contour and facial proportion features. We can obtain the facial mask and facial landmarks of the generic face template in registration beforehand.



(a) (b) (c)

Figure 4.3: Generic face template as a target face. (a) Face template, (b) corresponding facial mask, and (c) corresponding facial landmarks.

For the coming facial frame, we apply the facial landmark detector by Dlib [77] to locate and track 68 facial landmarks. We use piecewise geometric transformation with facial landmarks as anchor points to standardize the person's face to a generic

face template, shown as the transformed face in Figure 4.2. To implement piecewise geometric transformation, we divide the images using Delaunay triangulation [97] with the anchor points as vertices of the triangles and estimate geometric transformation parameters for each triangle. Then, we apply the geometric transformation on the pixels within each Delaunay triangle using the corresponding geometric transformation parameters.

4.3.2 Scale Space Decomposition

The difference-of-Gaussian (DoG) representation [98] of an image allows us to decompose the image contents into different spatial-frequency subbands and analyze the behaviors of the components in different subbands.

To eliminate the facial surface features, we decompose the transformed face into different spatial-frequency subbands and then reconstruct the face using only partial subbands. A DoG representation with N levels is constructed by taking the differences of Gaussian-blurred images. The n-th subband is computed as follows:

$$D_n = G_n - G_{n+1}, \quad n = 1, ..., N-1. \tag{4.1}$$

where G_1 is the input image and $G_n, n = 2, ..., N$ is the blurred image of G_1 using a Gaussian filter with the mean 0 and the standard deviation $\sigma^{n-1}, \sigma > 1$. In the DoG representation, we have $D_N = G_N$.

Since we desire to include only the facial pixels in the DoG representation and exclude the background region, we apply masked filtering as follows:

$$G_n = \frac{(T \bigodot M) \bigotimes W_n}{M \bigotimes W_n}, \quad n = 2, ..., N. \tag{4.2}$$

where T is the transformed image, M is the binary facial mask of the template, W_n is the Gaussian filter at level n, $\bigodot$ denotes element-wise multiplication, and $\bigotimes$ denotes image convolution. The filtering process is identical to the normal filtering when the pixels are in the facial region. The filtering process excludes the pixels outside the facial region. For the pixel on the boundary of the facial region, the filtered value is normalized by the filter weights contributing to the filtering processing.

We reconstruct the face with the selected levels in the DoG representation,

$$R = \sum_{n \in \mathcal{L}} D_n. \tag{4.3}$$

where $\mathcal{L}$ represents the set of the selected levels, whose elements ranging from 1 to N. Conceptually, the levels with smaller indices contain relatively high-frequency texture, *i.e.,* reflected illumination on the face which indicates the person's facial surface features; the levels with larger indices contain the person's skin color information that we intend to maintain for rPPG analysis purposes. Therefore, we incline to select lower pyramid levels to reconstruct the face. After the reconstruction and blending with the generic face template, we obtain the identity-protecting face that can be shared with the public with fewer concerns of identity privacy leakage.

4.4 Experimental Results

We conducted experiments on the facial videos collected for rPPG research in the PURE dataset [81], which consists of 10 subjects. The pulse signals were recorded simultaneously with a pulse oximeter. We study the physiological fidelity and the privacy strength of the proposed algorithm and also do an ablation study to investigate the contributions of different modules. In our experiments, we set $N = 7$ and $\sigma = 2$ in spatial decomposition.

4.4.1 Analysis of Physiological Signal in DoG Layers

To investigate the contribution of the different layers in the DoG representation to the physiological fidelity, we analysis the physiological signal components in different layers. We extract the rPPG signals from each DoG layer and compare the similarity of the power spectral density (PSD) with the rPPG signal extracted from the original facial videos. The higher similarity indicates the more component of the physiological signal in that layer. We use Periodogram to compute the PSD and evaluate the similarity as

$$s_{x,y} = \frac{\mathrm{PSD}_x^T \cdot \mathrm{PSD}_y}{|\mathrm{PSD}_x||\mathrm{PSD}_y|} \tag{4.4}$$

where PSD_x and PSD_y are the PSD vectors of the DoG layer and the original facial videos.

Figure 4.4 presents the similarity of the physiological signal between the DoG

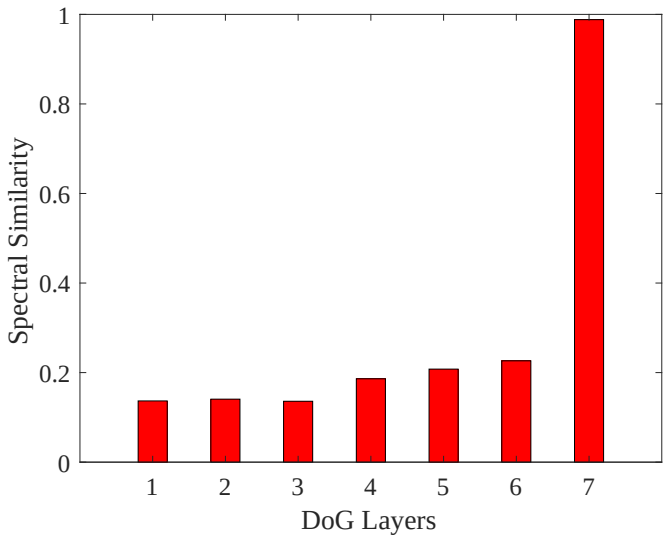


Figure 4.4: Spectral similarity of the physiological signal between the DoG layers and the original facial videos.

layers and the original facial videos. We can see that Layer 7 retains almost all the component of the physiological signal, and the similarity reaches 0.98. The rest of the layers (Layer 1 to 6) only retains a few component of the physiological signal. Among these layers, we observe a relatively abrupt increase in the similarity starting at Layer 4. Overall, Layer 1 to 3 hardly contains the physiological signal; Layer 4 to 6 contains a small amount of the physiological signal; Layer 7 retains most of the component of the physiological signal. This observation is consistent with our conceptual knowledge that the upper layers contain the facial texture and surface features and the lower layers maintain the skin color information which is the medium of the physiological signals.

4.4.2 Performance of Anonymization Transform

To show the identity privacy strength of the anonymization transform, we choose different combinations of the DoG layers to reconstruct the identity-protecting faces. We reduce the layers one by one from the upper layers to reconstruct the transformed face. We denote "Rn" as the reconstructed face with n-th to N-th layers. For example, we use the 3rd to 7th layers to reconstruct "R3".

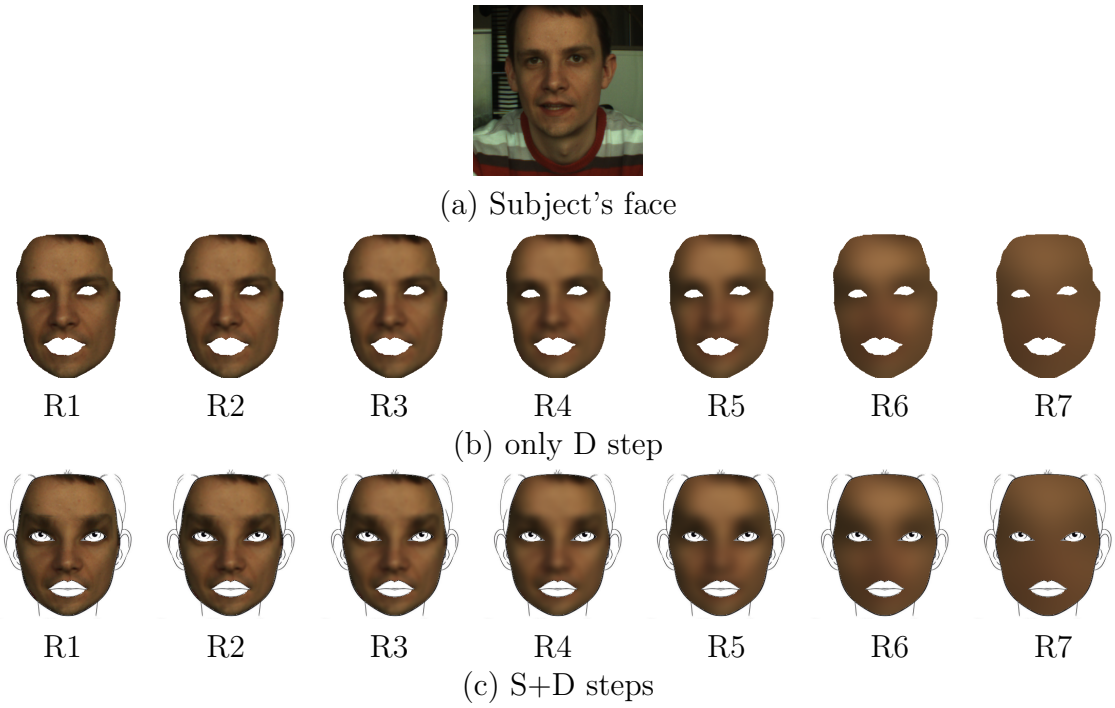


Figure 4.5: Identity-protecting faces using different layers in reconstruction. (a) Subject's face, (b) only D step, and (c) S+D steps.

We compare the identity-protecting results with and without standardization step. For short, "S" denotes the standardization step, and "D" denotes the decomposition step. Note that, "R1" of D is equivalent to the original face; "R1" of S+D is equivalent to the standardization face. Figure 4.5 shows the identity-protecting

faces using different layers in reconstruction. We can observe that when we exclude more upper layers of the DoG representation in the reconstruction, the more detailed facial features are removed. Specifically, "R4" is a turning point of the elimination of the facial features. The textures around the nose area are clear before "R4", and are removed after "R4". When we only use the D step, the facial contour and proportional features still remain, which may leak some subject's identifiable features. In contrast, D+S steps can remove this kind of feature by standardizing the faces onto a generic template.

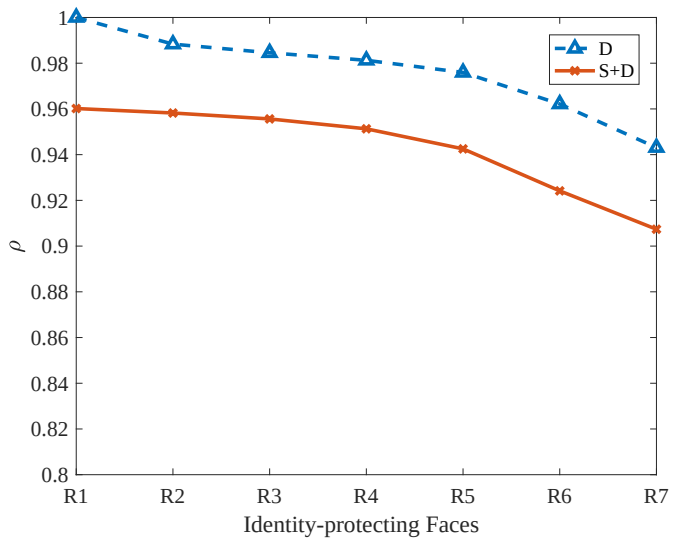


Figure 4.6: Correlation of the rPPG signals extracted from the original videos and the processed videos.

Figure 4.6 presents the physiological fidelity of the proposed anonymization transform. We use the widely-used rPPG algorithm, POS [15], to extract the rPPG signal from the facial videos. We use Pearson's correlation coefficient to evaluate

the physiological fidelity of the proposed algorithm.

$$\rho = \frac{(\mathbf{x} - \mu_{\mathbf{x}})^T(\mathbf{y} - \mu_{\mathbf{y}})}{\sqrt{(\mathbf{x} - \mu_{\mathbf{x}})^T(\mathbf{x} - \mu_{\mathbf{x}})}\sqrt{(\mathbf{y} - \mu_{\mathbf{y}})^T(\mathbf{y} - \mu_{\mathbf{y}})}} \tag{4.5}$$

where $\mathbf{x}$ and $\mathbf{y}$ denote the rPPG signals extracted from the original video and the processed video, respectively. We can see that as we reduce the number of layers to reconstruct the face, the rPPG correlation between the processed video and the original video decreases. Nevertheless, the Pearson's correlation coefficient is greater than 0.9 in R7, when we remove most of a person's appearance features). We also observe that starting from R4, the dropping slope becomes slightly sharper than before. This observation is consistent with results in Section 4.4.1 that Layer 4 to 6 contain more components of the physiological signal than Layer 1 to 3. Since Layer 1 to 3 hardly contain the physiological signal, Pearson's correlation curve is flat at the beginning. As more and more component of the physiological signal is removed from Layer 4 to 6, the dropping slope increases. On the whole, we can see that the rPPG signals are mostly retained after being processed by our proposed algorithm.

4.4.3 Facial Feature Visualization

To investigate how strong the privacy strength of the anonymization transform is, we visualize the facial feature embedding of the original faces and the transformed faces. We use a deep learning-based face recognition model by Dlib to extract facial features from the original face and the transformed face. The Dlib model, based on Resnet-34, outputs a 128D feature vector and has a state-of-the-art accuracy

performance of 99.38% on the Labeled Faces in the Wild (LFW) datase [99]. The closer distance between two facial features indicates the high probability that the two faces can be recognized as the same person. We use principle component analysis (PCA) to reduce 128D to 2D for visualization.

Figure 5.4 presents the facial feature embedding in the different levels of the reconstructed face. We denote the original faces as the people's indices and the transformed faces (*e.g.*, privacy-protecting faces) as "transformed". We can see that as the level index increases, the facial features of the privacy-protecting faces gradually separate from the distribution of the original facial features and concentrate into an area away. "R4" is a starting point that the facial features of the privacy-protecting faces split away from the original facial features, and the feature distance increases sharply between the privacy-protecting faces and the original faces. This suggests that the privacy-protecting faces in large level index remove the identifiable appearance features, and it is hard to correspond these faces with the original ones.

4.4.4 Recognizability of Privacy-Protecting Faces

We evaluate the recognizability of the privacy-protecting faces by building face recognition classifiers with the original faces and test the recognition performance on the privacy-protecting faces. To reduce the computational complexity, we first apply PCA to reduce the dimension of the facial features extracted in Section 4.4.3 from 128D to 30D. We train two types of classifiers with the original faces in this experiment.

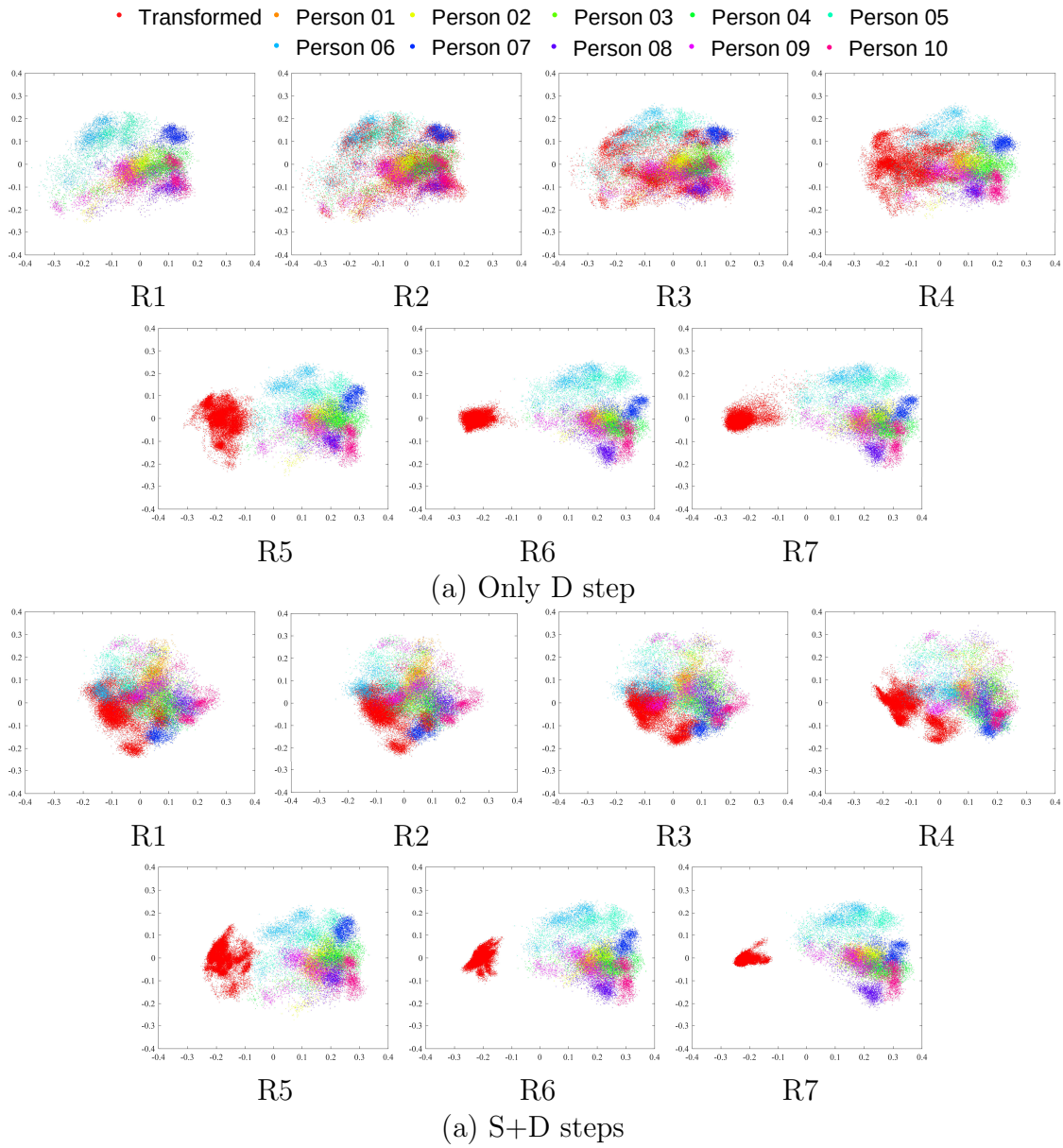


Figure 4.7: Visualization of facial feature embedding in different levels of reconstructed face. (a) Only D step, (b) S+D steps.

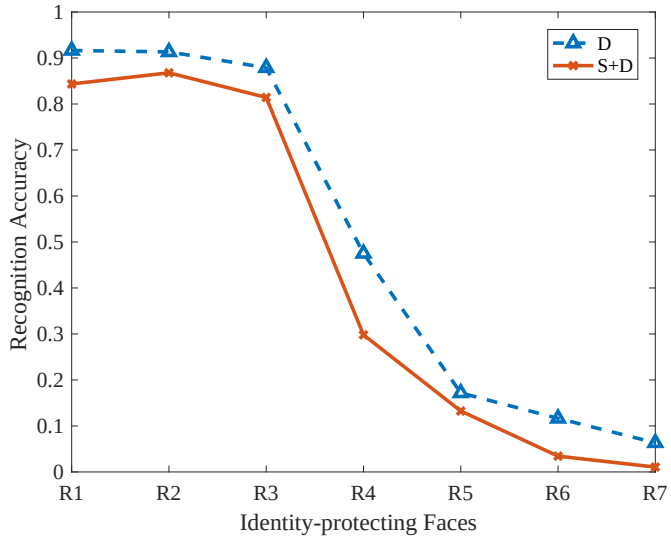
- Gaussian: we train Gaussian models for each person, *i.e.*, $\mathcal{M}_i \sim \mathcal{N}(\mu_i, \Sigma_i), i = 1, ..., 10$. The recognition prediction of the facial feature w can be computed using maximum likelihood: $\arg\max_i \quad \mathcal{M}_i(w)$

- SVM: we train a multi-class SVM [100] with a radial basis function kernel.

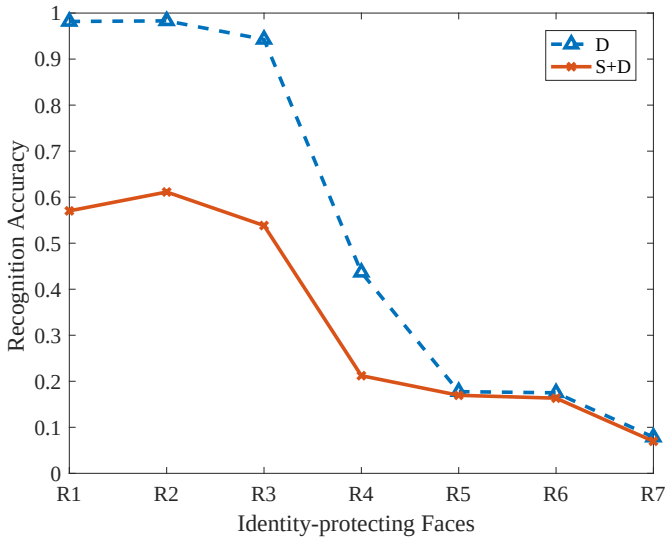
We then input privacy-protecting faces with different levels to test the recognition performance, which is presented in Figure 4.8. We also study the influence of the S step on recognizability. We can see that "only D step" and "S+D steps" have similar behaviors in recognition performance. From R1 to R3, the accuracy maintains a plateau. Starting from R4, the recognition accuracy decreases abruptly, and the accuracy reduces significantly to no more than 10% at R7. This curve trend is consistent with the visualization of the identity-protecting faces in Figure 4.5 that more facial details are removed starting from R4. Comparing "only D step" and "S+D steps", we can see that "S+D steps" has lower recognition accuracy, which suggests that the S step enhances the privacy strength of the identity-protecting faces positively. In addition, the S step has significant benefits in concealing people's identities for the SVM classifier, especially from R1 to R3.

4.5 Chapter Summary

In this chapter, we have proposed an anonymization transform to protect the identity privacy of facial videos in rPPG data collection. We first standardize the input facial images to a generic face template and then remove the local skin color variation via spatial decomposition and reconstruction. Using the proposed transform, we can conceal the person's facial proportional features and facial surface features, possibly inferred from the color variation on the face. Experimental results show that the proposed algorithm can eliminate identifiable appearance features from facial videos, and in the meantime, retain the rPPG signal in high fidelity for



(a) Gaussian



(b) SVM

Figure 4.8: Recognition accuracy of identity-protecting faces. (a) Gaussian model and (b) SVM.

post-processing and analysis. The proposed algorithm can encourage the sharing of facial video data by the community for R&D of rPPG technology to foster the advancement of this area, with few concerns of identity privacy risk. Also, the anonymization transform can be used as a pre-processing when people capture selfie

videos for rPPG analysis to conceal their identifiable features and protect their identity privacy.

Chapter 5

Distribution Parity for Group Fairness

5.1 Overview

In the context of decision making, fairness typically means the equivalent (non-discriminatory) decision performances across different groups of people based on their inherent or acquired characteristics. From this point of view, a number of recent studies introduced *demographic parity* (DP) [35,36] and *equalized odds* (EO) [37] to characterize and evaluate the fairness level in the machine learning models. Since the goal of these definitions is to equalize certain probability statistics computed from the confusion matrix of classification, this type of definitions were named as *classification parity* in [101]. To take the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) software as an example, DP means that parole granting rates should be equal across the race groups; and EO enforces that among defendants who would or would not have gone on to commit a violent crime if released, the rates are equal across the race groups.

The fairness algorithms in the literature mainly fall into three categories:

- Preprocessing: the studies by [102, 103] map the data to a fair representation in a latent space satisfying the defined fairness or independent of the protected attribute. Ideally, removing correlated features and finding independent representation to the protected attribute may lead to undesirable decision outcomes. Imagine a company scores randomly on the applicants over two groups. Even though the scoring function is independentof the groups, it is likely that a considerable number of unqualified applicants are selected at last.

- Constraints on training: the works by [104, 105] constrain the classifier on DP and EO, respectively, during the training. Fair classification can also be reduced to a set of sub-problems [106]. The tradeoff is required between accuracy and fairness constraints.

- Postprocessing: the work by [37] searches proper thresholds on scores over different groups. The studies by [107, 108] calibrate on the prediction score such that the probability of positive label is equal to the prediction score. The postprocessing requires access to protected attributes in the test phase.

Classification parity can help machine learning models achieve equitable decision performances across the groups from a probability point of view, such as positive proportion of decisions, true positive rate, false positive rate, and other statistics alike. However, such parity of the group-wise performances is generally not retained, when we tune the decision threshold. Trained under the constraint of classification parity, the classifiers only "delicately" achieve fairness requirement in the default decision thresholds, and are sensitive and vulnerable to the change

of decision thresholds. We will illustrate the limitation of classification parity in Section 5.2.2.

To alleviate the limitation of classification parity, we introduce a new notion of *threshold invariant fairness* in this chapter. Unlike classification parity, threshold invariant fairness enforces a stronger condition on the classifier such that it has a consistent fairness level of classification results against the change of decision threshold.

To develop our proposed notion, we find that it is sufficient to design the scoring function of the classifier to equalize the distributions of the risk scores over all groups. From classification parity to threshold invariant fairness, the equalization focus shifts from statistical attributes at a given decision rule to the probability distribution of the risk scores, suggesting that our proposed notion imposes a stronger constraint than that by the classification parity. In this chapter, we propose two approximation approaches to equalize the distributions of risk scores over all groups and incorporate them into a variety of differentiable classifiers, such as logistic regression and support vector machine. The problem can be solved by a gradient-based optimizer. Our new methods have two advantages compared with the prior art:

- We do not require the protected attributes in the test phase;
- We can tune the threshold to modify the positive rate of the classifier predictions and maintain the fairness as much as possible.

5.2 Background

5.2.1 Classification Parity

A machine learning model consists of a mapping function $g(\cdot)$ and a threshold t, shown in Figure 5.1. The mapping function $g(\cdot)$ maps the input feature X to a score, and then the prediction label is determined by a threshold t.

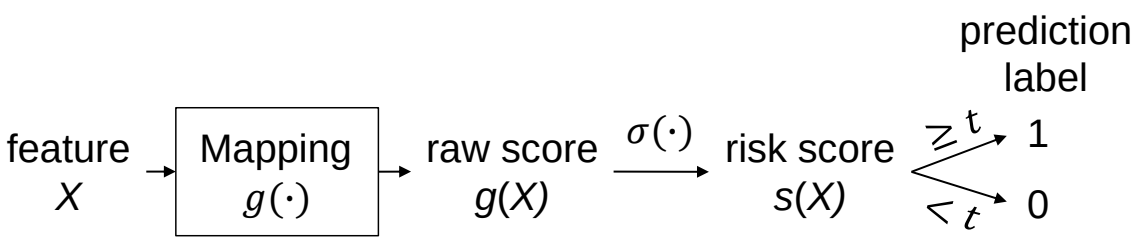


Figure 5.1: Working pipeline of machine learning models.

Classification parity is an approach to alleviate the machine bias by equalizing performance measure of classification across the groups with the protected attributes to achieve a group-wise fair classification. As early reviewed, *demographic parity* [35, 36] and *Equalized Odds* [37] are two well-known fairness definitions in classification parity. We assume that one input sample with observed m-dimensional features $X \in \mathbb{R}^m$ has a protected attribute $A \in \{0, 1\}$, and a ground truth label $Y \in \{0, 1\}$; and $\hat{Y} \in \{0, 1\}$ is the classifier's prediction. In a simplified recidivism example, A may represent the defendant's race as African-American versus Caucasian, Y represents whether the defendant recommits another crime after being released. The definition of two existing classification parities are described as follows.

Definition 1 *(Demographic Parity). A predictor $\hat{Y}$ satisfies demographic parity if*

$$P(\hat{Y} = 1|A = 0) = P(\hat{Y} = 1|A = 1). \tag{5.1}$$

Definition 2 *(Equalized Odds). A predictor $\hat{Y}$ achieves equalized odds if the true positive rates and false positive rates across the groups are equal, respectively, i.e.,*

$$\begin{cases} P(\hat{Y} = 1|A = 0, Y = 0) = P(\hat{Y} = 1|A = 1, Y = 0), \\ P(\hat{Y} = 1|A = 0, Y = 1) = P(\hat{Y} = 1|A = 1, Y = 1). \end{cases} \tag{5.2}$$

We see that DP enforces that the selection rates are equal across all the protected attributes; EO enforces that the prediction accuracy is equally high across all the protected attributes. Based on the above parity definition, many papers have proposed algorithms to achieve classification parity via pre-processing [102, 109], post-processing [37], and regularization techniques [35, 36, 104–106, 110].

5.2.2 Limitations

As pointed out by [101], classification parity is a problematic measure of fairness. We consider a risk score $s(X) = P(\hat{Y} = 1|X)$, describing the probability that an input sample X will be predicted positive. Given a group $\mathcal{D}_a$ with protected attribute $A = a$, we can compute the distribution of $s(X)$ over the group $\mathcal{D}_a$, which we refer to as risk distribution $r(\mathcal{D}_a)$. Figure 5.2 shows hypothetical risk distributions for two groups with different protected attributes. In the model inference stage, an input sample is classified by comparing its risk score with a decision threshold.

The proportion of the positive predictions in one group is the fraction of the risk distribution that is to the right of the decision threshold, highlighted in Figure 5.2.

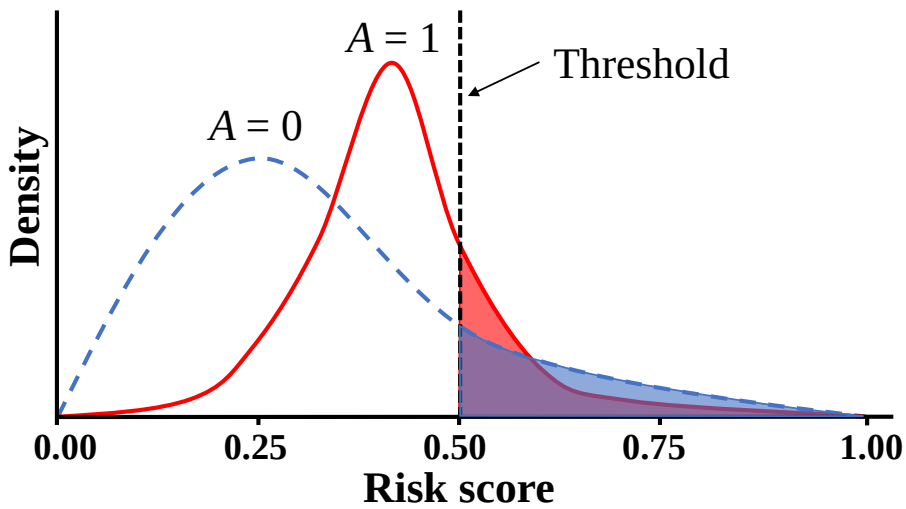


Figure 5.2: Hypothetical risk distributions of two groups (e.g., $A = 0$ and $A = 1$). When classification parity is satisfied, i.e., the areas of the highlighted regions are equal, the risk distributions can still differ from each other. Classification parity does not achieve equitable decision performance in terms of risk distribution.

According to the definitions reviewed in Section 5.2.1, to achieve classification parity is to equalize the positive (or negative) proportions among different groups. However, when the classifier achieves classification parity at a specific decision threshold, i.e., the areas of the highlighted regions are equal, the risk distributions can still differ from each other. This suggests that the classification parity can be easily destroyed by changing the decision threshold. The "fair" machine learning model under classification parity does not show equitable performances among different groups in terms of the risk distribution. The goal of this chapter is to propose a stronger condition on fairness to alleviate the above issue, and to develop systematic method by designing the scoring function during the classifier training to achieve it.

5.3 Distribution Parity for Threshold Invariant Fairness

5.3.1 Notations

Recall that an input sample with m-D features $X \in \mathbb{R}^m$ has protected attribute $A \in \{0, 1\}$, its corresponding ground truth label $Y \in \{0, 1\}$, and the classifier prediction is $\hat{Y} \in \{0, 1\}$. Define the scoring function in classifier $g : \mathbb{R}^m \mapsto \mathbb{R}$, mapping the m-D features to a scalar, the raw score. A *sigmoid* activation $\sigma(\cdot)$ is applied to confine the raw score in the range of $[0, 1]$, which can be interpreted as the risk score defined in Section 5.2.2, i.e., $s(X) = \sigma(g(X))$. Given the threshold $t \in [0, 1]$, the prediction of the sample X is determined as

$$\hat{Y} = \begin{cases} 0, & s(X) \leq t, \\ 1, & s(X) > t. \end{cases} \tag{5.3}$$

5.3.2 Fairness Definition

We define threshold invariant fairness [111, 112], a stronger condition on classification parity, in the context of DP and EO.

Definition 3 *(Threshold Invariant Fairness). Threshold Invariant Demographic Parity (TIDP) or Threshold Invariant Equalized Odds (TIEO) is achieved when DP or EO is satisfied, respectively, independent of the decision threshold t.*

We use the Calder-Verwer (CV) score [113] to measure classification parity. For simplicity, we denote that $P_a(\hat{Y}) := P(\hat{Y}|A = a)$ and $P_{a,y}(\hat{Y}) := P(\hat{Y}|A =$

$a, Y = y)$. The CV scores for DP and EO are defined as

$$\begin{aligned}\Delta\text{DP} =& \big|P_0(\hat{Y}=1) - P_1(\hat{Y}=1)\big|, \\ \Delta\text{EO} =& \frac{1}{2}(\big|P_{0,0}(\hat{Y}=1) - P_{1,0}(\hat{Y}=1)\big| + \big|P_{0,1}(\hat{Y}=1) - P_{1,1}(\hat{Y}=1)\big|).\end{aligned} \tag{5.4}$$

Note that the smaller the CV score is, the better the classifier achieves a fair classification. In the most ideal cases, the classification parity, DP or EO, is satisfied when ΔDP or ΔEO is zero.

In the following, we investigate the relationship between threshold invariant fairness and the risk distribution. Denote $f_a(s) := f(s|A = a)$ and $f_{a,y}(s) := f(s|A = a, Y = y)$ as the risk distributions over the group with the protected attribute A and label Y. Recall that the support of the risk score s is $[0, 1]$ and the decision threshold t ranges within $[0, 1]$. For the DP constraint, we have

$$\begin{aligned}P_a(\hat{Y}=1) = P_a(s(X) > t) = \textstyle\int_t^1 f_a(s)\mathrm{d}s, \\ P_a(\hat{Y}=0) = P_a(s(X) \leq t) = \textstyle\int_0^t f_a(s)\mathrm{d}s.\end{aligned} \tag{5.5}$$

Then,

$$\begin{aligned}\Delta\text{DP} &= \big|P_0(\hat{Y}=1) - P_1(\hat{Y}=1)\big| \\ &= \Big|\int_t^1 \big(f_0(s) - f_1(s)\big)\mathrm{d}s\Big| \\ &\leq \int_t^1 \big|f_0(s) - f_1(s)\big|\mathrm{d}s \leq \epsilon_{DP}(1-t).\end{aligned} \tag{5.6}$$

where we define the upper bound of the difference between two risk distributions $\big|f_0(s) - f_1(s)\big| \leq \epsilon_{DP}, \forall s \in [0, 1]$. Since $P_a(\hat{Y}=1) - P_a(\hat{Y}=0) = 1, \forall a \in \{0, 1\}$, we

have

$$\begin{aligned}\Delta\text{DP} &= \big|P_1(\hat{Y}=0) - P_0(\hat{Y}=0)\big| \\ &\leq \int_0^t \big|f_1(s) - f_0(s)\big|\mathrm{d}s \leq \epsilon_{DP} t.\end{aligned} \tag{5.7}$$

Combining (5.6) and (5.7), we have

$$\Delta\text{DP} \leq \epsilon_{DP} \cdot \min\{t, 1-t\} \leq \frac{1}{2}\epsilon_{DP}. \tag{5.8}$$

Remark 1 *(The sufficient condition of TIDP). When ϵ_{DP} reaches 0, DP is satisfied regardless of the decision threshold. Hence, equalizing the risk distributions between the positive and negative protected attributes, i.e., $f_0(s) = f_1(s)$, is the sufficient condition of TIDP.*

We denote $P_{a,y,\hat{Y}} := P(\hat{Y}=\hat{y}|A=a, Y=y)$ and $f_{a,y}(s) := f(s|A=a, Y=y)$ as the risk score of one sample and risk distributions over the group with protected attribute A and label Y, respectively. Given the decision threshold $t \in [0,1]$, we have

$$\begin{aligned}\Delta\text{EO} &= \frac{1}{2}\Big(\big|P_{0,0,1} - P_{1,0,1}\big| + \big|P_{0,1,1} - P_{1,1,1}\big|\Big) \\ &= \frac{1}{2}\Big(\big|\int_t^1 f_{0,0}(s) - f_{1,0}(s)\mathrm{d}s\big| + \big|\int_t^1 f_{0,1}(s) - f_{1,1}(s)\mathrm{d}s\big|\Big) \\ &\leq \frac{1}{2}\Big(\int_t^1 \big|f_{0,0}(s) - f_{1,0}(s)\big|\mathrm{d}s + \int_t^1 \big|f_{0,1}(s) - f_{1,1}(s)\big|\mathrm{d}s\Big) \\ &\leq \frac{1}{2}(\epsilon_0 + \epsilon_1)(1-t),\end{aligned} \tag{5.9}$$

where we define the upper bounds $\big|f_{0,0}(s) - f_{1,0}(s)\big| \leq \epsilon_0$ and $\big|f_{0,1}(s) - f_{1,1}(s)\big| \leq$

$\epsilon_1, \forall s \in [0,1]$. Also, we have

$$
\begin{aligned}
\Delta\text{EO} &= \frac{1}{2}\Big(\big|P_{0,0,1} - P_{1,0,1}\big| + \big|P_{0,1,1} - P_{1,1,1}\big|\Big) \\
&= \frac{1}{2}\Big(\big|\big(1 - P_{0,0,1}\big) - \big(1 - P_{1,0,0}\big)\big| + \big|\big(1 - P_{0,1,0}\big) - \big(1 - P_{1,1,0}\big)\big|\Big) \\
&= \frac{1}{2}\Big(\big|P_{1,0,0} - P_{0,0,0}\big| + \big|P_{1,1,0} - P_{0,1,0}\big|\Big) \\
&= \frac{1}{2}\Big(\big|\int_0^t f_{1,0}(s) - f_{0,0}(s)\mathrm{d}s\big| + \big|\int_0^t f_{1,1}(s) - f_{0,1}(s)\mathrm{d}s\big|\Big) \\
&\leq \frac{1}{2}\Big(\int_0^t \big|f_{1,0}(s) - f_{0,0}(s)\big|\mathrm{d}s + \int_0^t \big|f_{1,1}(s) - f_{0,1}(s)\big|\mathrm{d}s\Big) \\
&\leq \frac{1}{2}(\epsilon_0 + \epsilon_1)t.
\end{aligned}
\tag{5.10}
$$

Combining (5.9) and (5.10), for EO constraint, we have

$$
\Delta\text{EO} \leq \frac{1}{2}(\epsilon_0 + \epsilon_1) \cdot \min\{t, 1-t\} \leq \frac{1}{4}(\epsilon_0 + \epsilon_1). \tag{5.11}
$$

where we define the upper bounds $\big|f_{0,0}(s) - f_{1,0}(s)\big| \leq \epsilon_0$ and $\big|f_{0,1}(s) - f_{1,1}(s)\big| \leq \epsilon_1, \forall s \in [0,1]$.

Remark 2 *(The sufficient condition of TIEO). When ϵ_0 and ϵ_1 reach 0, EO is satisfied regardless of the decision threshold. Hence, equalizing the risk distributions of the positive/negative samples between positive and negative protected feature attributes, i.e., $f_{0,y}(s) = f_{1,y}(s),\ \forall y \in \{0,1\}$, respectively, is the sufficient condition of TIEO.*

Instead of equalizing statistical attributes, threshold invariant fairness requires equalization of risk distributions, which is a stricter condition than the family of classification parity.

5.4 Equalization of Risk Distribution

So far, we know that equalizing risk distributions across different groups is a sufficient condition to achieve threshold-invariant fairness. One straightforward post-processing approach is to find a mapping function to equalize the histogram of risk scores over one group to another group, a strategy similar to histogram matching in image processing [114]. Here, we focus on holistic systematic approaches to equalize risk distributions during the classifier training by formulating an appropriate fairness regularization.

In this section, we present two approaches to design the regularizer between two groups $\mathcal{D}_0$ and $\mathcal{D}_1$ in the whole sample set $\mathcal{D}$.

5.4.1 Approach 1: Histogram Approximation (HA)

Define $C = \{c_j\}_{j=1}^{N}$ as the bin centers of N-bin histogram with equal interval and the bin width $\delta = c_j - c_{j-1}$. The count of the samples from the group $\mathcal{D}_i$ in the bin $(c - \frac{\delta}{2}, c + \frac{\delta}{2})$ is expressed as

$$n_c = \sum_{X \in \mathcal{D}_i} \Pi_c(s(X)), \tag{5.12}$$

where $\Pi_c(\cdot)$ is a rectangular function:

$$\Pi_c(x) = \begin{cases} 1, & x \in (c - \frac{\delta}{2}, c + \frac{\delta}{2}), \\ 0, & \text{otherwise.} \end{cases} \tag{5.13}$$

Since $\Pi_c(\cdot)$ is not differentiable, we approximate it with a Gaussian kernel,

$$n_c = \sum_{X \in \mathcal{D}_i} G_c(s(X)) = \sum_{X \in \mathcal{D}_i} \exp(-\frac{(s(X) - c)^2}{2\sigma_c^2}). \tag{5.14}$$

Hence, the histogram of the risk score in $\mathcal{D}_i$ can be expressed as

$$h(\mathcal{D}_i) = normalize\big([n_{c_1}, n_{c_2}, ..., n_{c_N}] \big), \tag{5.15}$$

where $normalize(\cdot)$ scales the histogram such that the histogram entries sum to one. Note that $h(\mathcal{D}_i)$ is an N-D vector since the histogram has N bins. The HA method formulates a differentiable histogram via a Gaussian kernel to construct the risk distribution.

We use the symmetric combination of Kullback–Leibler (KL) divergence to evaluate the distance of the risk distributions between two groups $\mathcal{D}_0$ and $\mathcal{D}_1$.

$$\begin{aligned} d\big(r(\mathcal{D}_0), r(\mathcal{D}_1)\big) =& D_{KL}\big(h(\mathcal{D}_0)||h(\mathcal{D}_1)\big) + D_{KL}\big(h(\mathcal{D}_1)||h(\mathcal{D}_0)\big) \\ =& \sum_{i=1}^{N} \big(h_{(i)}(\mathcal{D}_0) - h_{(i)}(\mathcal{D}_1)\big) \log \frac{h_{(i)}(\mathcal{D}_0)}{h_{(i)}(\mathcal{D}_1)}. \end{aligned} \tag{5.16}$$

where subscript (i) denotes the i-th entry of the histogram. It can be seen that the distance d is non-negative and approaches zero when two risk distributions are equal, i.e., $r(\mathcal{D}_0) = r(\mathcal{D}_1)$.

5.4.2 Approach 2: Gaussian Assumption (GA)

Since the risk scores are the sigmoid of the raw scores, i.e., $s(X) = \sigma(g(X))$, equalizing the distribution of raw scores would lead to the equalization of the risk distribution. Given two groups $\mathcal{D}_0$ and $\mathcal{D}_1$, we assume that the distributions of raw scores obey the same Gaussian distribution over $\mathcal{D}_0$ and $\mathcal{D}_1$, and the mean and variance are sufficient to characterize the distributions. We estimate means and variances of the distributions of raw score over $\mathcal{D}_0$ and $\mathcal{D}_1$ using maximum likelihood estimate:

$$\mathcal{D}_i : \begin{cases} \hat{\mu}_i = \frac{1}{|\mathcal{D}_i|} \sum_{X \in \mathcal{D}_i} g(X) \\ \\ \hat{\sigma}_i^2 = \frac{1}{|\mathcal{D}_i|} \sum_{X \in \mathcal{D}_i} \big(g(X) - \hat{\mu}_i\big)^2 \end{cases}, \ i \in \{0, 1\}. \tag{5.17}$$

Recall that KL divergence of two 1-D Gaussian distributions $\mathcal{N}_0(\mu_0, \sigma_0^2)$ and $\mathcal{N}_1(\mu_1, \sigma_1^2)$ is

$$D_{KL}\big(\mathcal{N}_0 || \mathcal{N}_1\big) = \frac{1}{2}\big(\log \frac{\sigma_1^2}{\sigma_0^2} + \frac{(\mu_0 - \mu_1)^2 + \sigma_0^2}{\sigma_1^2} - 1\big). \tag{5.18}$$

Based on (5.18), we define the distance between two risk distributions $r(\mathcal{D}_0)$ and $r(\mathcal{D}_1)$ as

$$\begin{aligned} d\big(r(\mathcal{D}_0), r(\mathcal{D}_1)\big) &= D_{KL}\big(\mathcal{N}_0 || \mathcal{N}_1\big) + D_{KL}\big(\mathcal{N}_1 || \mathcal{N}_0\big) \\ &= \frac{1}{2}\big(\frac{(\hat{\mu}_0 - \hat{\mu}_1)^2 + \hat{\sigma_0}^2}{\hat{\sigma}_1^2} + \frac{(\hat{\mu}_1 - \hat{\mu}_0)^2 + \hat{\sigma}_1^2}{\hat{\sigma}_0^2} - 2\big). \end{aligned} \tag{5.19}$$

Similar to (5.16), the distance d is non-negative and approaches zero when the means and variances from $\mathcal{D}_0$ and $\mathcal{D}_1$ are equal, i.e., $\hat{\mu}_0 = \hat{\mu}_1$ and $\hat{\sigma}_0^2 = \hat{\sigma}_1^2$.

Note that the distance d in both approaches discussed in this section are differentiable, which can facilitate the classifier optimization using gradient-based methods.

5.4.3 Classifier Formulation

For simplicity, we assume that the sample set $\mathcal{D}$ has a binary protected attribute, i.e., $|A| = 2$. To distinguish the DP and EO cases, we name the regularizers as $E_{f,DP}$ and $E_{f,EO}$, respectively. For DP case, $\mathcal{D}$ is splited into two groups: negative protected attribute $\mathcal{D}_0$ and positive protected attribute $\mathcal{D}_1$. The regularizer for this case $E_{f,DP}$ is defined as

$$E_{f,DP}(\mathcal{D}) = d(r(\mathcal{D}_0), r(\mathcal{D}_1)). \tag{5.20}$$

For EO case, $\mathcal{D}$ is splited into four groups: negative and positive samples with negative protected attribute, $\mathcal{D}_{0,n}$ and $\mathcal{D}_{0,p}$, respectively, and negative and positive samples with positive protected attribute, $\mathcal{D}_{1,n}$ and $\mathcal{D}_{1,p}$, respectively. The regularizer for this case $E_{f,EO}$ is defined as

$$E_{f,EO}(\mathcal{D}) = \sum_{y \in \{n,p\}} d\big(r(\mathcal{D}_{0,y}), r(\mathcal{D}_{1,y})\big). \tag{5.21}$$

Note that the regularizer can be extended to the case of multiple protected attributes, i.e., $|A| > 2$.

As a proof-of-concept, we consider two kinds of classifiers: logistic regression

(LR) and support vector machine (SVM). It is worthwhile to note that our proposed method can be easily extended to other complex differentiable classifiers. Define the classifier $g(x) = w^{\mathrm{T}}x + b$, and the risk score is computed as $s(x) = \sigma(g(x))$.

LR Model: The loss function L_{LR} for logistic regression can be expressed as

$$L_{\mathrm{LR}}(w, b) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} L_{ce}(s(X), Y) + \eta E_f(\mathcal{D}), \tag{5.22}$$

where $L_{ce}(\cdot)$ denotes the cross-entropy loss of the risk score and the label, and η is a positive hyperparameter. Since (5.22) is differentialable, we can use gradient-based methods to minimize (5.22) with respect to w and b.

Linear SVM Model: Pegasos was proposed to solve SVM with the hinge loss version using a sub-gradient solver [115], i.e.,

$$L_{\mathrm{Pegasos}}(w, b) = \frac{\lambda}{2}||w||^2 + \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \max\{0, 1 - Y \cdot g(X)\}, \tag{5.23}$$

where λ is a positive hyperparameter, and the label $Y \in \{-1, 1\}$ in an SVM denotes negative/positive label.

By incorporating the fairness regularizer into (5.23), we obtain the loss function as

$$L_{\mathrm{LSVM}}(w, b) = \frac{\lambda}{2}||w||^2 + \eta E_f(\mathcal{D}) + \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \max\{0, 1 - Y \cdot g(X)\}, \tag{5.24}$$

where λ and η are positive hyperparameters. Similar to Pegasos in (5.23), we can apply sub-gradient solvers, due to the sub-differentiability of (5.24).

Kernel SVM Model: In a kernel SVM, we use a mapping function $\phi(\cdot)$ to transform the original features to a higher dimensional subspace. Substituting original features X with the mapped features $\phi(X)$, we can rewrite (5.24) as

$$L_{\text{KSVM}}(w, b) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \max\{0, 1 - Y \cdot g(\phi(X))\} + \frac{\lambda}{2}||w||^2 + \eta E_f(\phi(\mathcal{D})). \quad (5.25)$$

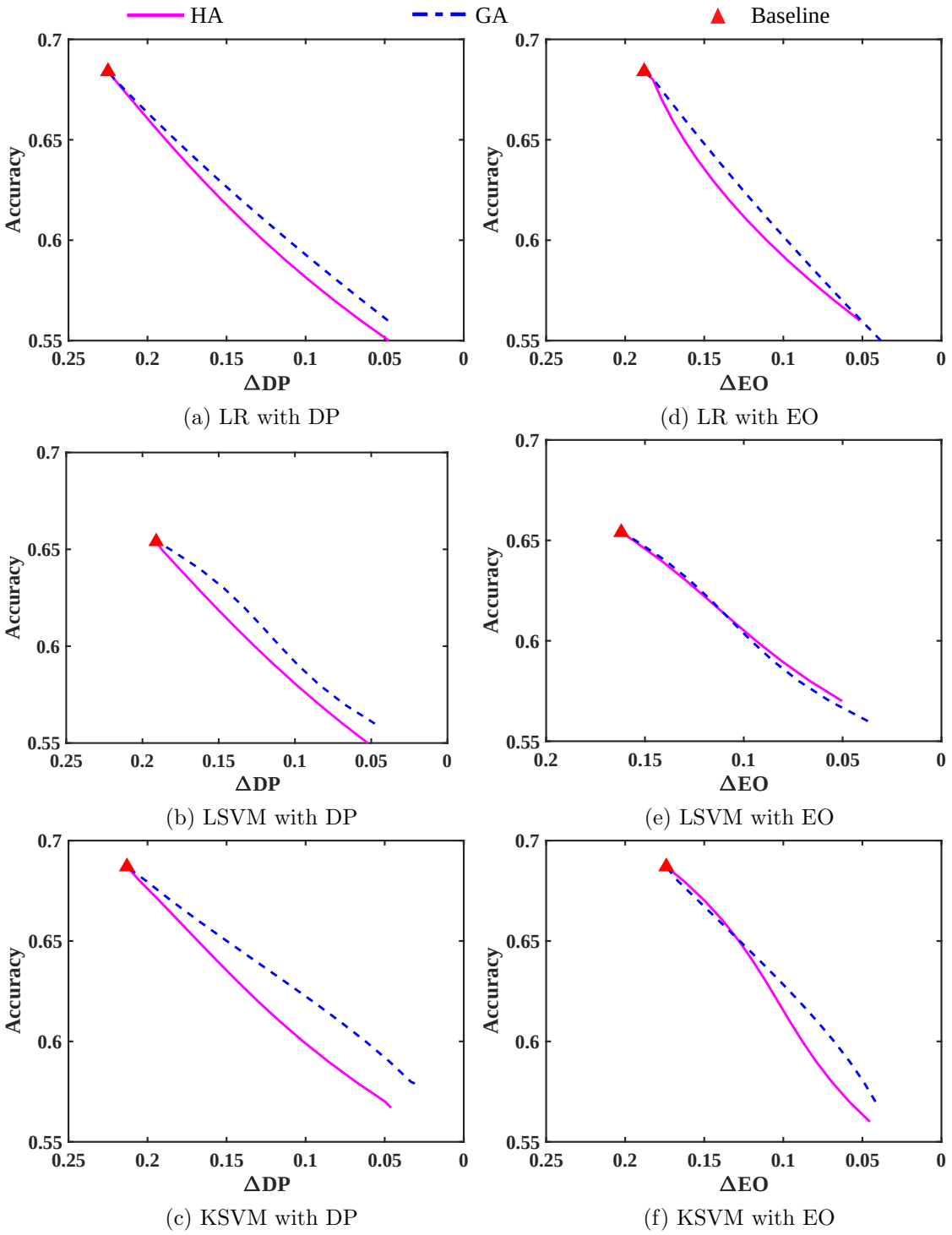
According to the Representer Theorem [116], the optimal solution of (5.25) can be spanned by the training samples, i.e., it is of the form $w = \sum_{i=1}^{|\mathcal{D}|} \alpha_i \phi(X_i)$. Hence, the kernel SVM predictor becomes $\tilde{g}(x) = \sum_{i=1}^{|\mathcal{D}|} \alpha_i \phi(X_i)^{\mathrm{T}} \phi(x) + b = \sum_{i=1}^{|\mathcal{D}|} \alpha_i K(X_i, x) + b$, where we define the kernel operator $K(X_i, x) = \phi(X_i)^{\mathrm{T}} \phi(x)$. Equation (5.25) can be rewritten with respect to α as

$$L_{\text{KSVM}}(\alpha, b) = \frac{\lambda}{2} \sum_{i,j=1}^{|\mathcal{D}|} \alpha_i \alpha_j K(X_i, X_j) + \eta E_f(\phi(\mathcal{D})) + \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \max\{0, 1 - Y \cdot \tilde{g}(X)\}, \quad (5.26)$$

where λ and η are positive hyperparameters. To calculate the fairness regularizer, we can use the kernel trick to compute the risk scores and the risk distributions over the mapped sample set $\phi(\mathcal{D})$. Similarly, we can use subgradient-based methods to optimize the loss function (5.26) with respect to α.

5.5 Experimental Results

In this section, we carry out experiments on the COMPAS risk assessment dataset compiled by ProPublica [34] and evaluate how well our proposed method can alleviate the problem of threshold variant fairness in the prior art of classification



(a) LR with DP

(d) LR with EO

(b) LSVM with DP

(e) LSVM with EO

(c) KSVM with DP

(f) KSVM with EO

Figure 5.3: Tradeoff between model accuracy and classification parity index: (a) LR with the DP constraint, (b) LSVM with the DP constraint, (c) KSVM with the DP constraint, (d) LR with the EO constraint, (e) LSVM with the EO constraint, and (f) KSVM with EO constraint. The red filled triangles indicate the baseline cases in each classifier.

parity.

5.5.1 Dataset and Feature Processing

ProPublica compiled a list of all criminal defendants screened through the COMPAS tool in Broward County, Florida, during 2013 to 2014.[1] The dataset contains defendants' records of prison times, demographics (e.g., gender, race, and age), criminal histories (current charge type, charge degree, and number of prior crimes), the COMPAS risk scores, and the ground truth of recidivism within two years after the screening. Previous discussions on bias decisions from machine learning on this dataset can be found in [33, 117, 118].

For simplicity, we only consider a subset of defendants whose race was either African-American or Caucasian, which is the protected attribute in our study. The set of features used in the classification task is summarized in Table 5.1. We only use the protected attribute, "race", as an indicator of the groups in the training and exclude it from the features for the classification task. The features are a combination of continuous and categorical features. For continuous features, we subtract their mean and scale them to unit variance; for categorical features, we use 0/1 to encode the features, expect for the labels in SVM where $-1/+1$ encoding is used. The dataset was randomly split into training (70%) and test (30%).

[1] The dateset can be downloaded from `https://github.com/propublica/compas-analysis`.

Feature	Type	Preprocessing
Age	Continuous	Normalization
Gender	Binary	0-1 encoding
Race (Protected)	Binary	0-1 encoding
Priors count	Continuous	Normalization
Charge degree	Binary	0-1 encoding
Recidivism (Label)	Binary	0-1 encoding

Table 5.1: Preprocessed Features Used in COMPAS Data

5.5.2 Tradeoff Between Accuracy and Fairness

We have trained three types of classifiers: LR, linear SVM (LSVM), and kernel SVM (KSVM). For SVM, we set hyperparameter $\lambda = \frac{1}{10 \cdot |\mathcal{D}|}$. We used radial basis function (RBF) kernel with $\gamma = 0.5$ in KSVM, i.e., $K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||_2^2)$. We applied gradient descent to optimize the models with momentum 0.9 and learning rate starting from 0.1 to 0.0001.

We tuned η from 0 to 5 in each classifier to show the tradeoff curve of accuracy versus fairness in Figure 5.3 via the two proposed equalization approaches: histogram approximation (HA) and gaussian assumption (GA). Note that the classifiers with $\eta = 0$ are the unconstrained classifiers, considered as *baseline*.

As shown in Figure 5.3, we can see that our proposed methods are effective at alleviating disparate learning performances across the groups. The CV scores of ΔDP and ΔEO reduce from around 0.2 to 0.05, with the help of the proposed fairness regularizers, whereas the resulting accuracies drop by approximately 0.1 on a $[0, 1]$ scale. Generally, our proposed methods can be used in different types of classifiers (e.g., LR, LSVM, and KSVM) and different types of classification parity

indices (e.g., DP and EO). Comparing two equalization methods of HA and GA, the models with GA based fairness regularizer have slightly better performances than those with the HA approach in terms of the accuracy cost to achieve the same fairness level.

5.5.3 Visualization of Risk Distribution

In this section, we use linear regression classifiers as an example to visualize the influence of the proposed fairness regularizers on risk distributions. Figure 5.4 and 5.5 shows the risk distributions of this type of classifiers in three example cases: with no constraints (baseline), medium constraints, and large constraints. For the DP constraint, we present the risk distributions of the African-American group and the Caucasian group; for the EO constraint, we show the risk distributions of African-American recidivists, African-American non-recidivists, Caucasian recidivists, and Caucasian non-recidivists, respectively.

Several observations can be made from Figure 5.4 and 5.5.

- The distributions from the baseline classifiers are distinct among the groups, suggesting the discriminating decision performances for different groups. The unconstrained classifiers suffer from some bias in their predictions.

- Our proposed methods progressively equalize the risk distributions among the groups as we weigh more on fairness regularizers. When we impose medium constraints, the corresponding risk distributions become closer in shape than that in the baseline cases. When we keep increasing the constraints to a signifi-

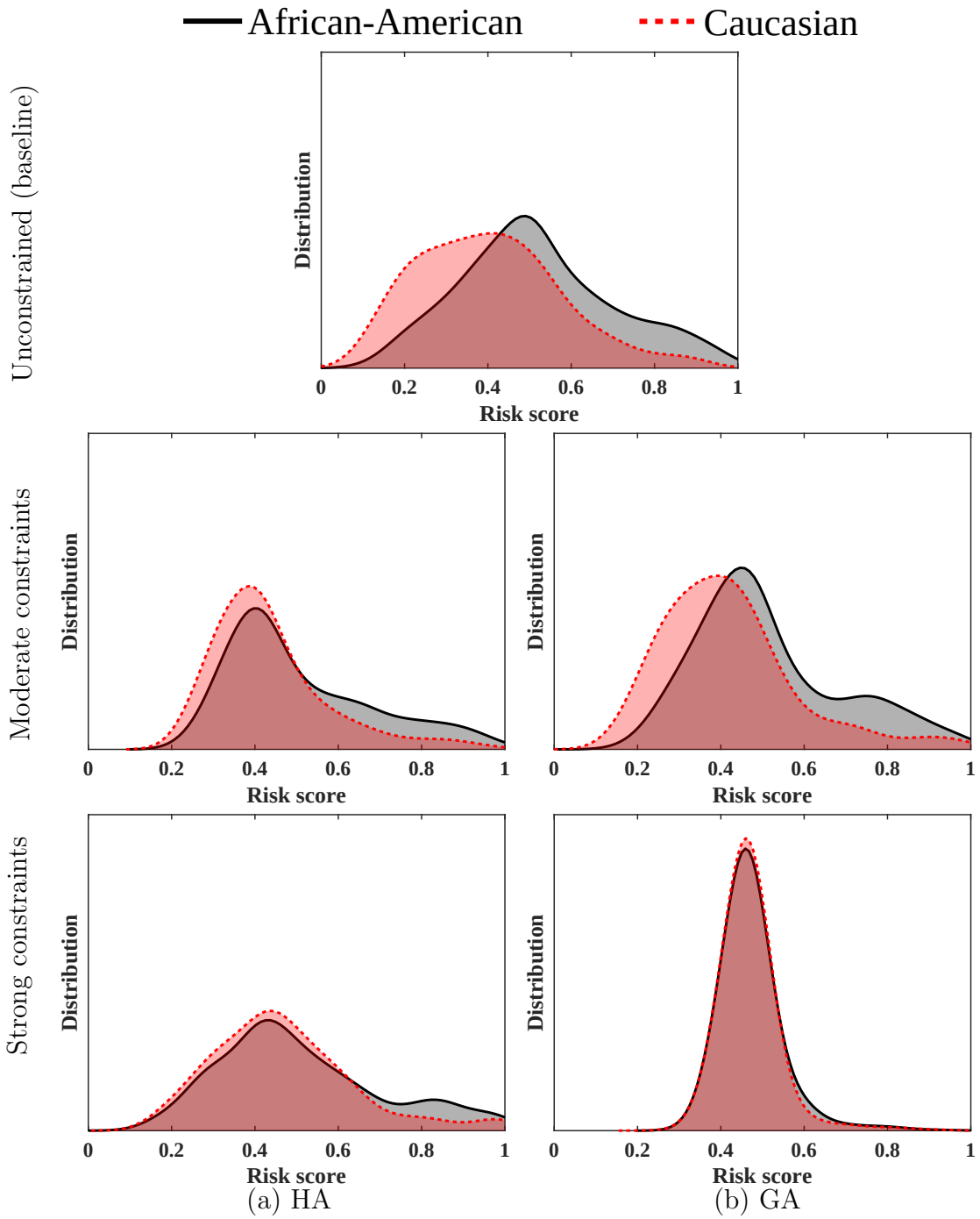


Figure 5.4: Risk distributions of LR classifiers with different levels of DP fairness regularizers. The first row is the baseline, i.e., unconstrained classifier; the second row is the classifier with moderate constraints; the third row is the classifier with strong constraints. (a) HA and (b) GA.

cant level, the corresponding risk distributions almost overlap with each other, indicating the effectiveness of the regularizer to equalize the risk distributions.

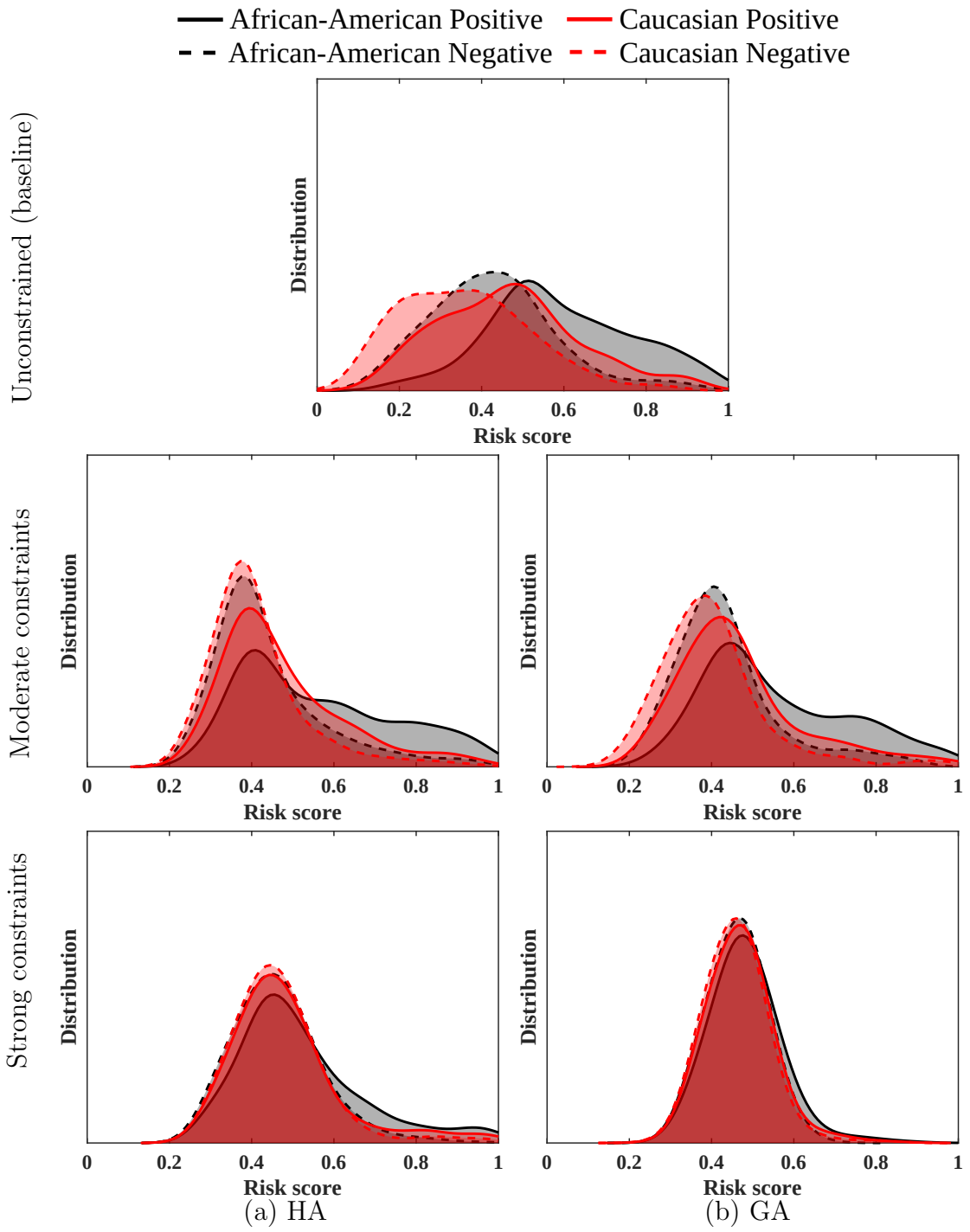


Figure 5.5: Risk distributions of LR classifiers with different levels of EO fairness regularizers. The first row is the baseline, i.e., unconstrained classifier; the second row is the classifier with moderate constraints; the third row is the classifier with strong constraints. (a) HA and (b) GA.

5.5.4 Parity Variation Versus Threshold

We conducted experiments on three types of classifiers (e.g., LR, LSVM, and KSVM) with 2 equalization approaches (e.g., HA, GA). In this section, we name the proposed classifiers as "{classifier type}-{equalization}". For instance, "LR-HA" refers to LR classifier with HA approach. We compare our proposed approaches with the prior works based on DP constraint [105] and EO constraint [104]. We refer to them as "Zafar-DP" and "Zafar-EO" here, respectively. The baseline refers to the unconstrained LR classifier. Note that all these methods do not use protected attributes to make predictions in the test stage. We show the variation of classification parity versus the decision threshold of the risk score for a given classifier. For fair comparison, we tune the parameter η to make the proposed classifiers have comparable performances to the prior works, in terms of prediction accuracy. Performance accuracy and the CV scores of ΔDP and ΔEO come from the classifiers with the default decision threshold.

To investigate the parity variation versus the decision threshold, we present the variation of the classification parity by changing the decision threshold ranging from 0.3 to 0.7. Table 5.2 presents the performances of the classifiers with DP constraint and Table 5.3 presents the performances of the classifiers with EO constraint. "Interval" refers to the interval length of the parity variation and "STD" denotes the standard deviation. Figure 5.6 illustrates the variations of classification parity versus the risk score threshold ranging from 0.3 to 0.7. Less fluctuation and more flatness of the curve suggest that the classifier achieves higher threshold-invariant

parity.

As we expect, the baseline classifier has the best performance in terms of accuracy (68.4%), but worst in CV scores of classification parity (ΔDP= 0.225 and ΔEO= 0.188). It spans a wider range and has a larger variance in parity when decision threshold ranges from 0.3 to 0.7. The variation curves in Figure 5.6 are also above the classifiers with fairness constraints. With the fairness constraints during the training, the CV scores are both reduced in Table 5.2 and 5.3 at the cost of the accuracy performances, and variation ranges and variances decrease to a lower level, indicating the better equitable decision performances among the groups in terms of classification parity. For the DP constraint in Table 5.2, the proposed classifiers have slightly better DP consistency against the change of decision threshold than Zafar-DP, according to the SD of ΔDP. For the EO constraint in Table 5.3, the proposed classifiers have a noticeable advantage in invariant EO against the change of decision threshold. Compared with Zafar-EO, the proposed classifiers have smaller interval ranges and smaller variances of ΔEO.

	Accuracy	**ΔDP**	**Interval**	**STD**
Baseline	68.4%	0.225	0.145	0.044
Zafar-DP	57.4%	0.060	0.079	0.024
LR-HA	56.6%	0.075	0.089	0.024
LR-GA	56.9%	0.066	0.093	0.023
LSVM-HA	56.2%	0.035	0.073	0.022
LSVM-GA	56.5%	0.048	0.097	0.020
KSVM-HA	57.0%	0.059	0.057	0.014
KSVM-GA	58.4%	0.064	0.098	0.024

Table 5.2: Performance Under DP Constraint

From Figure 5.6, we can observe that our proposed methods have minor fluctu-

	Accuracy	**ΔEO**	**Interval**	**STD**
Baseline	68.4%	0.188	0.123	0.037
Zafar-EO	63.2%	0.120	0.154	0.047
LR-HA	62.7%	0.117	0.077	0.024
LR-GA	62.9%	0.117	0.098	0.028
LSVM-HA	63.3%	0.137	0.109	0.029
LSVM-GA	63.0%	0.129	0.062	0.018
KSVM-HA	63.2%	0.112	0.067	0.012
KSVM-GA	62.9%	0.096	0.040	0.009

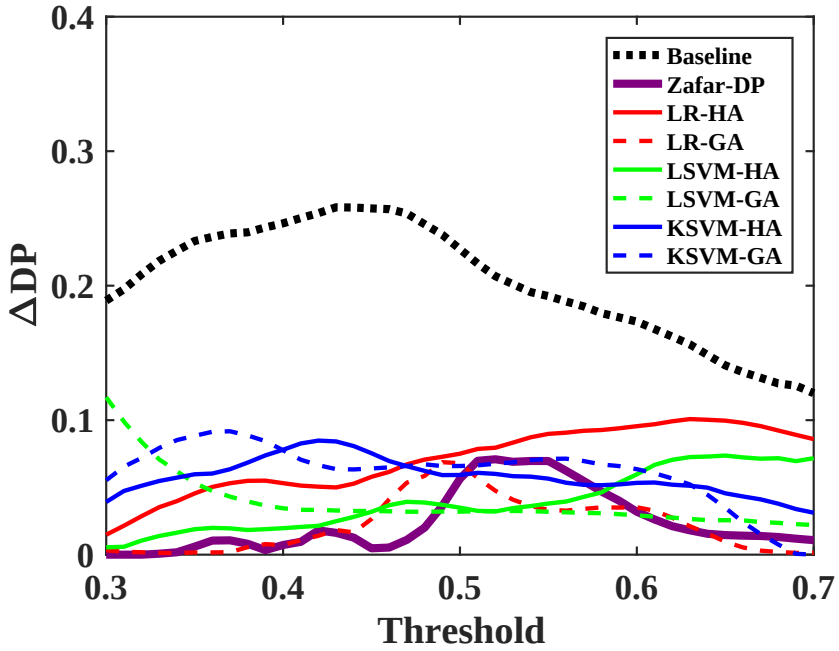
Table 5.3: Performance Under EO Constraint

ations on the classification parity curves, compared with Zafar-DP [105] and Zafar-EO [104], respectively. Slightly modifying the decision threshold does not affect the classification parity to a large extent in the proposed classifiers. This suggests that our proposed constraints are effective in reducing the sensitivity of classifiers to the decision threshold in terms of classification parity. The noticeable drops in the CV scores in the three types of the classifiers demonstrate that our methods can be applied to a variety of classifiers.

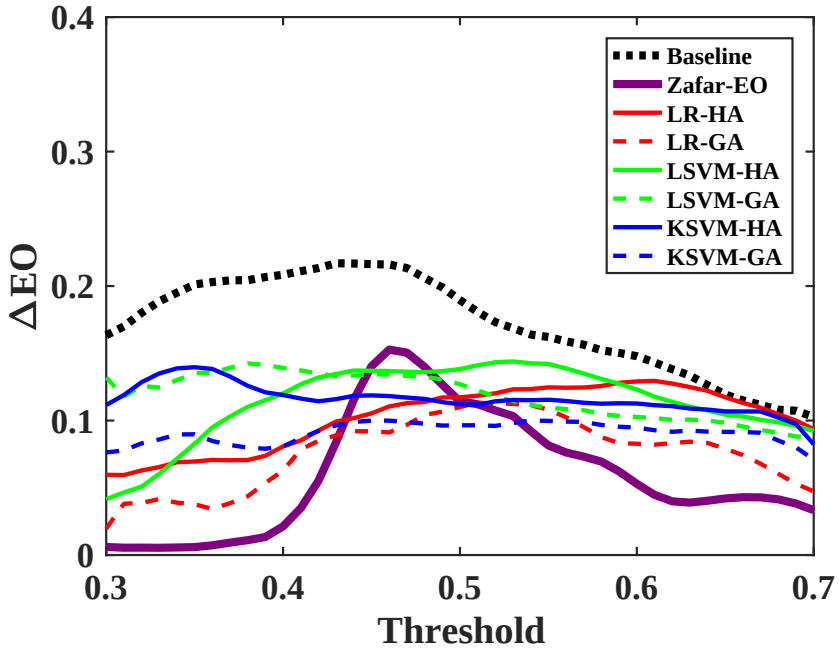
5.5.5 Discussion

From the experimental results, we can see that two proposed methods are effective in equalizing the risk distributions so as to achieve threshold invariant fairness. In this section, we discuss and summarize the difference between the two methods.

The HA method uses an N-bin histogram to compute the risk distribution. To make the histogram operation differentiable, a Gaussian kernel is used to approximate the accumulation process in each histogram bin. The HA method requires two



(a)



(b)

Figure 5.6: Variations of classification parity versus decision threshold: (a) classifiers with the DP constraint and (b) classifiers with the EO constraint. The figures are best viewed in color.

hyperparameters: the number of bins N and the variance σ_c^2 in (5.14). Their settings influence the computational cost and the precision of the distribution approxima-

tion. The HA method computes the weight of every sample to each histogram bin via a Gaussian kernel and then sum up the weights in each bin to construct the risk distribution. The overall computational complexity of computing risk distribution in HA is $O(MN)$, where M is the number of training samples.

The GA method assumes the Gaussian distribution and uses only the statistics of the first and second moments (mean and variance) to characterize the risk distribution. The estimation of means and variances has a computational complexity of $O(M)$, which is lower than the HA method. Considering the similar performances between the two methods and the difference in computational complexities, we can see that the GA method is a preferred approach to achieve the equalization of risk distributions.

In addition, our proposed methods can be extended to the case of multiple protected attributes and other complex differentiable classifiers. For example, the proposed methods are compatible with neural network classifiers. To make our methods adaptive to the batch-based training of neural networks, we can modify gradient descent optimizer to batch-based optimizer (e.g., stochastic gradient descent) and calculate the risk distributions over one training batch instead of over the whole training set.

5.6 Chapter Summary

In this chapter, we have introduced a novel fairness notion in machine learning models. Different from the prior definition of classification parity, the new notion

aims to build fair classifiers that are not sensitive to the decision thresholds, and achieves this by equalizing the risk distributions among different groups. We perform distribution equalization using histogram approximation and Gaussian assumption, respectively, and incorporate them into logistic regression and SVM classifiers. Experimental results show that our fairness notion allows better control of the fairness level against the decision threshold of the classifier, and has broad compatibility to multiple types of machine learning classifiers.

Chapter 6

Conclusion and Future Perspectives

This dissertation has studied three emerging types of AI-based sensing and learning systems and their potential risks and security issues in our social activities, beyond their excellent learning performance in completing practical tasks.

First, we investigate intellectual property protection and access control for DNNs widely used in commercial products and services. DNNs can be considered as emerging intellectual property to encourage healthy business investment and competition cycles. Unlike owner authentication schemes through post-piracy investigation, our proposed access-control design in model training can actively block piracy and illicit use when the model is accessed from unauthorized users. Specifically, this design can make the DNN model work properly if the authorized user transforms the input with the correct key beforehand, while the input without correct authorized transformation leads to prediction performance of low accuracy. We establish threat models considering multiple scenarios and investigate possible attacks on the access-control design to improve the resistance of the proposed method. The experimental results show the effectiveness of the proposed access-control design and its resilience against several attacks and threats.

This design framework can serve as a generic framework to enable the access control of DNNs. The protective transform module is a lightweight module distributed to authorized users. When the authorized users want to obtain the DNN outputs, they can first transform the input images using the protective transform module with the correct key and then feed the transformed images to the DNN to get accurate predictions. The protective transform module can be executed in a trusted platform module to increase the security level. The corresponding DNN can run on any off-the-shelf platform since we do not require additional hardware or modify the DNN structures for the DNN model part.

We, so far, have investigated perturbation transformation in protective transform module – additive perturbation and spatial/warping perturbation. Other transformations or a combination of transformations can be studied in the protective transform module to enrich the transformation methods and enhance the difficulty of the adversary's attack. To improve the resilience against attacks, we can further study the threat model and consider more adversarial scenarios during the access-control design.

We also introduce a potential application – a privacy-aware smart camera. We encapsulate the protective transform module into the camera, so that only pictures taken from the smart camera with the correct key can be authorized to have accurate predictions. Other practical applications that can benefit from our proposed protection framework may arise in the near future.

The second part of the dissertation has investigated the privacy issues in an emerging sensing technology, namely, remote photoplethysmography (rPPG), which

can extract a person's physiological signal in facial videos recorded by an RGB camera. Over the past decade, rPPG technology has been prospering, enabling the continuous monitoring of vital signs, such as HR, using regular mass-market digital cameras in daily life. The rPPG technology modality is associated with two privacy risks. First, some hackers can disclose the physiological information from the facial videos recorded during online conferences and video conversations without the person's consent, which raises physiological privacy issues. Second, rPPG technologies usually require facial videos as the main inputs. The facial videos are sensitive and contain identifiable appearance features, which raises identity privacy issues.

We address the physiological privacy risk by developing *PulseEdit* to conceal the true physiological signal trace in facial videos. The proposed algorithm can offer the users two options of editing operations on the physiological signal in facial videos regarding physiological privacy protection. PulseEdit also provides a capability of robustly concealing the physiological signal from videos with both static scenarios and head motion scenarios (*e.g.*, talking, translation, and rotation).

Since PulseEdit can edit the physiological signals in facial videos, it is possible to employ PulseEdit as an adversarial operation to rPPG-based visual security algorithms. As a proof-of-concept, we have studied to what extent PulseEdit can circumvent rPPG-based liveness detection and deepfake detection algorithms. The experimental results demonstrate noticeable performance drops between the original videos and the PulseEdit videos, indicating that PulseEdit can successfully mitigate the rPPG-based visual security algorithms. From the perspective of threat modeling for these visual security algorithms, our PulseEdit research suggests that it is cru-

cial to investigate this and other similar vulnerabilities and improve the rPPG-based visual security algorithms against potential adversarial operations.

In the current form as a proof-of-concept, we focus on altering the rPPG information in each frame and have not explicitly considered concealing the manipulation traces. Forensic tools such as steganalysis can detect the presence of perturbation from the uncompressed frames if available. Nevertheless, we find that lossy video compression is a possible step to improve the resistance of the edited frames against forensic analysis and retain the edited rPPG signal in the video. We can extend the proposed algorithm by investigating the inclusion of various forensic detectability criteria into the algorithm to gain insights into its ability as an anti-forensic tool, and the competing direction of detecting the manipulations made by PulseEdit. In addition, the current form of PulseEdit perturbs the facial pixels independently, and the future algorithm can take spatial and temporal correlations of facial pixels into consideration for the pixel perturbation to further minimize the perceptual distortion of facial videos.

To address the identity privacy risk, we propose an anonymization transform to remove the person's appearance features but retain the rPPG signals for the rPPG analysis. We first standardize the face to a generic face template to remove the facial contour and proportional features. Then, spatial decomposition is applied to the standardized face, and we reconstruct the face using specific layers of the decomposition for different privacy strengths. The proposed transform can be used to process the facial videos before uploading them to the rPPG analysis software to protect identity privacy. More importantly, it can encourage the sharing of facial

video data by the R&D community of rPPG technology to foster advancement in this area, with few concerns of identity privacy risks.

The third part of the dissertation has studied group fairness in machine learning models. The prior art on the definition and algorithms of group fairness mainly focused on equitable decision performances across the groups from a decision probability perspective, such as the positive proportion of decisions, true positive rate, false positive rate, and other statistics alike. Such parity of the group-wise performances is, however, generally not retained when we tune the decision threshold of the classifier.

To address the above disadvantage, we have introduced a novel fairness notion to build fair classifiers that are not sensitive to the decision thresholds. We achieve this by equalizing the risk distributions among different groups of populations. We perform distribution equalization using two approximation methods using histogram approximation and Gaussian assumption, respectively. Experimental results have shown that our fairness notion allows better control of the fairness level against the decision threshold of the classifier and has broad compatibility to multiple types of machine learning classifiers, such as logistic regression and SVM classifiers.

Several extensions of our work fairness can be explored in the future. The first is the extension to the case of multiple protected attributes. Our work focuses on single binary protected attribute, while multiple protected attributes are more common in real-world applications and social activities. One challenge here is how to formulate the distribution equalization on multiple protected attributes. The second extension is to more complex classifiers, for example, neural network classifiers.

Deep neural networks have been widely used in recent machine learning models. Extending the fairness notion to deep model training can benefit the practical application of deep learning models. One possible solution to making our methods adaptive to the batch-based training of neural networks is to modify the commonly used gradient descent optimizer to a batch-based optimizer (e.g., stochastic gradient descent) and calculate the risk distributions over one training batch instead of over the whole training set. Furthermore, we can investigate the influence of distribution equalization in batch on the risk distribution of the whole dataset.

In the advent of the Internet era, AI-based sensing and learning systems have been playing an increasingly important role in our social activities. Beyond their excellent performance to facilitate our daily life, they also bring about such important issues as security, privacy, and fairness in our society. Therefore, when we develop new algorithms and systems, it is crucial to proactively incorporate security, privacy, and fairness perspectives, making new AI algorithms and systems increasingly better and safer for practical usage and social activities.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, pages 1097–1105, 2012.

[3] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[4] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018.

[5] Samet Akcay, Mikolaj E Kundegorski, Chris G Willcocks, and Toby P Breckon. Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery. *IEEE Transactions on Information Forensics and Security*, 13(9):2203–2215, 2018.

[6] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[7] Yaron Gurovich, Yair Hanani, Omri Bar, Guy Nadav, Nicole Fleischer, Dekel Gelbman, Lina Basel-Salmon, Peter M. Krawitz, Susanne B. Kamphausen, Martin Zenker, Lynne M. Bird, and Karen W. Gripp. Identifying facial phenotypes of genetic disorders using deep learning. *Nature Medicine*, 25:60–64, 2019.

[8] Awni Y. Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H. Tison, Codie Bourn, Mintu P. Turakhia, and Andrew Y. Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25:65–69, 2019.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Wim Verkruysse, Lars O Svaasand, and J Stuart Nelson. Remote plethysmographic imaging using ambient light. *Optics Express*, 16(26):21434–21445, 2008.

[13] Gerard De Haan and Vincent Jeanne. Robust pulse rate from chrominance-based rPPG. *IEEE Trans. Biomed. Eng.*, 60(10):2878–2886, 2013.

[14] Guha Balakrishnan, Fredo Durand, and John Guttag. Detecting pulse from head motions in video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3430–3437, 2013.

[15] Wenjin Wang, Albertus C den Brinker, Sander Stuijk, and Gerard de Haan. Algorithmic principles of remote PPG. *IEEE Trans. Biomed. Eng.*, 64(7):1479–1491, 2016.

[16] Xiaobai Li, Jie Chen, Guoying Zhao, and Matti Pietikainen. Remote heart rate measurement from face videos under realistic situations. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4264–4271, 2014.

[17] Sergey Tulyakov, Xavier Alameda-Pineda, Elisa Ricci, Lijun Yin, Jeffrey F. Cohn, and Nicu Sebe. Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[18] Radim Špetlík, Vojtech Franc, and Jirí Matas. Visual heart rate estimation with convolutional neural network. In *British Machine Vision Conference*, 2018.

[19] Qiang Zhu, Mingliang Chen, Chau-Wai Wong, and Min Wu. Adaptive multi-trace carving based on dynamic programming. In *Asilomar Conference on Signals, Systems, and Computers*, pages 1716–1720, 2018.

[20] Xuesong Niu, Shiguang Shan, Hu Han, and Xilin Chen. RhythmNet: End-to-end heart rate estimation from face via spatial-temporal representation. *IEEE Trans. Image Process.*, 29:2409–2423, 2019.

[21] Qiang Zhu, Mingliang Chen, Chau-Wai Wong, and Min Wu. Adaptive multi-trace carving for robust frequency tracking in forensic applications. *IEEE Trans. Inf. Forensics Security*, 16:1174–1189, 2020.

[22] Mingliang Chen, Qiang Zhu, Harrison Zhang, Min Wu, and Quanzeng Wang. Respiratory rate estimation from face videos. In *IEEE EMBS Conf. on Biomedical and Health Informatics*, pages 1–4, 2019.

[23] Mingliang Chen, Qiang Zhu, Min Wu, and Quanzeng Wang. Modulation model of the photoplethysmography signal for vital sign extraction. *IEEE J. Biomed. and Health Informatics*, 2020.

[24] Ming-Zher Poh, Daniel J McDuff, and Rosalind W Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE Trans. Biomedical Engineering*, 58(1):7–11, 2010.

[25] Daniel J McDuff, Javier Hernandez, Sarah Gontarek, and Rosalind W Picard. COGCAM: Contact-free measurement of cognitive stress during computer tasks with a digital camera. In *CHI Conf. on Human Factors in Computing Systems*, pages 4000–4004, 2016.

[26] Bing-Fei Wu, Chun-Hsien Lin, Po-Wei Huang, Tzu-Min Lin, and Meng-Liang Chung. A contactless sport training monitor based on facial expression and remote-PPG. In *IEEE Conf. on Systems, Man, and Cybernetics*, pages 846–851, 2017.

[27] Siqi Liu, Pong C Yuen, Shengping Zhang, and Guoying Zhao. 3D mask face anti-spoofing with remote photoplethysmography. In *European Conf. on Computer Vision*, pages 85–100, 2016.

[28] Siqi Liu, Xiangyuan Lan, and Pong C Yuen. Remote photoplethysmography correspondence feature for 3D mask face presentation attack detection. In *European Conf. on Computer Vision*, pages 558–573, 2018.

[29] Javier Hernandez-Ortega, Julian Fierrez, Aythami Morales, and Pedro Tome. Time analysis of pulse-based face anti-spoofing in visible and nir. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, pages 544–552, 2018.

[30] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. FakeCatcher: Detection of synthetic portrait videos using biological signals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[31] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *California Law Review*, 104:671, 2016.

[32] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[33] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals and it's biased against blacks. *ProPublica*, 2016.

[34] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. *ProPublica*, 2016.

[35] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.

[36] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 259–268, 2015.

[37] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

[38] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[39] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2017.

[40] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2014.

[41] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshop*, 2017.

[42] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[43] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018.

[44] Shumeet Baluja and Ian Fischer. Learning to attack: Adversarial transformation networks. In *AAAI*, 2018.

[45] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *ACM International Conference on Multimedia Retrieval*, pages 269–277, 2017.

[46] Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(1):3–16, 2018.

[47] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: an end-to-end watermarking framework for ownership protection of deep neural networks. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 485–497, 2019.

[48] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Asia Conference on Computer and Communications Security*, pages 159–172, 2018.

[49] Jingjing Zhao, Qingyue Hu, Gaoyang Liu, Xiaoqiang Ma, Fei Chen, and Mohammad Mehedi Hassan. AFA: Adversarial fingerprinting authentication for deep neural networks. *Computer Communications*, 2019.

[50] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244, 2020.

[51] Hui Xu, Yangfan Zhou, Jiang Ming, and Michael Lyu. Layered obfuscation: a taxonomy of software obfuscation techniques for layered security. *Cybersecurity*, 3:1–18, 2020.

[52] Abhishek Chakraborty, Ankit Mondai, and Ankur Srivastava. Hardware-assisted intellectual property protection of deep learning models. In *ACM/IEEE Design Automation Conference*, pages 1–6, 2020.

[53] Mingliang Chen and Min Wu. Protect your deep neural networks from piracy. In *IEEE International Workshop on Information Forensics and Security*, 2018.

[54] S Cheung Sen-ching, Muhammad Usman Rafique, and Wai-Tian Tan. Privacy-preserving distributed deep learning with privacy transformations. In *IEEE International Workshop on Information Forensics and Security*, pages 1–7, 2018.

[55] Shiguo Lian. A block cipher based on chaotic neural networks. *Neurocomputing*, 72(4-6):1296–1301, 2009.

[56] Nooshin Bigdeli, Yousef Farid, and Karim Afshar. A novel image encryption/decryption scheme based on chaotic neural networks. *Engineering Applications of Artificial Intelligence*, 25(4):753–765, 2012.

[57] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Neural Information Processing Systems*, pages 2017–2025, 2015.

[58] Yann LeCun. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[59] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[60] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[61] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 389–398, 2018.

[62] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graphics*, 31(4), 2012.

[63] Wenjin Wang, Albertus C den Brinker, Sander Stuijk, and Gerard de Haan. Robust heart rate from fitness videos. *Physiological Measurement*, 38(6):1023, 2017.

[64] Qiang Zhu, Chau-Wai Wong, Chang-Hong Fu, and Min Wu. Fitness heart rate measurement using face videos. In *IEEE Conf. on Image Processing*, pages 2000–2004, 2017.

[65] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[66] Mauro Barni, Giulia Droandi, and Riccardo Lazzeretti. Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing. *IEEE Signal Process. Mag.*, 32(5):66–76, 2015.

[67] Liang Du, Meng Yi, Erik Blasch, and Haibin Ling. Garp-face: Balancing privacy protection and utility preservation in face de-identification. In *IEEE International Joint Conference on Biometrics*, pages 1–8, 2014.

[68] Marta Gomez-Barrero, Christian Rathgeb, Javier Galbally, Julian Fierrez, and Christoph Busch. Protected facial biometric templates based on local gabor patterns and adaptive bloom filters. In *International Conference on Pattern Recognition*, pages 4483–4488, 2014.

[69] Stelvio Cimato, Marco Gamassi, Vincenzo Piuri, Roberto Sassi, and Fabio Scotti. A multi-biometric verification system for the privacy protection of iris templates. In *International Workshop on Computational Intelligence in Security for Information Systems*, pages 227–234, 2009.

[70] Sheng Li and Alex C Kot. Fingerprint combination for privacy protection. *IEEE Trans. Inf. Forensics Security*, 8(2):350–360, 2012.

[71] Veeru Talreja, Matthew C Valenti, and Nasser M Nasrabadi. Deep hashing for secure multimodal biometrics. *IEEE Trans. Inf. Forensics Security*, 16:1306–1321, 2020.

[72] Alejandro Peña, Julian Fierrez, Aythami Morales, and Agata Lapedriza. Learning emotional-blinded face representations. In *International Conference on Pattern Recognition*, pages 3566–3573, 2021.

[73] Blaž Bortolato, Marija Ivanovska, Peter Rot, Janez Križaj, Philipp Terhörst, Naser Damer, Peter Peer, and Vitomir Štruc. Learning privacy-enhancing face representations through feature disentanglement. In *International Conference on Automatic Face and Gesture Recognition*, pages 45–52, 2020.

[74] Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ruben Tolosana. SensitiveNets: Learning agnostic representations with application to face images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(6):2158–2164, 2020.

[75] Philipp Terhörst, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Unsupervised privacy-enhancement of face representations using similarity-sensitive noise transformations. *Applied Intelligence*, 49(8):3043–3060, 2019.

[76] Weixuan Chen and Rosalind W Picard. Eliminating physiological information from facial videos. In *IEEE Conf. on Automatic Face and Gesture Recogition*, pages 48–55, 2017.

[77] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[78] J-C Terrillon, Mahdad N Shirazi, Hideo Fukamachi, and Shigeru Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *IEEE Conf. on Automatic Face and Gesture Recogition*, pages 54–61, 2000.

[79] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. L1 trend filtering. *SIAM review*, 51(2):339–360, 2009.

[80] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Internation Conf. on Learning Representation*, 2015.

[81] Ronny Stricker, Steffen Müller, and Horst-Michael Gross. Non-contact video-based pulse rate measurement on a mobile service robot. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 1056–1062, 2014.

[82] Si-Qi Liu and Pong C Yuen. A general remote photoplethysmography estimator with spatiotemporal convolutional network. In *International Conference on Automatic Face and Gesture Recognition*, pages 481–488, 2020.

[83] Sungjun Kwon, Jeehoon Kim, Dongseok Lee, and Kwangsuk Park. ROI analysis for remote photoplethysmography on facial video. In *Annual Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 4938–4941, 2015.

[84] Miroslav Goljan, Jessica Fridrich, and Rémi Cogranne. Rich model for steganalysis of color images. In *IEEE Workshop on Information Forensics and Security*, pages 185–190, 2014.

[85] Jan Kodovsky, Jessica Fridrich, and Vojtěch Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Security*, 7(2):432–444, 2011.

[86] Jishen Zeng, Shunquan Tan, Guangqing Liu, Bin Li, and Jiwu Huang. WISERNet: Wider separate-then-reunion network for steganalysis of color images. *IEEE Trans. Inf. Forensics Security*, 14(10):2735–2748, 2019.

[87] Ashwin Swaminathan, Min Wu, and KJ Ray Liu. Digital image forensics via intrinsic fingerprints. *IEEE Trans. Inf. Forensics Security*, 3(1):101–117, 2008.

[88] Matthew C Stamm, Min Wu, and KJ Ray Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[89] Ashwin Swaminathan, Min Wu, and KJ Ray Liu. Component forensics. *IEEE Signal Process. Mag.*, 26(2):38–48, 2009.

[90] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020.

[91] Umur Aybars Ciftci and Lijun Yin. Heart rate based face synthesis for pulse estimation. In *International Symposium on Visual Computing*, pages 540–551, 2019.

[92] Mohammad Soleymani, Jeroen Lichtenauer, Thierry Pun, and Maja Pantic. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*, 3(1):42–55, 2011.

[93] Zheng Zhang, Jeff M Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andy Horowitz, Huiyuan Yang, et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3446, 2016.

[94] Gee-Sern Hsu, ArulMurugan Ambikapathi, and Ming-Shiang Chen. Deep learning with time-frequency representation for pulse estimation from facial videos. In *IEEE International Joint Conference on Biometrics*, pages 383–389, 2017.

[95] Wenjin Wang, Albertus C Den Brinker, and Gerard De Haan. Single-element remote-ppg. *IEEE Transactions on Biomedical Engineering*, 66(7):2032–2043, 2018.

[96] Pavel Korshunov and Touradj Ebrahimi. Using face morphing to protect privacy. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 208–213, 2013.

[97] Philip J Benson. Morph transformation of the facial image. *Image and Vision Computing*, 12(10):691–696, 1994.

[98] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1-2):225–270, 1994.

[99] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[100] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[101] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[102] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[103] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3992–4001, 2017.

[104] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web*, pages 1171–1180, 2017.

[105] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

[106] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69, 2018.

[107] Cynthia S Crowson, Elizabeth J Atkinson, and Terry M Therneau. Assessing calibration of prognostic risk scores. *Statistical Methods in Medical Research*, 25(4):1692–1706, 2016.

[108] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017.

[109] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[110] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 797–806, 2017.

[111] Mingliang Chen, Aria Shahverdi, Sarah Anderson, Se Yong Park, Justin Zhang, Dana Dachman-Soled, Kristin Lauter, and Min Wu. Transparency tools for fairness in ai (luskin). In *Research in Mathematics and Public Policy*, pages 47–80. 2020.

[112] Mingliang Chen and Min Wu. Towards threshold invariant fair classification. In *Conference on Uncertainty in Artificial Intelligence*, pages 560–569, 2020.

[113] Toon Calders and Sicco Verwer. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

[114] Rafael C Gonzalez, Richard E Woods, and Steven L Eddins. *Digital Image Processing*. Pearson Education, 2001.

[115] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30, 2011.

[116] George Kimeldorf and Grace Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

[117] William Dieterich, Christina Mendoza, and Tim Brennan. *COMPAS risk scales: Demonstrating accuracy equity and predictive parity*. Northpointe Inc., 2016.

[118] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580, 2018.