

TOWARD SELF-SUPERVISED AND PRIVACY-PRESERVING  
REMOTE HEART RATE ESTIMATION FROM FACIAL  
VIDEOS

by

DIVIJ GUPTA

A thesis submitted to the  
Department of Electrical and Computer Engineering  
in conformity with the requirements for  
the degree of Master of Applied Science

Queen's University  
Kingston, Ontario, Canada

May 2023

Copyright © Divij Gupta, 2023

# Abstract

Remote heart rate (HR) estimation has become increasingly feasible through advances in deep learning in recent years. A popular approach for this purpose is remote photoplethysmography (rPPG) which aims to measure the volumetric changes in blood flow using computer vision techniques, which in turn can be used for remote HR estimation. While there are several challenges faced by modern deep learning solutions for rPPG estimation, in this thesis, we focus on addressing two major problems. First, is the reliance on large amounts of labeled data for effective training. Second, is the privacy concerns when performing remote HR estimation, which is caused due to the use of videos of face in this process. To reduce the reliance of video representation learning on labeled data as well as for improved performance, we introduce a solution based on self-supervised contrastive learning for remote HR monitoring, which makes use of various augmentations of the original input videos to learn robust spatiotemporal video representations. We propose the use of 3 spatial and 3 temporal augmentations for training an encoder through our contrastive framework, followed by fine-tuning of the encoder for rPPG and HR estimation. Our experiments on two publicly available datasets, COHFACE and PURE showcase the improvement of our proposed approach over several related works as well as supervised learning baselines, as our results approach the state-of-the-art. We also perform thorough

experiments to showcase the effects of using different design choices such as the video representation learning method, the augmentations used in the pre-training stage, and others. We also demonstrate the robustness of our proposed method over the supervised learning approaches on reduced amounts of labeled data. To address the second problem (privacy), we propose a data perturbation method that involves extraction of certain areas of the face with less identity-related information, followed by pixel shuffling, and blurring. Our experiments on two rPPG datasets (PURE and UBFC) show that our approach reduces the accuracy of facial recognition algorithms by over 60%, with minimal impact on rPPG extraction. We also test our method on three facial recognition datasets (LFW, CALFW, and AgeDB), where our approach reduced performance by nearly 50%. Our findings demonstrate the potential of our approach as an effective privacy-preserving solution for rPPG estimation.

## Acknowledgments

I would like to express my most sincere gratitude to everyone who has supported me along this eventful journey. First and foremost, I would like to thank my supervisor, Prof. Ali Etemad, for his continued guidance, encouragement, patience, and support. I would also like to thank my labmates and friends for their help and support on numerous occasions. Lastly, I would like to thank my family for their continued faith in me, without which the journey through graduate school would have been extremely strenuous.

## Statement of Originality

The following work described is my own and I hereby certify the intellectual content of this thesis is the product of my own work. To the best of my knowledge, all references and contributions of other individuals have been acknowledged, cited, and sourced appropriately.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>Statement of Originality</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>x</b>
<b>Glossary of Abbreviations</b>	<b>xii</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 Smart City Contextualization . . . . .	3
1.1.2 IoT Architecture . . . . .	5
1.2 Problem and Motivation . . . . .	7
1.3 Solutions Overview . . . . .	9
1.4 Contributions . . . . .	10
1.5 Publications . . . . .	11
1.6 Organization of Thesis . . . . .	12
<b>Chapter 2: Related Work</b>	<b>13</b>
2.1 rPPG Estimation . . . . .	13
2.1.1 Classical Methods . . . . .	13
2.1.2 Deep Learning Methods . . . . .	15
2.1.3 Hybrid Methods . . . . .	16
2.1.4 Remote HR estimation . . . . .	16
2.2 Self-supervised Learning . . . . .	17
2.2.1 Pre-text Tasks for Images . . . . .	17

2.2.2	Pre-text Tasks for Videos . . . . .	18
2.2.3	Contrastive Learning . . . . .	20
2.3	Facial Recognition . . . . .	21
2.3.1	Classical Methods . . . . .	21
2.3.2	Deep Learning Methods . . . . .	21
2.4	Privacy Preservation . . . . .	22
2.4.1	Privacy Attacks . . . . .	22
2.4.2	General Privacy-preserving Methods . . . . .	22
2.4.3	Privacy-preserving Methods for Faces . . . . .	23
<b>Chapter 3: Self-supervised Learning</b>		<b>24</b>
3.1	Method . . . . .	24
3.1.1	Solution Architecture . . . . .	24
3.1.2	RoI Detection . . . . .	25
3.1.3	Stage 1: Self-supervised Pre-training . . . . .	26
3.1.4	Stage 2: Supervised Fine-tuning . . . . .	29
3.1.5	Loss Functions . . . . .	29
3.1.6	HR Calculation . . . . .	31
3.2	Experiment Setup . . . . .	31
3.2.1	Datasets . . . . .	31
3.2.2	Evaluation Scheme and Metrics . . . . .	33
3.2.3	Comparisons . . . . .	33
3.2.4	Implementation . . . . .	38
3.3	Results and Discussions . . . . .	39
3.3.1	Performance and Comparison . . . . .	40
3.3.2	Impact of 3D Convolutions . . . . .	43
3.3.3	Impact of Negative Pairs in Pre-training . . . . .	46
3.3.4	Impact of Different Facial Regions . . . . .	49
3.3.5	Impact of Different Augmentations . . . . .	50
3.3.6	Performance on Reduced Labels . . . . .	52
<b>Chapter 4: Privacy Preservation</b>		<b>53</b>
4.1	Method . . . . .	53
4.1.1	Problem Setup . . . . .	53
4.1.2	Perturbation Method . . . . .	53
4.1.3	rPPG Estimation Backbone . . . . .	55
4.1.4	Loss Function . . . . .	55
4.2	Experiment Setup . . . . .	56
4.2.1	Datasets . . . . .	56
4.2.2	Evaluation Scheme and Metrics . . . . .	57

4.2.3	Training . . . . .	58
4.3	Results and Discussions . . . . .	59
4.3.1	Results on Pixel-wise Shuffling . . . . .	59
4.3.2	Impact of Patch-wise Shuffling . . . . .	63
4.3.3	Comparison with Other Methods . . . . .	65
4.3.4	Reconstruction Attempt . . . . .	66
4.3.5	Impact on Public Benchmarks . . . . .	67
<b>Chapter 5:</b>	<b>Conclusion and Future Work</b>	<b>68</b>
5.1	Conclusion . . . . .	68
5.2	Future Work . . . . .	70
<b>Bibliography</b>		<b>71</b>

# List of Tables

3.1	Architectural details of the encoder used in our proposed method. . .	28
3.2	Architectural details of the (2+1)D encoder (Encoder B) used in our experiments. . . . .	35
3.3	Comparison of our proposed method with prior works on COHFACE.	38
3.4	Comparison of our proposed method with prior works on PURE. . . .	39
3.5	Comparison of our proposed method with prior works on PURE (MPEG-4). . . . .	40
3.6	Impact of different encoders for pre-training (full RoI) for COHFACE.	44
3.7	Impact of different encoders for pre-training (full RoI) for PURE (MPEG-4). . . . .	44
3.8	Impact of different encoders for pre-training (cheek as RoI) for COHFACE. . . . .	45
3.9	Impact of different encoders for pre-training (cheek as RoI) for PURE (MPEG-4). . . . .	45
3.10	Impact of different encoders for pre-training (forehead as RoI) for COHFACE. . . . .	46
3.11	Impact of different encoders for pre-training (forehead as RoI) for PURE (MPEG-4). . . . .	46

3.12	Impact of including and excluding negative pairs in pre-training (full RoI) for COHFACE. . . . .	47
3.13	Impact of including and excluding negative pairs in pre-training (full RoI) for PURE (MPEG-4). . . . .	47
3.14	Impact of including and excluding negative pairs in pre-training (cheek as RoI) for COHFACE. . . . .	48
3.15	Impact of including and excluding negative pairs in pre-training (cheek as RoI) for PURE (MPEG-4). . . . .	48
3.16	Impact of including and excluding negative pairs in pre-training (forehead as RoI) for COHFACE. . . . .	50
3.17	Impact of including and excluding negative pairs in pre-training (forehead as RoI) for PURE (MPEG-4). . . . .	50
4.1	Overview of the facial recognition datasets. . . . .	57
4.2	Comparison of various parameters for our proposed method on PURE. . . . .	60
4.3	Comparison of various parameters for our proposed method on UBFC. . . . .	61
4.4	Comparison with other privacy-preserving methods on rPPG estimation. . . . .	65
4.5	Results of using our method on facial recognition datasets in terms of verification accuracy %. . . . .	67

# List of Figures

1.1	An overview of PPG collection using an oximeter, as well as rPPG estimation using a standard video camera. . . . .	2
1.2	Depiction of a general IoT layout where an rPPG estimation system can be integrated into. . . . .	4
1.3	A typical 3-layered IoT architecture in the context of rPPG estimation.	5
3.1	The overall layout of the proposed two-stage approach. . . . .	25
3.2	Example of a sample clip after being processed by different augmentations.	27
3.3	Sample frames showing the varying conditions in PURE dataset. . . .	32
3.4	Illustration of the 3D and (2+1)D convolutions. T stands for the temporal filter size, while S stands for the spatial filter size. . . . .	36
3.5	An overview of the self-supervised learning strategies used in this study.	37
3.6	Visualization of predicted rPPG for different conditions presented in the datasets. . . . .	41
3.7	Correlation plots for COHFACE (left), and PURE (MPEG-4) (right).	42
3.8	Bland-Altman plots for COHFACE (left), and PURE (MPEG-4) (right).	43
3.9	Performance of self-supervised and fully supervised approaches on reduced amounts of labelled data for COHFACE (left), and PURE (MPEG-4) (right). . . . .	51

4.1	An overview of our proposed privacy-preserving data perturbation pipeline. . . . .	54
4.2	Sample pairs of images belonging to the same subject from CALFW dataset showing varying imaging conditions. . . . .	58
4.3	Correlation plots for UBFC (left), and PURE (right). . . . .	62
4.4	Bland-Altman plots for UBFC (left), and PURE (right). . . . .	62
4.5	Visualization of predicted rPPG for PURE (top), and UBFC (bottom). . . . .	63
4.6	Visualization of ArcFace embeddings reduced to 2 dimensions with PCA for a) Face and b) RoI+Sh+B for 5 subjects in PURE. . . . .	63
4.7	The impact of different kernel sizes used for blurring, on ID for both PURE and UBFC datasets. . . . .	64
4.8	Visualization of pixel and patch-wise perturbed images. . . . .	64
4.9	Reconstruction attempts. a) RoI, b) RoI+Sh+B, c) Reconstructed RoI from UNet, and d) Reconstructed RoI from Pix2Pix. . . . .	66

## Glossary of Abbreviations

**(2+1)D** (2+1)-Dimensional. 34, 38–40, 43

***R*** Correlation. 33, 38–40, 44–50, 57, 59

**1D** 1-Dimensional. 34

**2D** 2-Dimensional. 15, 17, 34

**3D** 3-Dimensional. 9, 15, 16, 28, 34, 38–40, 43, 52, 55

**BDCT** Block Discrete Cosine Transform. 23, 65

**bpm** beats per minute. 33, 57

**CAN** Convolutional Attention Network. 16, 38–40

**CDCConv** Central Difference Convolution. 16, 38, 39

**CIoT** Cognitive IoT. 4, 5

**CNN** Convolutional Neural Network. 15–17, 21, 28, 38, 39, 55

**ConvLSTM** convolutional LSTM. 15, 38, 39

**DP** Differential Privacy. 23

**fps** frames per second. 31, 32, 56

**GAN** General Adversarial Network. 16, 23, 66, 70

**HR** heart rate. i, 1, 6, 10, 15–17, 31, 33, 42, 43, 48, 57, 60, 68, 70

**ID** identification accuracy. 58, 60–62, 64

**IoT** Internet of Things. 2–5, 7

**LSTM** Long Short-Term Memory. 15

**MAE** Mean Absolute Error. 33, 38–40, 42, 44–50, 57, 59–61, 65

**PCA** Principal Component Analysis. 21, 58

**PFE** Physiological signal Feature Extraction. 16, 38, 39

**PPG** Photoplethysmography. 1, 2, 8, 25, 29–33, 42, 48, 52–54, 56, 60

**RGB** red, green and blue. 13, 14, 17

**RMSE** Root Mean Square Error. 33, 38–40, 44–50, 57, 59–61, 65

**RoI** Region of Interest. 13, 24–27, 29, 33, 39, 43, 45–47, 49, 50, 54, 55, 58–61, 63, 64,  
66–68

**rPPG** remote photoplethysmography. i, ii, vi, ix, x, 1–16, 23, 25, 30, 31, 34, 37,  
41–43, 47, 49, 53–66, 68–70

**TFA** Temporal Face Alignment. 16, 38, 39

# Chapter 1

## Introduction

### 1.1 Background

Photoplethysmography (PPG) is an optical measurement that indicates the changes in blood volume. It is a relatively cheap and non-invasive method which uses a light source and detector to measure the change in light variation caused by blood flow through the flesh [1]. A variety of different types of information is carried by or can be derived from PPG signals [2, 3], including hemoglobin levels, cardiovascular conditions, heart rate (HR), cardiac output, blood pressure, oxygen saturation level, and even a subject's respiration rate. The signals have been used in a variety of non-medical applications as well, for instance in emotion recognition [4], cognitive load assessment [5], and others.

While PPG is conventionally measured through an oximeter worn by the user on a finger, studies have shown that blood flow, and consequently PPG, can also be measured from afar [6], as blood flow causes subtle color variations at the surface of the skin. This process, termed remote photoplethysmography (rPPG) eliminates all forms of contact while giving the same benefits as a PPG signal acquired through an

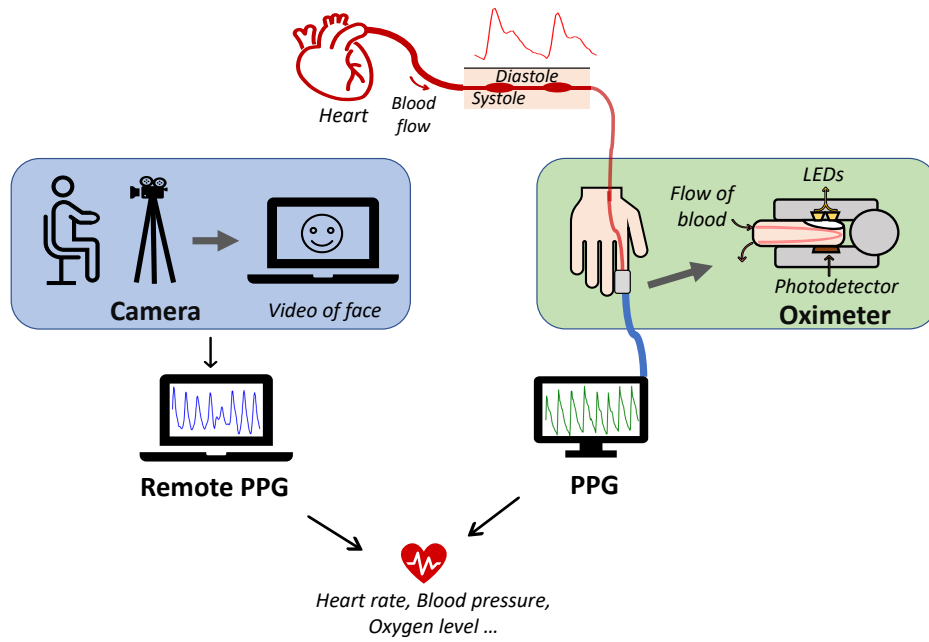


Figure 1.1: An overview of PPG collection using an oximeter, as well as rPPG estimation using a standard video camera.

oximeter. Hence, this is very useful in scenarios such as pandemics or virtual settings where direct access to the skin is not advised or always possible. Furthermore, since rPPG requires only a camera, it is very easy to integrate into existing Internet of Things (IoT)-enabled smart environments which comprise cameras, data transmission channels, and cloud servers for storing and processing information [7]. The various vitals and information that can be extracted from a PPG signal and the *non-contact* remote acquisition of rPPG provide a strong motivation for rPPG to be incorporated into smart homes, workplaces, hospitals and others [8, 9]. An overview of PPG and rPPG estimation is depicted in Figure 1.1.

### 1.1.1 Smart City Contextualization

A smart city constitutes a variety of smart environments such as smart hospitals, smart homes, smart workplaces, smart vehicles and others. While each of these process different information for different purposes, on a rudimentary level, they comprise similar components namely data acquisition devices, communication channels, and cloud servers. Moreover, these environments generally work towards the common goal of utilizing advanced technologies to add value and convenience to the lives of the citizens of a smart city and enhance their quality of life [10].

An rPPG estimation system falls primarily in line with remote health monitoring which has become an integral part of various smart environments. With the adoption of the remote health monitoring paradigm, medical professionals can monitor the vitals and other physical symptoms of a patient without having to be in the same place as the patient and provide them with adequate consultation. In smart environments such as smart homes and smart workplaces, remote monitoring of vitals helps to monitor health without having to leave the premises or commute to medical institutions [11, 12]. This benefits especially those who lack mobility such as the senior and physically challenged people [13, 14]. In smart vehicles too, the vitals of the user can be tracked and appropriate feedback could be provided on the go [15]. These varied use cases discussed, further the impact and importance of an rPPG estimation system.

In Figure 1.2, we illustrate the integration of an rPPG estimation system into an existing IoT layout. The layout comprises of the user in various surroundings with access to a camera, local devices to run inference algorithms, medical professionals to provide consultations, and lastly a centralized cloud server for continuous training and maintenance of data and/or algorithms. The user can run a remote vitals checkup in

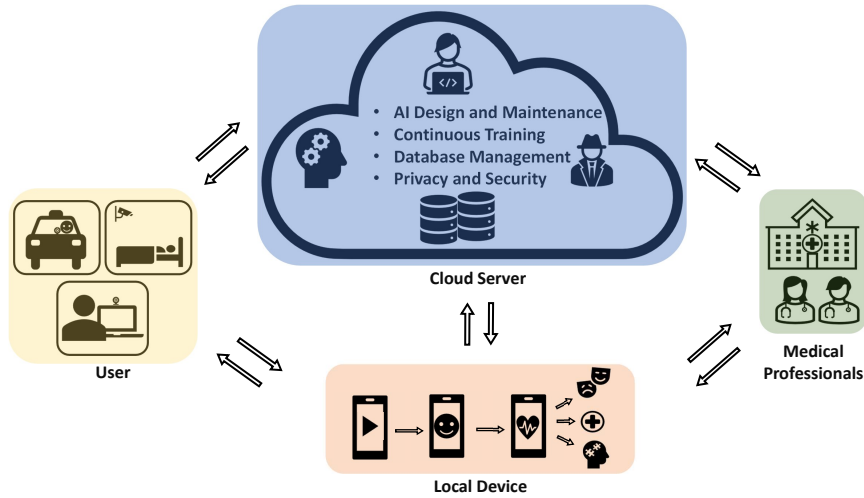


Figure 1.2: Depiction of a general IoT layout where an rPPG estimation system can be integrated into.

the layout as long as they have access to a camera for capturing the video of their face. Next, the rPPG estimation system would be run on any available local device for inferring the rPPG signal from the face video. After estimating the rPPG, further insights such as vitals, emotions (for mood and mental health management), and others can be derived from it and be made available to the user and appropriate medical consultation can be provided if needed. All these processes would take place in conjunction with the cloud server. For capturing the face video, a good camera which is common in smart environments will suffice. For computing on local devices, deep learning algorithms have been known to be deployed on a variety of devices besides computers, e.g., mobile devices and micro-controllers [16, 17], which makes the integration of rPPG estimation into any existing IoT system seamless, allowing for it to be used across multiple environments.

Lastly, since rPPG estimation involves training of intelligent algorithms, the contextualization can also be broadened to base upon the Cognitive IoT (CIoT)

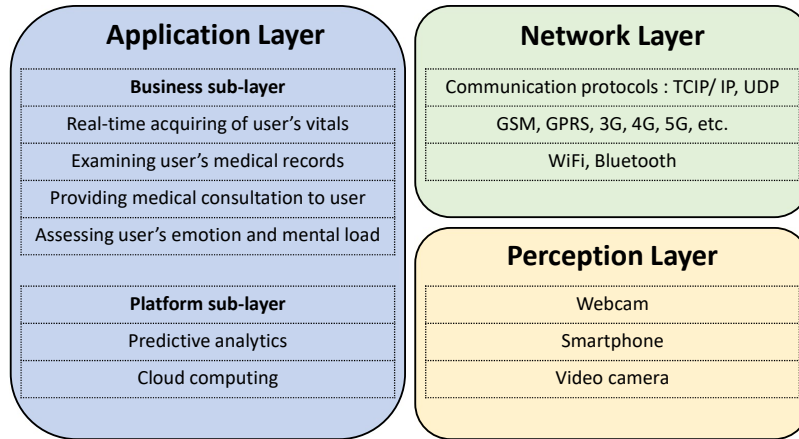


Figure 1.3: A typical 3-layered IoT architecture in the context of rPPG estimation.

[18] paradigm. The CIoT paradigm is an upgrade over the IoT paradigm as it incorporates intelligence into the existing IoT components, thereby adding another layer of ‘smartness’ in the smart environment. Despite that, for simplicity as done in [19], we too adopt the general IoT paradigm for the contextualization of rPPG estimation in the next section.

### 1.1.2 IoT Architecture

IoT architectures are often modelled using three layers of abstraction, namely perception layer, network layer, and application layer [20]. In the following, we describe rPPG estimation in the context of an IoT architecture for smart environments with consideration of these three layers. An overview of the architecture is shown in Figure 1.3.

**Perception Layer.** This layer serves as the information source of an IoT system and involves data acquisition devices. For rPPG estimation, this layer comprises of the various cameras present in the smart environments. Examples include cameras

in smartphones, webcams on computers, driver/passenger-facing cameras in vehicles, video cameras in smart homes, and others. Any of these can be used to record video of the face to be used subsequently for inference.

**Network Layer.** This layer is responsible for conveying the information from the perception layer to the application layer. This layer is essentially based on the existing Internet and mobile telecommunication infrastructure. Some examples of the components include General Packet Radio Service, Fourth/Fifth-Generation communication, WiFi, and others which provide wireless and long-distance communication. For the rPPG estimation system, the Internet can be accessed through smart computing devices such as smartphones, computers, smart camera systems, and others. Also, Bluetooth can be used by the smart devices for short-distance communication.

**Application Layer.** This layer processes the information received from the perception layer into useful applications. It can be further divided into platform and business sub-layers. The platform sub-layer comprises various algorithms and protocols designed, run, and updated on the cloud to ensure the smooth functioning of the entire architecture. Mainly four functions are carried out at the platform sub-layer:

- (1) Algorithm Design and Maintenance,
- (2) Continuous Training,
- (3) Database Management, and
- (4) Privacy and Security.

The rPPG estimation system would be a part of this sub-layer and be used for inference of HR remotely and without the need for physical sensors to come in contact with the user, and also undergo subsequent training on new data. Other models used to derive vitals and other information from the rPPG signals too would be a

part of this layer. Besides the engineering team, medical professionals would also contribute to this sub-layer by providing field expertise to better guide the designing of the algorithms. The business sub-layer uses the final extracted information to meet the goals and requirements of the stakeholders. The user as well as the medical professionals could be notified of the user's vitals if/when required and also be alerted appropriately in the case of any anomalies to ensure prompt diagnosis of any symptoms. Besides remote health monitoring, other applications such as stress assessment [21] at workplaces or emotion recognition [22] while driving can be conceptualized. Since this layer is very crucial to the working of the entire architecture, it would be paid extra attention to secure it safely and to ensure all-time connectivity.

A key aspect of our overall envisioned architecture for rPPG estimation in smart environments is the notion of ubiquity, while also maintaining and protecting user privacy. As a result, as we show in the following subsections, we design our deep learning solutions with this notion in mind, where we utilize specific regions of the face with less identifiable features for rPPG estimation. This approach would allow for the user's full facial image to not be entered into the pipeline, therefore, allowing for better privacy preservation. Additionally, in the IoT design, special attention should be given to utilizing privacy-preserving approaches such as federated learning [23], and secure software and cloud practices [24] to ensure user security and privacy.

## 1.2 Problem and Motivation

In recent years, rPPG estimation, has become more robust as a result of advances in computer vision and deep learning [25, 26]. However, a major limitation of supervised deep learning solutions is the reliance on huge amounts of *annotated* data for proper

training. To address this, self-supervised learning has lately begun to gain momentum in the field of deep learning. The central concept behind this paradigm is to generate pseudo-labels instead of human-annotated labels, which would then be used to train the model. These pseudo-labels are often derived by performing various augmentations (transformations) on the available data. The model is then trained to recognize these augmentations, for instance, by detecting that two different transformations applied to the same input are indeed renditions of the *same* information. This will allow the network to learn informative representations from the input data without requiring the actual output labels. Following the self-supervised learning step, fine-tuning is often applied to train specific layers of the network for the downstream task.

Concerning the data, there is yet another problem faced in rPPG estimation. Compared to other anatomical regions such as fingers, and wrist, where standard PPG is often collected from, rPPG signals are stronger on the *face* [27] given the volume of blood often flowing through it. Moreover, the face is the most readily accessible region to capture using a camera, hence is the pre-dominant body part for collecting rPPG data [28, 29, 30]. However, the use and distribution of such datasets and algorithms trained on them is a deep concern in terms of *privacy*. The use of facial data for rPPG estimation makes such applications highly sensitive as the face is one of the most crucial modes of biometrics which can be used to identify/authenticate and track individuals [31]. Recent research has also revealed that the security of deployed intelligent systems can be jeopardized which can lead to the leakage/reconstruction of sensitive information [32, 33]. While some privacy-preserving methods have been proposed to address this concern in general, the alterations used in the methods may impede accurate rPPG estimation, the effect of which remains largely unexplored.

### 1.3 Solutions Overview

In this work, to provide an effective approach for rPPG estimation while reducing reliance on output labels, we propose a deep learning solution that leverages contrastive self-supervised pre-training. Our model uses a 3-Dimensional (3D) convolution-based encoder to obtain representations of facial videos through self-supervised contrastive learning. The contrastive learning setup in our method works by generating embeddings of a video clip and its augmented counterpart, followed by maximizing the similarity between them while minimizing the similarity between embeddings of different clips. This helps the network learn effective spatiotemporal representations without the use of any labeled data. After pre-training, the encoder is fine-tuned for the downstream task of rPPG estimation. Our experiments on two public datasets, COHFACE [30] and PURE [28], demonstrate that our method outperforms several supervised learning algorithms and also provides robust results when trained on reduced amounts of labeled data.

Next, to enable rPPG estimation from facial videos without risking or sacrificing privacy, we propose a simple yet highly effective pipeline for obtaining a privacy-preserving face representation to extract rPPG from. Our method first performs the extraction of pre-selected facial regions with the goal of excluding key identifying features, followed by shuffling of the pixels and blurring the outcome to obtain a privacy-preserving face representation. Through this, we destroy spatial consistency of facial regions at pixel-level while maintaining the overall color intensity of the pixels which is crucial for the extraction of rPPG signals as demonstrated in [34, 35]. We perform various experiments on two publicly available datasets, PURE [28] and UBFC [29], and demonstrate that rPPG can be accurately measured using our face

representations, while the detection of identities becomes excessively challenging. Additional experiments on three public facial recognition datasets LFW [36], CALFW [37], and AgeDB [38], show that the new face representations generated from our method significantly deteriorate the performance of a widely used facial recognition system.

#### 1.4 Contributions

Our contributions in this thesis can be summarized as follows:

- We propose a two-stage approach based on self-supervised contrastive pre-training and fine-tuning as an effective solution to reduce the reliance of rPPG estimation on *labeled* data. We perform thorough experiments on two publicly available datasets and validate the effectiveness of our method, showing that our solution achieves strong results in measuring rPPG and estimating HR without the need for contact-based sensors.
- We perform a large number of experiments to evaluate the impact of different design choices such as the pairing strategy and the augmentations used in the self-supervised training paradigm, the video representation learning technique, and the facial regions taken for extracting the rPPG. Further experiments demonstrate that our solution performs robustly when the amount of *labeled* data for training is reduced.
- We propose a new face representation that conceals user identity and enhances privacy in the case of any leakage or reconstruction of data through malicious attacks. Our approach includes the selection of facial regions followed by the

shuffling of pixels and blurring. Our experiments on two publicly available rPPG datasets show that the proposed technique allows for rPPG to be effectively measured with minimal degradation while the recognition rate of identity is significantly reduced.

- Our comparison to other privacy-preserving representations for the face demonstrates that those techniques do not facilitate accurate estimation of rPPG. Furthermore, our thorough experiments validate the different design choices associated with the proposed method such as the order of shuffling, grouping of pixels, and others. Finally, we demonstrate that our proposed privacy-preserving scheme causes a huge decline in the performance of existing facial recognition systems when tested on three public datasets.

## 1.5 Publications

The following papers have resulted from this research :

- [\[39\]](#): **Divij Gupta**, Ali Etemad, “Self-supervised Remote Monitoring of Heart Rate from Videos”, *AAAI Workshop on Human-Centric Self-Supervised Learning*, 2022.
- **Divij Gupta**, Ali Etemad, “Remote Heart Rate Monitoring in Smart Environments from Videos with Self-supervised Pre-training”, *Under Review*, 2022.
- **Divij Gupta**, Ali Etemad, “Privacy-Preserving Remote Heart Rate Estimation from Facial Videos”, *Under Review*, 2023.

## 1.6 Organization of Thesis

The rest of this thesis is organized as follows:

**Chapter 2** presents a comprehensive overview of prior works on rPPG estimation. We review several methods utilizing classical and deep learning methods, and also methods that combine these two types of approaches. Next, we review self-supervised learning as it is a key component of the work presented in this thesis. This is followed by a brief review of classical and deep learning methods for facial recognition. Lastly, we briefly review existing privacy concerns in deep learning methods, followed by different methods to mitigate these issues, especially concerning data involving faces.

**Chapter 3** presents our method using self-supervised contrastive learning as a pre-training step for rPPG estimation. In this chapter, we describe the methodology including the data pre-processing, the architectural details, and different components of our two-stage approach, along with the losses and hyperparameters used during training. We then present the experiment setup and results, including comparisons to other methods, ablations, and sensitivity studies.

**Chapter 4** presents our method for preserving the privacy of subjects in rPPG estimation. In this chapter, we describe the methodology including the privacy-preserving data perturbation mechanism, the architectural details as well as the loss and hyperparameters used during training. We then present our experiments and results to evaluate impact on both rPPG estimation and face de-identification.

**Chapter 5** concludes the thesis by providing a summary of our work followed by a discussion of potential future research work.

## Chapter 2

### Related Work

In this section, we review prior works on rPPG estimation using both classical and deep learning approaches. We follow this with a review of literature on self-supervised representation learning. Finally, prior works on face recognition and privacy preservation are reviewed.

#### 2.1 rPPG Estimation

##### 2.1.1 Classical Methods

A number of classical image processing methods have used color space transformations and signal processing approaches to estimate rPPG. [27] proposed one of the first rPPG methods wherein the average color intensities of manually selected Region of Interest (RoI)s were computed for each frame. These channel mean intensities were then tracked temporally across the frames to obtain three traces, one for each of the red, green and blue (RGB) color channels. These traces were then band-pass filtered to remove noise, and it was concluded that the green channel contained the strongest rPPG signal. Following, in [40], only the green channel trace was computed

for the lower part of the face and then processed by a variety of filters to obtain the rPPG. In [41], however, all the three RGB traces for the entire face, were used and decomposed into three independent signals using Independent Component Analysis. Next, the signal with the highest peak in the power spectrum was selected for further processing to obtain the rPPG. In CHROM [34], the RGB traces were computed for all the facial skin pixels, projected onto a proposed chrominance subspace, band-pass filtered and combined linearly to obtain rPPG. A similar approach was taken in POS [35] where the RGB traces were projected onto an orthogonal color space to estimate rPPG signal. In 2SR [42], a slightly different approach was used wherein the skin pixels were detected and a subspace of the skin pixels was created for each frame. Next, the temporal rotation across the subspaces was tracked to estimate the rPPG signals. A key difference in 2SR with respect to other works was the use of the spatial distribution of the skin pixels which was discarded in other classical methods since they used the average intensity values of the skin pixels in a frame.

An interesting approach was taken in [43] where Eulerian video magnification was proposed. The authors used spatial decomposition, temporal filtering, and spatial reconstruction to amplify both the color as well as low-amplitude motion in the video. Since rPPG estimation primarily relies on the color variations on the skin surface, using color magnification made the variations more pronounced which could be used to obtain rPPG. In another approach in the same work, instead of the color variations caused by the blood flow, the expansion of the blood vessels was magnified. This provided another pathway of estimating rPPG signals from the skin surface.

### 2.1.2 Deep Learning Methods

More recently, deep learning has been used for rPPG estimation from facial videos. In HR-CNN [25], a two-stage Convolutional Neural Network (CNN) architecture was proposed, comprising of vanilla convolutions wherein the rPPG signals were estimated from the face videos and then used to predict HR. In PhysNet [44], different spatiotemporal models based on CNN and Long Short-Term Memory (LSTM) were explored for rPPG estimation. In DeeprPPG [26], a lightweight CNN architecture was used along with a novel rPPG aggregating strategy to adaptively combine rPPG signals from different skin regions. In [45], 2-Dimensional (2D) and 3D convolutions were used for the backbone architecture, followed by spatiotemporal strip pooling in the last layers to add attention to the feature maps.

In ETA-rPPGNet [46], a network was proposed in which a time-domain sub-network was used to reduce the redundant video information by extracting the crucial spatial features followed by a time-domain attention network to effectively predict rPPG and HR from the sub-network features. In [47], a multi-hierarchical spatiotemporal CNN was proposed. In [48], a two-stream architecture was proposed wherein two video inputs, the cropped face video (trunk branch) and the mask of the skin pixels (mask branch) were used. The trunk branch comprised of a combination of CNNs and convolutional LSTM (ConvLSTM)s [49] while the mask branch only had CNNs with intermediate fusion to the trunk branch through attention mechanism for improved processing of the skin pixels.

In [50], another two-stream network was proposed where the current frame (appearance) and its normalized difference with the next frame (motion) were processed in two different CNN pathways with intermediate fusions to provide attention to the

motion stream based on the appearance. This network is commonly referred to as the Convolutional Attention Network (CAN). In [51], a similar approach to [50] was proposed, but in turn replaced the standard convolutions with Central Difference Convolution (CDConv)s [52], allowing for improved processing of the spatial and temporal information in the feature maps. In [53], CAN was modified to introduce the Temporal Shift Module [54] for improved temporal modelling of the feature maps for rPPG estimation. In [55], the authors used the Convolutional Block Attention Module [56] to provide spatiotemporal attention in a 3D CDConv-based CNN architecture. In [57], the authors proposed two blocks namely the Physiological signal Feature Extraction (PFE) block and the Temporal Face Alignment (TFA) to tackle problems in rPPG estimation pertaining to changing face-camera distance and face motion.

### 2.1.3 Hybrid Methods

A number of prior works have combined classical image processing techniques with CNNs. In [58], the video frames were first pre-processed separately using orthogonal color space projection [35] and motion normalization [50], and then concatenated for processing by a CNN with different attention modules to provide spatiotemporal attention for rPPG estimation. In [59], the authors first used [34] to extract the rPPG signals and then refined them using a conditional General Adversarial Network (GAN) [60, 61].

### 2.1.4 Remote HR estimation

Another set of works focus directly on estimating the HR from facial videos. In [62], phase-based video motion processing [63] was used to magnify subtle color changes

and reduce the motion artifacts, followed by a CNN for remote HR estimation. In [64], the facial frames were transformed to YUV color space, divided into patches, and averaged to compute YUV traces as done previously for computing RGB traces. The traces obtained for the patches were then concatenated to form a spatiotemporal map. The spatiotemporal maps for adjacent video clips were then processed by a 2D CNN, the output representations of which were further processed by a Gated Recurrent Unit [65] to utilize the relation between adjacent clips and finally predict the HR.

Research has also been performed to leverage other modalities for rPPG estimation. For instance, some works have worked with using near-infrared sensors to record facial videos and estimate rPPG from them [66, 67]. In another work [68] depth information was used to fit a 3D face model and track it over the frames to negate the effect of large motions and then estimate rPPG from the RGB data.

## 2.2 Self-supervised Learning

Self-supervised learning aims to reduce the reliance of supervised learning approaches on human-annotated labels while also learning meaningful representations for enhanced performance. This training paradigm generally relies on generating pseudo-labels for pre-training neural networks prior to fine-tuning them for downstream tasks. A major differentiating factor among the self-supervised approaches lies in the design of the pretext learning step.

### 2.2.1 Pre-text Tasks for Images

In [69, 70], pseudo-classes were created by distorting images through combinations of rotation, translation, color shifts, and scaling. Each original image contributed towards

the generation of one pseudo-class. Thereafter, the network was trained to distinguish among these classes to make the network learn effective semantic representations robust to distortions. In [71], a patch was randomly sampled from an image, after which another patch was sampled from its neighborhood grid with the first patch in the center. Next, the network was trained to predict the position of the second patch with respect to the first one, thus learning key contextual information. In [72], original input images were divided into several patches as puzzle pieces, and the pretext task of the network was to solve the puzzle. As a result, key visual representations and spatial consistency were learned, resulting in improved performance on the downstream task. In [73], the images were rotated by certain angles, and the pretext task of the network was to successfully predict these rotation angles. In [74], certain regions of the image were cropped and the network was trained to fill in the regions as the pretext task. This helped the network better learn contextual information in images and perform better in subsequent downstream tasks. In [75], image colorization was explored as a pre-text task where the network was trained to colorize the grayscale version of the original image.

### 2.2.2 Pre-text Tasks for Videos

Other prior works have also explored different pre-text tasks specific to videos. Since a video has an added temporal dimension compared to images, many works focus on utilizing temporal ordering to design pre-text tasks to allow for the network to learn rich spatiotemporal representations for downstream tasks. In [76], frames from a high-motion video were sampled, and each set of frames was used to generate three tuples. One tuple comprised ordered frames (positive), while the other two tuples

comprised frames where the order was changed (negative). Next, the network was trained to classify whether the tuple had ordered or unordered frames. In [77], each video was sampled into a number of clips, after which the frames of one of the clips were shuffled. Following this step, the network was trained to find the location of the video clip with shuffled inputs among other non-shuffled clips in an odd-one-out manner. In [78], a number of clips were sampled from a video and shuffled among themselves. Subsequently, the network was trained to learn to predict the order of the shuffling. In [79], the order of frames of the videos was reversed, after which, the network was trained to classify from the optical flow of the input video whether the video was in reverse or not. Other tasks such as colorization [80], jigsaw-puzzle solving [81], and rotation prediction [82] have also been explored as a pre-text task for learning representations from videos.

Similar to the use of [83] for self-supervision in natural language processing, Masked Autoencoder [84] was recently explored for self-supervised computer vision tasks. In [84], random patches of the original image were masked and the autoencoder was trained to reconstruct the original image from the input patches. While the pre-text task is similar to [74], the Masked Autoencoder was based on Vision Transformers [85] instead of vanilla convolutions, allowing for the use of mask tokens [83] and positional embeddings [86]. This helped the model learn holistic representations by encompassing the rich semantic information and be used in the downstream tasks. Furthermore, the self-supervision strategy has been explored for videos as well [87]. The paradigm of self-supervised learning has been applied to a wide variety of problems such as image classification [72], wearable-based activity recognition [88], signal-based emotion recognition [89, 90], and more.

### 2.2.3 Contrastive Learning

Contrastive learning is another type of self-supervised learning which has gained momentum in the past few years and has shown tremendous improvement across various computer vision tasks such as object detection [91], facial expression recognition [92, 93], gaze estimation [94], signal analysis [95] and others. SimCLR [96] proposed the use of different augmentations to create pseudo-samples of the original data and train the network to learn features to maximize the similarity between the augmented counterparts of the same original sample and also to minimize the similarity between the augmented counterparts of two different original samples. A number of other contrastive learning approaches such as MoCo [97], NNCLR [98], have also been proposed to take advantage of different aspects of the data for learning effective representations.

Another approach toward self-supervised learning bears considerable similarities with the likes of SimCLR [96], but only focuses on maximizing the similarity between the augmented counterparts of the same original image, and omits the similarity minimization between the two samples of the different original samples altogether. This paradigm of self-supervised learning is termed non-contrastive learning. Many approaches such as BYOL [99], SimSiam [100], and others have been proposed to explore non-contrastive learning frameworks, and have shown strong results in various domains [101, 102].

## 2.3 Facial Recognition

### 2.3.1 Classical Methods

In [103, 104], the authors proposed using contour and edge detectors to identify facial landmarks and distinguish between different faces based on them. Principal Component Analysis (PCA) was used in [105, 106] to extract the features with the most variance from the training face dataset which were termed Eigenfaces. New faces were then projected onto the subspace spanned by the Eigenfaces and compared with the known Eigenfaces to identify the new face. Feature descriptors, such as Histogram of Gradients [107], Local Binary Pattern [108], and Scale-Invariant Feature Transform [109], and others have also been used to generate feature descriptions of faces, which are then used for identification or verification.

### 2.3.2 Deep Learning Methods

Deep learning methods use standard CNN architectures to generate high-dimensional feature embeddings from facial images, followed by classification [31]. [110], [111] used Inception [112] and VGG [113] as their CNN backbones respectively and used the Triplet loss to optimize their methods. Other methods use ResNets [114] as their backbone with novel loss functions such as  $L_2$ -softmax [115], AM-softmax [116], Ring Loss [117], A-softmax [118], CosFace [119], ArcFace [120], Circle Loss [121] and others, to generate better separable embeddings in this context. Another avenue of research in face recognition focuses on using modalities such as depth-maps [122, 123], light-field images [124, 125], thermal images [126], and others in a stand-alone or multi-modal manner.

## 2.4 Privacy Preservation

### 2.4.1 Privacy Attacks

Privacy attacks on deployed deep learning models can broadly be classified into white-box and black-box attacks [33]. In the white-box attack, the attacker has full knowledge of the deep learning model, including the architecture, parameters or weights, and the training data. In a black-box attack, the attacker has no knowledge of the deep learning model and has access only to the input/output behavior of the model. While white-box attacks are more severe than black-box attacks, they are far less realistic than the latter. However, in both settings, it is possible to reconstruct/obtain data used in the training process. The attacks described above are applicable to all kinds of data processed by deep learning models such as visual [127], textual [128], graph [129], and others. Recently, in [130], the authors were able to recover several of the training images used in the training of latent diffusion models which are widely popular, further highlighting the issue of the vulnerability of data leakage in deep learning-based systems.

### 2.4.2 General Privacy-preserving Methods

Existing privacy-preserving approaches can be broadly divided into two categories depending upon the nature of their mechanism, into *encryption* or *perturbation*. Methods using encryption include Homomorphic Encryption [131], Secure Multiparty Computation [132], and others. However, encryption methods often incur high computation costs making them unsuitable for many deep learning systems. The other set of methods use perturbation in the training process in an effort to reduce the risk of successful privacy attacks. The most common perturbation technique used is

Differential Privacy (DP) [133]. In DP, noise is added to the optimization process to prevent the model from strongly learning from any particular training sample. Specific to visual data, InstaHide [134] is a data perturbation method wherein every training sample is encoded using a weighted sum of itself and other training samples, after which the signs of the pixels of the composite image are randomly flipped. Another general privacy-preserving strategy for images has been explored in the works of [135, 136], wherein the authors divided a given image into blocks and then performed pixel perturbations within the blocks.

### 2.4.3 Privacy-preserving Methods for Faces

Specific to datasets containing faces, privacy-preserving approaches include the addition of noise, masking, and blurring, among others [137]. However, many of these approaches are known to be reversible [138], resulting in a need for privacy-preserving approaches for facial images with strong security and irreversibility. Other methods transform the face image to another domain such as in [139], wherein the authors take the Block Discrete Cosine Transform (BDCT) of the image, followed by channel-wise shuffling and combining to obtain the transformed image. Some methods seek to combine the concept of DP along with other transformations such as in [140, 141] where the authors add noise to the BDCT and the Eigenface representation of the face image respectively to make them privacy-preserving. Another set of works such as [142, 143] use GANs to generate synthetic faces for face de-identification while preserving information such as structural similarity, facial attributes and others for use in subsequent tasks. However, the effect of any privacy-preserving mechanism for rPPG extraction remains largely unexplored.

## Chapter 3

### Self-supervised Learning

#### 3.1 Method

In this section, we present our first proposed method in detail. Our proposed solution consists of several components, namely, an encoder, a contrastive learning framework, and loss functions.

##### 3.1.1 Solution Architecture

Our method consists of two separate main stages: (1) self-supervised contrastive pre-training, and (2) supervised fine-tuning. First, we take a raw input video clip, and detect RoI, namely the forehead and cheeks. Next, subsequent to the detection of the RoI, we enter the first stage of our method where the RoI clip is processed by a Data Augmentation Module to generate an augmented RoI clip. After this, the RoI and its augmented counterpart are passed through an encoder and subsequently, the projection head to generate lower-level feature embeddings. This is done for all the input RoI clips. The contrastive loss is then used to learn strong representations by maximizing the similarity between embeddings belonging to the same RoI clip, while minimizing

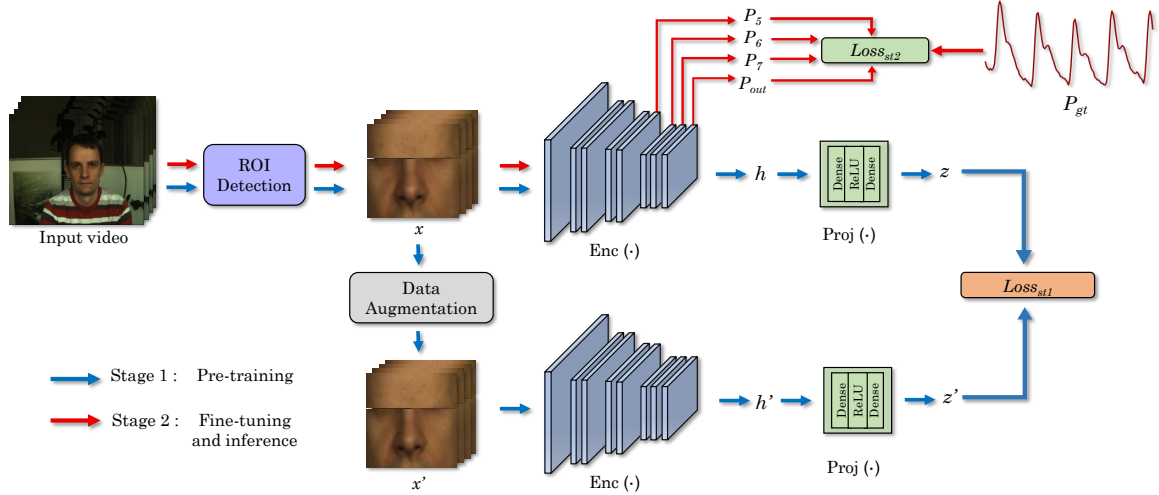


Figure 3.1: The overall layout of the proposed two-stage approach.

the similarity between embeddings from separate RoI clips. Subsequently, in stage 2, we use the encoder from stage 1 and fine-tune it using the RoI clip as the input and the corresponding PPG signal as the output via smooth L1 Loss. The architecture of our solution is illustrated in Figure 3.1. Through the following subsections, we describe each component of our solution mentioned above, in detail.

### 3.1.2 RoI Detection

Since observable changes in blood flow, and thus rPPG, are stronger around the forehead and the cheek regions [144, 145], we detect and crop these regions as our RoI, using the Dlib-face Detector [146] (see Figure 3.1). Next, we concatenate these regions for each frame and resize the outcome to  $64 \times 64$  pixels. For each video, we use a sliding window with a length of 128 frames and a stride of 8 frames to obtain several smaller clips. Similarly, we segment the ground-truth PPG signals such that the time-synchronicity between each video clip and the corresponding PPG segment is maintained (this will be used in the supervised fine-tuning stage). This cropping of the

RoI also provides a layer of security as it does not use the regions with high levels of discrimination in facial recognition such as the periorcular region, the lips, and others [147, 148, 31] while using the regions with relatively lower levels of discrimination namely the forehead and cheek [149].

### 3.1.3 Stage 1: Self-supervised Pre-training

As mentioned earlier, our self-supervised pre-training step consists of a data augmentation module, an encoder, and a projection head. Here we describe each component in detail.

**Data Augmentation.** This module applies a set of augmentations to the input RoI clip  $x$  with  $M$  frames,  $x_1, x_2, \dots, x_M$ , to generate  $x'$  which also consists of  $M$  frames. In the proposed method, we use two categories of augmentations: (i) spatial, and (ii) temporal. In terms of spatial augmentations, we employ the following:

- *Rotation*, where all the frames are rotated by the same angle  $\theta \in \mathbb{N}$ , where  $\theta$  is chosen randomly from  $\{1, 2, \dots, 360\}$ ;
- *Crop*, where for a frame with height  $H$  and width  $W$ , we randomly select a cropping scale  $\gamma \in \mathbb{R}$  from  $[0.25, 0.75]$ , and choose the cropping window anchor with coordinates  $i \in \mathbb{N}$  and  $j \in \mathbb{N}$ . The anchor coordinates are chosen randomly from  $\{0, 1, \dots, W - \gamma W\}$  and  $\{0, 1, \dots, H - \gamma H\}$  respectively. Accordingly, the crop is performed between  $(i, j)$  and  $(i + \gamma W, j + \gamma H)$ , followed by resizing of the output to  $H \times W$ ;
- *Flip*, where every frame is flipped along the vertical axis. Mathematically, for pixel value with coordinate  $(i, j)$  in frame  $x'_m$  from  $x'$  and corresponding frame  $x_m$  from  $x$ , we have  $x'_m(i, j) = x_m(W - i, j)$ .

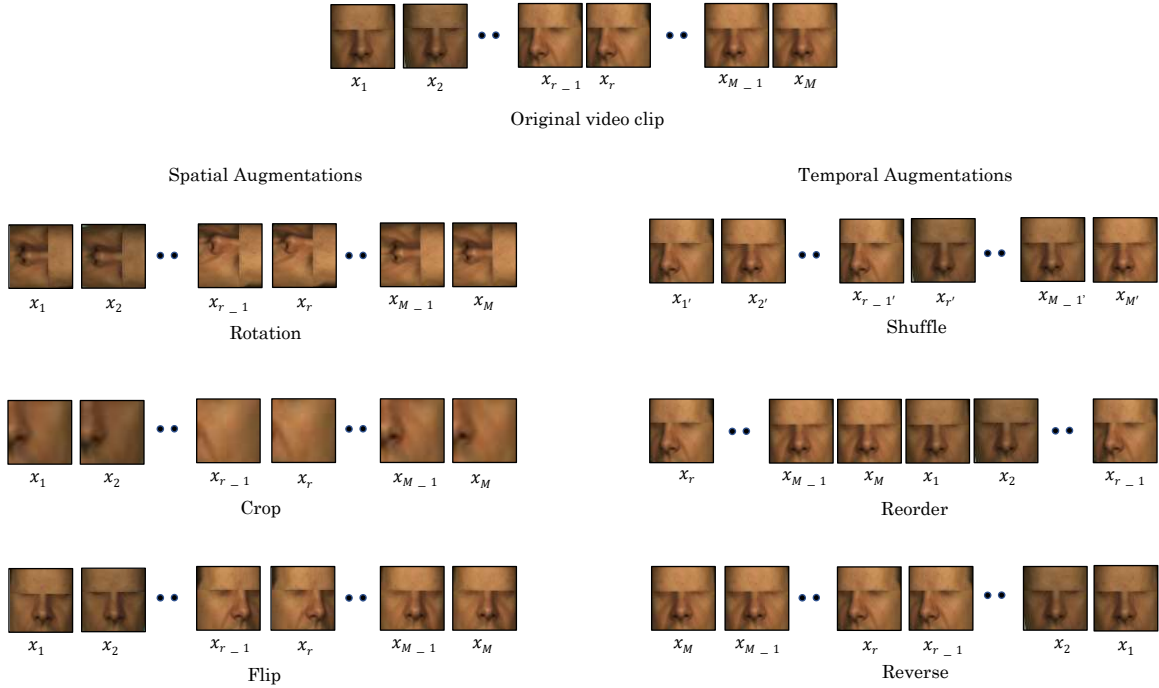


Figure 3.2: Example of a sample clip after being processed by different augmentations.

As mentioned, we also perform temporal augmentations, as follows:

- *Shuffle*, where frames  $x_1, x_2, \dots, x_M$  are shuffled randomly to obtain  $x'$  with a different order of frames  $x_{1'}, x_{2'}, x_{3'}, \dots, x_{M}'$ ;
- *Reorder*, where a random index  $r$  is selected to cut the video into two clips  $x_a = x_1, x_2, \dots, x_{r-2}, x_{r-1}$ ;  $x_b = x_r, x_{r+1}, x_{r+2}, \dots, x_{M-1}, x_M$ .  $x'$  is then synthesized as  $x' = [x_b, x_a]$ ;
- *Reverse*, where the order of frames is reversed to obtain  $x' = [x_M, x_{M-1}, \dots, x_2, x_1]$ .

We visualize the effect of the different augmentations on a sample input in Figure 3.2. In our experiments, input RoI clip  $x$  and its augmented counterpart  $x'$  make up a *positive* pair between which the similarity is maximized with contrastive learning

Table 3.1: Architectural details of the encoder used in our proposed method.

Block	Layers	Kernel Size	Output Size
Input	-	-	$128 \times 64 \times 64 \times 3$
ConvBlock1	Conv	16, [1,5,5]	$128 \times 62 \times 62 \times 16$
	AvgPool	[1,2,2]	$128 \times 31 \times 31 \times 16$
ConvBlock2	Conv	32, [3,3,3]	$128 \times 31 \times 31 \times 32$
	Conv	32, [3,3,3]	$128 \times 31 \times 31 \times 32$
	AvgPool	[1,2,2]	$128 \times 15 \times 15 \times 32$
ConvBlock3	Conv	64, [3,3,3]	$128 \times 15 \times 15 \times 64$
	Conv	64, [3,3,3]	$128 \times 15 \times 15 \times 64$
	AvgPool	[1,2,2]	$128 \times 7 \times 7 \times 64$
ConvBlock4	Conv	64, [3,3,3]	$128 \times 7 \times 7 \times 64$
	Conv	64, [3,3,3]	$128 \times 7 \times 7 \times 64$
	Conv	64, [3,3,3]	$128 \times 7 \times 7 \times 64$
	GlobalAvgPool	[1,7,7]	$128 \times 1 \times 1 \times 64$
	Squeeze	-	$128 \times 64$
	Aggregation	1, [1]	$128 \times 1$
Output	-	-	$128 \times 1$

(to be presented in Section 3.1.5). Alternatively, for two different input clips  $x_1$  and  $x_2$ , where  $x_1 \neq x_2$ , the samples  $(x_1, x_2)$ ,  $(x'_1, x_2)$ ,  $(x_1, x'_2)$ , and  $(x'_1, x'_2)$  constitute the *negative* pairs, between which the similarity is minimized.

**Encoder.** We use a 3D CNN architecture as our encoder. The initial input is passed through a  $1 \times 5 \times 5$  kernel that tends to extract information from each video frame. Next, our model performs 3D convolutions with kernel  $3 \times 3 \times 3$  on the resulting embeddings. The detailed architecture is given in Table 3.1. Each convolution operation is followed by a ReLU activation and batch-normalization. Mathematically, for any input  $x$ ,  $h = Enc(x)$ , where  $Enc(\cdot)$  is the encoder and  $h$  is the intermediate embedding of  $x$ .

**Projection Head.** Following the encoder, we use a projection head to map the obtained embedding onto a lower-dimensional space. To this end, we use a 2-layer dense neural network with 64 and 16 neurons to generate the low-dimensional embedding of the output of the encoder. The final embedding,  $z$  is given by  $z = Proj(h)$ , where

$Proj(\cdot)$  is the projection head.

### 3.1.4 Stage 2: Supervised Fine-tuning

For the second stage of the proposed method, we discard the projection head  $Proj(\cdot)$  and the data augmentation module from the previous stage and only use the encoder. We fine-tune the entire encoder by using the RoI clip as the input and the PPG signal as the output. Furthermore, instead of using the output of only the last layer of the encoder for training, we use the output embeddings of the final four convolutional layers. This enables for more effective representations to be learned throughout different parts of the encoder.

### 3.1.5 Loss Functions

Below we describe the loss functions used for each stage of our method.

**Contrastive Loss.** We use the contrastive loss presented in [96] for the pre-training stage of our model. This loss helps in learning representations that maximize the similarity between positive pairs while minimizing the similarity between negative samples. For any positive pair  $(x_m, x_n)$  with corresponding projections  $(z_m, z_n)$ , the cosine similarity is given by:

$$\text{cosine}(z_m, z_n) = \frac{z_m^T z_n}{\|z_m\| \cdot \|z_n\|}. \quad (3.1)$$

Subsequently, the loss function is given as:

$$\text{Loss}_{st1}(z_m, z_n) = -\log \frac{\exp(\text{cosine}(z_m, z_n)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq m]} \exp(\text{cosine}(z_m, z_k)/\tau)}, \quad (3.2)$$

where  $1_{[k \neq m]} \in \{0, 1\}$  is the indicator function and outputs 0 iff  $k = m$  and 1 otherwise. Also,  $\tau$  is the temperature hyper-parameter and  $2N$  is the total number of samples resulting from augmenting the original  $N$  samples.

**Smooth L1 Loss.** We use the smooth L1 loss [150] for the second stage of training. This loss is a combination of both L1 and the L2 losses, and allows for switching between the two depending upon the difference between the amplitude values of the rPPG signal  $P_{out}$ , and the ground-truth PPG signal  $P_{gt}$ . This loss is given by:

$$\mathcal{L}(P_{out}, P_{gt}) = \begin{cases} \frac{1}{2} \frac{(P_{out} - P_{gt})^2}{\beta}, & |P_{out} - P_{gt}| < \beta \\ |P_{out} - P_{gt}| - \frac{1}{2} * \beta, & otherwise. \end{cases} \quad (3.3)$$

where  $\beta$  is a hyperparameter. Here, when the absolute difference between the estimated and ground-truth signals is smaller than  $\beta$ , the loss uses the L2 loss, otherwise it uses L1. The L2 loss is quite sensitive to large errors due to its square operation. Thus, to obtain a smooth output, i.e., more effective training, the loss shifts to L1 for signals with larger differences. As mentioned earlier, we apply this loss to the output embeddings of the final four convolutional layers as opposed to only the final layer, which enables for more effective representations to be learned throughout different parts of our network. Accordingly, we calculate the final loss for stage 2 by:

$$Loss_{st2} = \mathcal{L}(P_{out}, P_{gt}) + \alpha \sum_{i=\lambda1}^{\lambda2} \mathcal{L}(P_i, P_{gt}), \quad (3.4)$$

where  $\alpha$ ,  $\lambda1$ , and  $\lambda2$  are the hyper-parameters for the weights and layers included in the loss calculation, set to 0.5, 5, and 7 respectively.

### 3.1.6 HR Calculation

After obtaining the estimated rPPG signals at runtime, similar to [25, 26] we calculate the HR by measuring the largest peak obtained from the Welch power spectrum of the signal.

## 3.2 Experiment Setup

In this section, we first describe the datasets used in our study. This is followed by the evaluation scheme and the metrics used for comparison of our solution to prior work. And finally, we discuss in detail the variations of our method which we use to validate our design choices.

### 3.2.1 Datasets

We use two publicly available datasets, COHFACE [30] and PURE [28], for our experiments. Following is a description of each dataset.

- **COHFACE [30]:** This dataset comprises 160 facial videos and their corresponding PPG. There are a total of 40 subjects (28 males, and 12 females) and each subject contributes 4 videos. The videos have been recorded under two illumination settings (natural lighting and studio lighting). In the natural lighting setting, the face of the subject is unevenly illuminated from the light coming from the window blinds next to the subject. In the case of studio lighting, the face is evenly illuminated from the ceiling light and a 400W halogen spotlight. The videos have been recorded with a Logitech HD Webcam C525 at 20 frames per second (fps) while the blood volume pulse signals have been recorded using

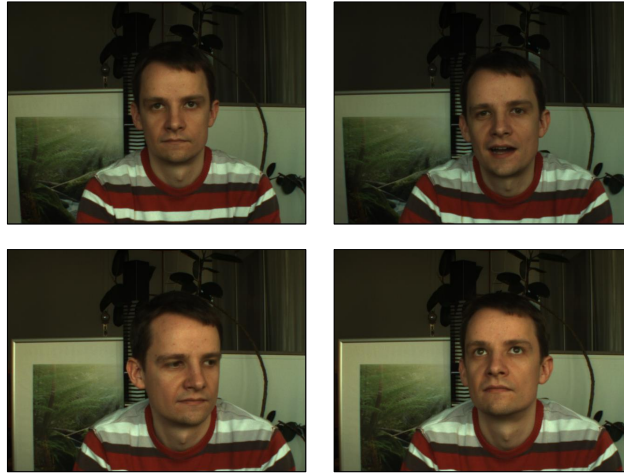


Figure 3.3: Sample frames showing the varying conditions in PURE dataset.

a contact TTL SA9308M sensor at 256 Hz sampling rate. The videos were compressed in MPEG-4 format with a resolution of  $640 \times 480$  pixels.

- **PURE [28]:** This dataset comprises 60 facial videos and their corresponding PPG. There are a total of 10 subjects (8 males, and 2 females) and each subject contributes 6 videos performing 6 different movements. The movements are steady sitting, talking, small face rotation, medium face rotation, slow face translation, and fast face translation. The videos have been recorded with an Eco274CVGE camera at 30 fps while the PPG signals have been recorded using finger pulse oximeter Pulox CMS50E at 60 Hz sampling rate. The videos have been stored with lossless compression in PNG format with resolution of  $640 \times 480$  pixels.

### 3.2.2 Evaluation Scheme and Metrics

For COHFACE, we use the subject split that has been designated and provided by the original authors of the dataset [30]. Specifically, the data from 24 subjects is used for training our model, while the data from the remaining 16 subjects is used for testing. For PURE, we use a 6-4 subject train-test split as commonly used in prior works such as [25, 26, 51].

To evaluate our model, the HR values obtained from the individual RoI clips of the same test video are averaged to generate one HR for each test video. These averaged HR values are then compared with the actual average HR calculated from the ground-truth PPG signals. The metrics we use for evaluation are, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) , both in beats per minute (bpm), along with Correlation ( $R$ ) of the predicted HR  $HR_{pred}$  and ground-truth HR  $HR_{gt}$ . For  $N$  test videos, the metrics are calculated as:

$$MAE = \frac{1}{N} \sum_{i=1}^N |HR_{pred}(i) - HR_{gt}(i)|, \quad (3.5)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (HR_{pred}(i) - HR_{gt}(i))^2}. \quad (3.6)$$

and

$$R = \frac{\sum_{i=1}^N (HR_{pred}(i) - \overline{HR_{pred}})(HR_{gt}(i) - \overline{HR_{gt}})}{\sqrt{\sum_{i=1}^N (HR_{pred}(i) - \overline{HR_{pred}})^2 \sum_{i=1}^N (HR_{gt}(i) - \overline{HR_{gt}})^2}}. \quad (3.7)$$

### 3.2.3 Comparisons

Here we briefly describe the other methods with which we compare our proposed solution.

**Prior Works.** We compare our work with several previous works discussed in Section 2.1. These works include [35, 34] which use classical image and signal processing approaches, as well as a large number of deep learning approaches. The deep learning approaches include methods with vanilla architectures such as [25, 26], two-stream approaches such as [50, 48], varied attention mechanisms [46, 45], methods combining classical approaches with deep learning such as [58, 62], and others.

**Video Representation Learning.** 3D convolutions are a common convolutional approach for processing 3D data such as videos. The 3D convolution does not distinguish among the dimensions of the data and treats the different dimensions equally. However, there are convolutional units such as [151, 52] and others which tend to break down the processing of spatiotemporal data across the dimensions to better process the information. Of these, the (2+1)-Dimensional ((2+1)D) convolution is widely used in video-based supervised as well as self-supervised learning [92, 152, 26]. In a (2+1)D convolution, the 3D convolution is decomposed into a combination of spatial (2D) and temporal (1-Dimensional (1D)) convolutions. The 2D convolution first extracts the spatial features from the input, after which the 1D convolution extracts the temporal features from these intermediate embeddings to complete the spatio-temporal processing. The separate processing is appropriate for rPPG estimation since there is less spatial variation in the facial videos as compared to the temporal one. There are also additional non-linearities (batch-normalization and ReLU) introduced in the intermediate step which helps in learning better representations. While our main solution uses 3D convolutions, for comparison purposes, we follow [151] to implement the (2+1)D approach. We use kernel sizes of  $1 \times 3 \times 3$  and  $3 \times 1 \times 1$  for the spatial and temporal convolution respectively. A detailed layout of the (2+1)D version of the

Table 3.2: Architectural details of the (2+1)D encoder (Encoder B) used in our experiments.

Block	Layers	Kernel Size	Output Size
Input	-	-	$128 \times 64 \times 64 \times 3$
ConvBlock1	Conv	16, [1,5,5]	$128 \times 62 \times 62 \times 16$
	AvgPool	[1,2,2]	$128 \times 31 \times 31 \times 16$
ConvBlock2	Conv	57, [1,3,3]	$128 \times 31 \times 31 \times 57$
	Conv	32, [3,1,1]	$128 \times 31 \times 31 \times 32$
	Conv	72, [1,3,3]	$128 \times 31 \times 31 \times 72$
	Conv	32, [3,1,1]	$128 \times 31 \times 31 \times 32$
	AvgPool	[1,2,2]	$128 \times 15 \times 15 \times 32$
ConvBlock3	Conv	115, [1,3,3]	$128 \times 15 \times 15 \times 115$
	Conv	64, [3,1,1]	$128 \times 15 \times 15 \times 64$
	Conv	144, [1,3,3]	$128 \times 15 \times 15 \times 144$
	Conv	64, [3,1,1]	$128 \times 15 \times 15 \times 64$
	AvgPool	[1,2,2]	$128 \times 7 \times 7 \times 64$
ConvBlock4	Conv	144, [1,3,3]	$128 \times 7 \times 7 \times 144$
	Conv	64, [3,1,1]	$128 \times 7 \times 7 \times 64$
	GlobalAvgPool	[1,7,7]	$128 \times 1 \times 1 \times 64$
	Squeeze	-	$128 \times 64$
	Aggregation	1, [1]	$128 \times 1$
Output	-	-	$128 \times 1$

encoder is presented in Table 3.2 while the convolution operation is depicted in Figure 3.4.

**Negative Pairs for Self-supervised Pre-training.** We also study the effect of using negative pairs in the self-supervised pre-training stage of our proposed method. While SimCLR [96] uses both the positive and negative pairs to train the self-supervised learning algorithm, there are other self-supervised techniques that do not use negative pairs. One such self-supervised paradigm is SimSiam [100], which we adopt for comparison purposes.

The framework used in SimSiam is similar to SimCLR, but with the inclusion of another dense network, the prediction head or the predictor. Similar to the projection

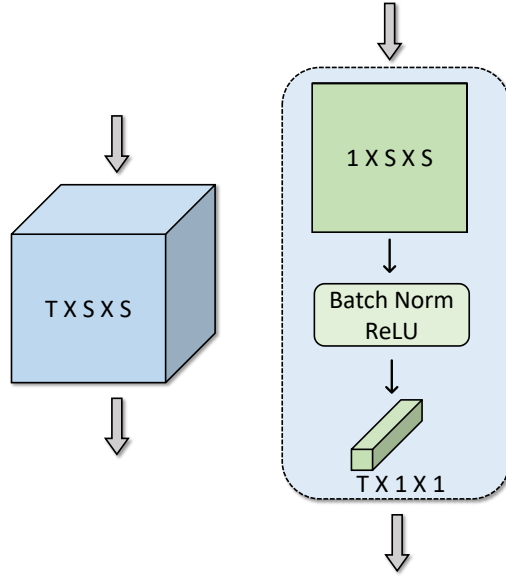


Figure 3.4: Illustration of the 3D and (2+1)D convolutions. T stands for the temporal filter size, while S stands for the spatial filter size.

head discussed in Section 3.1.3, the prediction head is used to map the lower level embedding  $z$  to another embedding space such that  $p = Pred(z)$ , where  $p$  is the prediction embedding and  $Pred(\cdot)$  is the prediction head. SimSiam trains the model such that the predictor learns to predict the representation of one view of the input such that it is similar to the projection of another. Through this, the very essential features of the input are learned by the model. SimSiam uses the predictor in only one branch of the model while applying a stop-gradient to the other. In this manner, the projection of one view of the input is constant with respect to the prediction of the other. The objective of SimSiam is to minimize the negative cosine similarity  $D$  between  $p$  and  $z$  given by:

$$D(p, z) = -\frac{p^T \cdot z}{\|p\| \cdot \|z\|}. \quad (3.8)$$

To ensure that both views of the input are processed by both branches of the framework,

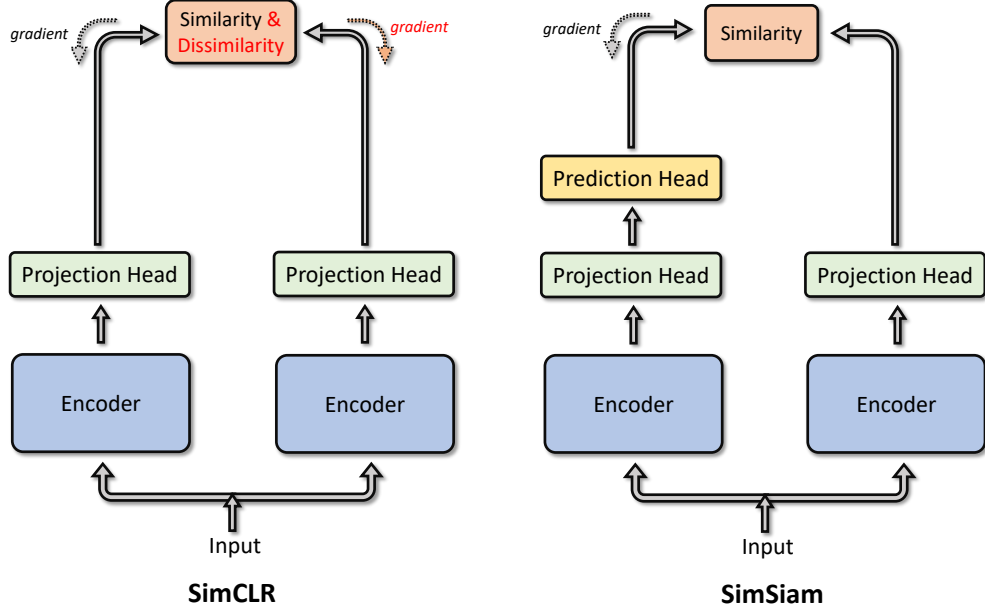


Figure 3.5: An overview of the self-supervised learning strategies used in this study.

the loss function is symmetrized. Therefore for a positive input pair  $(x_m, x_n)$ , the loss is given as:

$$Loss_{sim}(m, n) = \frac{1}{2}D(p_m, stopgrad(z_n)) + \frac{1}{2}D(p_n, stopgrad(z_m)). \quad (3.9)$$

An overview of the key differences among the self-supervised learning approaches is presented in Figure 3.5. For SimSiam, we use a 3-layer dense neural network with 64, 32, and 32 neurons as the projection head and a 2-layer dense neural network with 8 and 32 neurons as the prediction head. After pre-training, we follow the same procedure as our proposed method for fine-tuning the encoder for the rPPG estimation.

Table 3.3: Comparison of our proposed method with prior works on COHFACE.

Method	MAE↓	RMSE↓	R↑
LiCVPR [40]	19.98	25.59	-0.44
2SR [42]	20.98	25.84	-0.32
CHROM [34]	7.8	12.45	0.26
POS [35]	13.43	17.05	0.07
HR-CNN [25]	8.10	10.78	0.29
PhysNet [44]	8.59	11.60	0.36
CNN+ConvLSTM [48]	7.31	11.88	0.36
CAN [50]	6.89	13.89	0.34
DeeprPPG [26]	3.07	7.06	0.86
VitaSi [62]	7.16	9.59	0.61
MultiHeirCNN [47]	5.57	7.69	0.75
ETA-rPPGNet [46]	4.67	6.65	-
CNN+Att. [45]	5.19	7.52	-
CDConv-CAN [51]	<u>1.71</u>	<b>3.57</b>	<b>0.96</b>
TFA-PFE [57]	<b>1.31</b>	3.92	-
Sup. (3D)	2.62	4.59	0.90
Sup. ((2+1)D)	2.68	4.42	0.90
Ours w/o neg.	2.45	4.25	0.92
Ours	2.16	<u>3.61</u>	<u>0.94</u>

### 3.2.4 Implementation

The batch sizes for stage 1 and stage 2 of the training are set to 16 and 8, where we train the model for 50 and 10 epochs, respectively. The learning rates are set to 1e-4 and 2e-4 for stages 1 and 2, respectively, with Adam [153] used as the optimizer for both. The value of  $\beta$  is taken as 1 for one of the datasets (COHFACE) and 0.3 for the other (PURE). All the codes are written in PyTorch [154] and run on an NVIDIA GTX 2080 Ti GPU.

Table 3.4: Comparison of our proposed method with prior works on PURE.

Method	MAE↓	RMSE↓	R↑
LiCVPR [40]	28.22	30.96	-0.38
2SR [42]	2.44	3.06	0.98
CHROM [34]	2.07	2.50	0.99
POS [35]	3.14	10.57	0.95
HR-CNN [25]	1.84	2.37	0.98
PhysNet [44]	1.90	3.44	0.98
CNN+ConvLSTM [48]	0.88	1.58	0.99
CAN [50]	0.83	1.54	0.99
DeeprPPG [26]	<u>0.28</u>	<b>0.43</b>	0.99
ETA-rPPGNet [46]	0.34	0.77	-
CNN+Att. [45]	0.74	1.21	1.00
POS+MOT+CNN [58]	<b>0.23</b>	<u>0.48</u>	0.99
CDConv-CAN [51]	0.78	1.07	0.99
TFA-PFE [57]	1.44	2.50	-
Sup. (3D)	0.47	0.58	0.99
Sup. ((2+1)D)	0.50	0.62	0.99
Ours w/o neg.	0.46	0.58	0.99
Ours	0.43	0.58	0.99

### 3.3 Results and Discussions

In this section we present and discuss our results. We first compare the results with the existing prior works described above. Thereafter we study the impact of using a different technique for video representation learning and the impact of using negative pairs for self-supervised pre-training. Moreover, we study the effects of using different RoIs, namely the combined and individual regions of the forehead and the cheek and also the different augmentations for self-supervised pre-training. Lastly, we also compare the performance of the supervised and the proposed self-supervised method on reduced amounts of labelled data.

Table 3.5: Comparison of our proposed method with prior works on PURE (MPEG-4).

Method	MAE↓	RMSE↓	R↑
LiCVPR [40]	28.39	31.10	-0.42
2SR [42]	5.78	12.81	0.98
CHROM [34]	6.29	11.36	0.55
HR-CNN [25]	8.72	11.00	0.70
PhysNet [44]	5.39	11.05	-
CAN [50]	3.10	9.37	-
ETA-rPPGNet [46]	2.66	6.48	-
Sup. (3D)	0.97	1.2	0.99
Sup. ((2+1)D)	1.06	1.52	0.99
Ours w/o neg.	<u>0.78</u>	<u>0.95</u>	0.99
Ours	<b>0.74</b>	<b>0.93</b>	0.99

### 3.3.1 Performance and Comparison

Tables 3.3, and 3.4 present the results of our method on COHFACE and PURE, in comparison to prior works. The results show that our method approaches the state-of-the-art on both datasets with respect to all three evaluation metrics. A number of prior works [25, 46] have additionally used a compressed version of PURE dataset in MPEG-4 Visual format, which is denoted by ‘PURE (MPEG-4)’. We also use this approach for a more thorough evaluation of our solution, given that this compression is lossy, meaning that the quality of the data will decrease. In Table 3.5, we observe that on this dataset, our method achieves superior results compared to other works, indicating low sensitivity with respect to data quality.

Additionally, we implement two supervised versions of our model, one using the same 3D convolutions used in our final solution, while in the other, we use (2+1)D convolutions. We observe that our proposed method outperforms all the baselines by considerable margins on COHFACE and with smaller margins on PURE, demonstrating the clear benefits of the self-supervised aspect of our approach. This stands for both

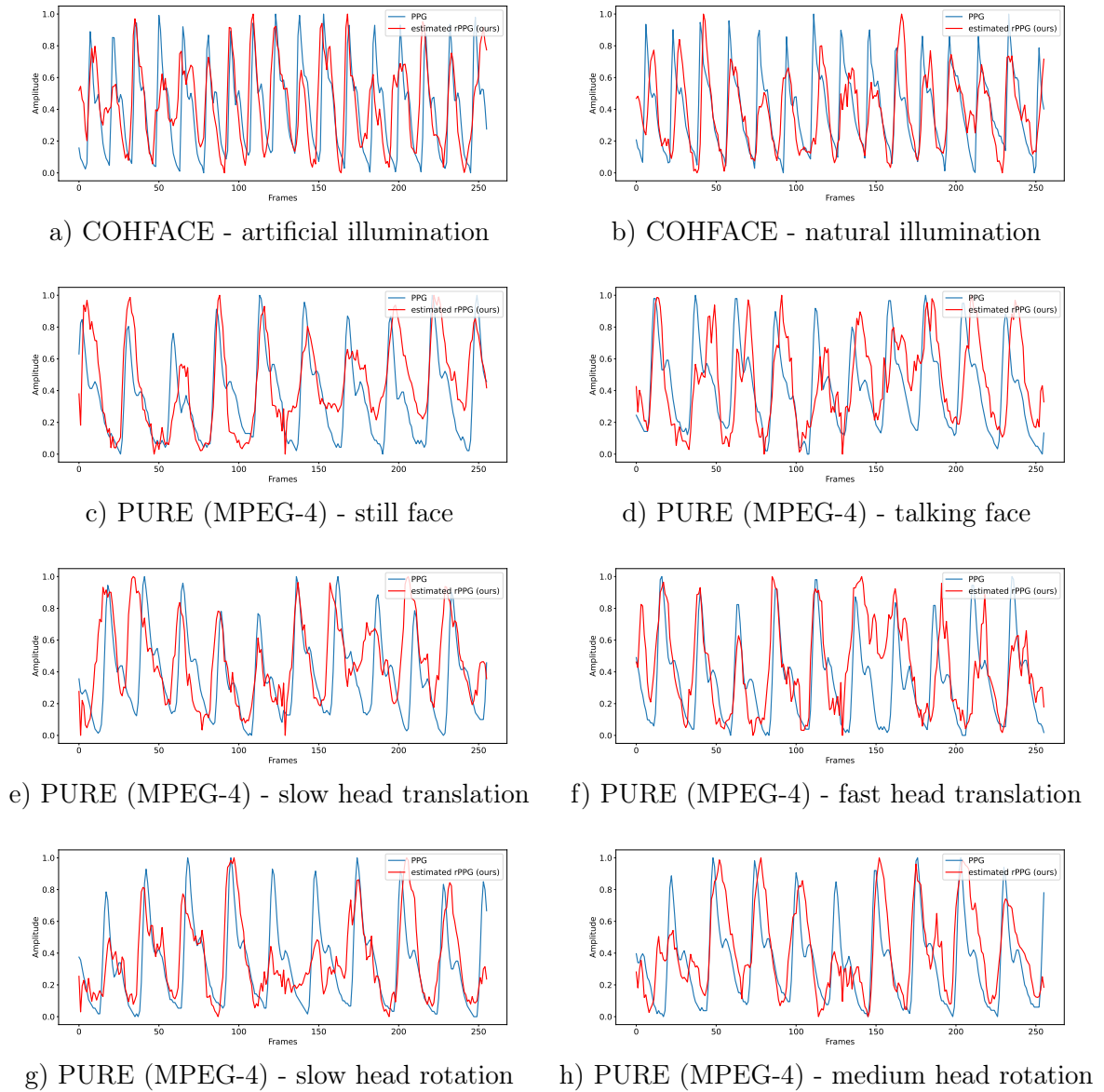


Figure 3.6: Visualization of predicted rPPG for different conditions presented in the datasets.

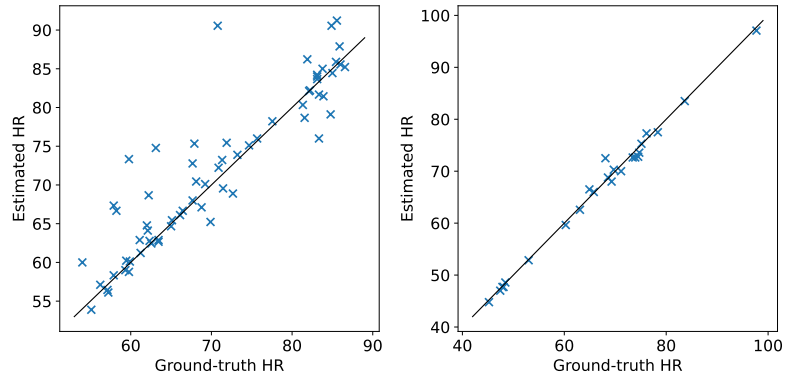


Figure 3.7: Correlation plots for COHFACE (left), and PURE (MPEG-4) (right).

cases of using as well as not using the negative pairs for the self-supervised pre-training.

Since PURE is stored in lossless format, our work and several prior works obtain an MAE of less than 1, and in some cases even less than 0.5. Moreover the improvement of self-supervised training over fully-supervised training is minimal (less than 0.1 in MAE). However, we notice that there is considerable improvement of self-supervised learning over the supervised method when using PURE (MPEG-4) instead of PURE. To better study the effects of using self-supervised learning over supervised learning and to further evaluate the performance of our proposed solution with respect to artifacts introduced through video compression [155], we only use PURE (MPEG-4) along with COHFACE for all the subsequent experiments.

To gain a better understanding about the quality of the rPPG produced by our model, we visualize sample segments of the estimated rPPG along with the ground-truth PPG in Figure 3.6 for varying conditions posed in the datasets. We observe that our model produces high quality rPPG signals, especially with the peaks being highly aligned with the corresponding PPG, which is the key factor in measuring metrics such as HR and heart rate variability. We also explore the correlation and the

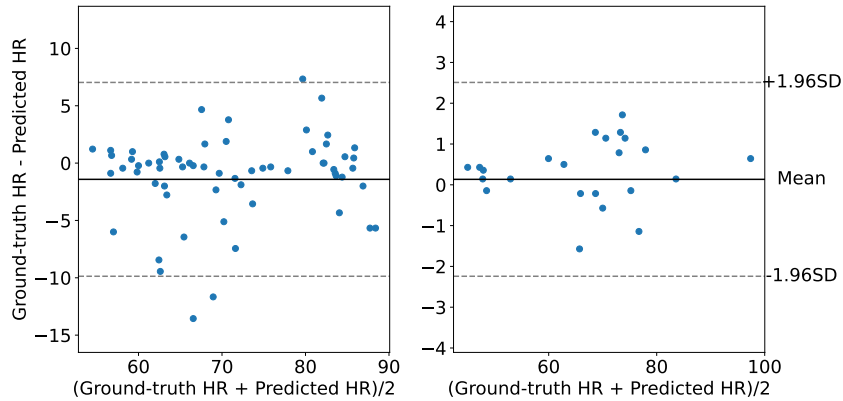


Figure 3.8: Bland-Altman plots for COHFACE (left), and PURE (MPEG-4) (right).

Bland-Altman [156] plots to better visualize the relation between our results and the ground-truth in Figures 3.7 and 3.8. As can be seen in the correlation plots between the predicted and the ground-truth HR values, our results correlate well with the ground-truth values with very few outliers. Similarly in the Bland-Altman plots, our results generally lie within the limits of agreement for both the datasets.

### 3.3.2 Impact of 3D Convolutions

In Tables 3.6, 3.7, 3.8, 3.9, 3.10, and 3.11, we compare the effect of using different video encoding methods by experimenting with 3D and (2+1)D convolutions. Encoder A refers to the encoder using 3D convolution while Encoder B refers to the encoder using (2+1)D convolutions. In the case of the fully-supervised baselines for all three instances of the RoI, using (2+1)D convolutions gives comparable results to the 3D counterpart. However, among the results obtained after self-supervised pre-training and fine-tuning, the best results across all the three metrics were obtained for the 3D convolution-based encoder. Since rPPG estimation relies on very slight changes in skin color, the additional non-linearities brought along with using (2+1)D convolution

Table 3.6: Impact of different encoders for pre-training (full RoI) for COHFACE.

Augmentation	Encoder A			Encoder B		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	2.96	4.44	0.90	3.09	5.58	0.86
Rot	2.78	4.84	0.88	<b>2.14</b>	3.61	<b>0.94</b>
Flip	<b>2.16</b>	<b>3.61</b>	<b>0.94</b>	2.57	4.08	0.92
Reverse	2.51	3.98	0.93	2.18	<b>3.50</b>	0.94
Reorder	2.59	4.32	0.91	2.48	4.07	0.91
Shuffle	2.22	3.67	0.94	2.40	3.68	0.93
Sup.	2.62	4.59	0.90	2.68	4.42	0.90

Table 3.7: Impact of different encoders for pre-training (full RoI) for PURE (MPEG-4).

Augmentation	Encoder A			Encoder B		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	0.83	1.22	0.99	1.11	1.43	0.99
Rot	0.81	1.05	0.99	1.25	1.82	0.99
Flip	0.88	1.20	0.99	1.10	1.60	0.99
Reverse	0.96	1.28	0.99	<b>0.72</b>	<b>1.02</b>	0.99
Reorder	1.18	1.79	0.99	1.46	2.81	0.97
Shuffle	<b>0.74</b>	<b>0.93</b>	0.99	1.58	2.92	0.97
Sup.	0.97	1.20	0.99	1.06	1.52	0.99

might interfere with the training process rather than helping in some cases.

Nevertheless, there is substantial improvement shown by the self-supervised approach over the supervised approach for both the encoders. For COHFACE, compared to the fully supervised learning baseline for Encoder A, the self-supervised learning approach using the *flip* augmentation, reduces MAE from 2.62 to 2.16, and RMSE from 4.59 to 3.61, while increasing  $R$  from 0.90 to 0.94. For PURE (MPEG-4), compared to the fully supervised learning baseline, the self-supervised learning approach using the *shuffle* augmentation, reduces MAE from 0.97 to 0.74 and RMSE from 1.20 to 0.93. Likewise, for COHFACE, compared to the fully supervised learning baseline for Encoder B, the self-supervised learning approach using the *rotation* augmentation,

Table 3.8: Impact of different encoders for pre-training (cheek as RoI) for COHFACE.

Augmentation	Encoder A			Encoder B		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	3.66	6.09	0.83	3.57	5.56	0.84
Rot	<b>2.55</b>	<b>3.92</b>	<b>0.90</b>	<b>2.63</b>	<b>4.02</b>	<b>0.89</b>
Flip	2.78	4.71	0.89	3.63	6.04	0.85
Reverse	2.88	4.81	0.89	3.45	5.37	0.86
Reorder	3.23	5.08	0.88	3.05	5.25	0.87
Shuffle	3.14	5.17	0.87	3.12	4.70	0.88
Sup.	3.03	5.17	0.87	3.28	5.31	0.85

Table 3.9: Impact of different encoders for pre-training (cheek as RoI) for PURE (MPEG-4).

Augmentation	Encoder A			Encoder B		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	1.93	3.00	0.97	1.65	<b>2.23</b>	0.98
Rot	1.13	1.56	0.99	<b>1.52</b>	2.41	0.98
Flip	<b>0.89</b>	<b>1.25</b>	<b>0.99</b>	1.74	2.45	0.98
Reverse	1.21	1.89	0.99	1.73	2.55	0.98
Reorder	1.58	2.20	0.98	1.83	2.63	0.98
Shuffle	1.48	2.10	0.98	1.99	3.18	0.97
Sup.	1.46	2.64	0.98	1.84	2.44	0.98

reduces MAE from 2.68 to 2.14 and RMSE from 4.42 to 3.61, while increasing  $R$  from 0.90 to 0.94. For PURE (MPEG-4), compared to the fully supervised learning baseline, the self-supervised learning approach using the *reverse* augmentation, reduces MAE from 1.06 to 0.72 and RMSE from 1.52 to 1.02. We observe similar improvements when using the cheek and the forehead as separate RoIs in Tables 3.8, 3.9, 3.10 and 3.11, except in the setting with using the forehead as the RoI along with Encoder B for COHFACE where the MAE are very similar.

Table 3.10: Impact of different encoders for pre-training (forehead as RoI) for CO-HFACE.

Augmentation	Encoder A			Encoder B		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	4.68	7.11	0.78	4.74	7.33	0.73
Rot	<b>4.01</b>	<b>5.55</b>	<b>0.84</b>	4.79	<b>7.13</b>	<b>0.80</b>
Flip	4.67	7.22	0.75	5.00	7.50	0.72
Reverse	4.54	6.55	0.77	5.53	8.25	0.66
Reorder	5.11	7.43	0.72	5.31	8.48	0.66
Shuffle	5.28	7.93	0.74	5.71	8.43	0.71
Sup.	4.96	7.32	0.71	<b>4.73</b>	7.33	0.72

Table 3.11: Impact of different encoders for pre-training (forehead as RoI) for PURE (MPEG-4).

Augmentation	Encoder A			Encoder B		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	2.55	3.62	0.96	3.13	4.94	0.94
Rot	2.23	3.37	0.97	2.12	3.37	0.96
Flip	2.55	3.36	0.97	2.22	3.14	0.97
Reverse	2.38	3.33	0.96	2.06	3.60	0.96
Reorder	<b>1.90</b>	<b>2.58</b>	<b>0.98</b>	4.70	8.24	0.81
Shuffle	2.36	3.88	0.95	<b>1.87</b>	<b>2.90</b>	<b>0.97</b>
Sup.	2.47	4.10	0.95	2.61	4.44	0.94

### 3.3.3 Impact of Negative Pairs in Pre-training

Next, in Tables 3.12, 3.13, 3.14, 3.15, 3.16, and 3.17, we compare the effect of using negative pairs for self-supervised pre-training. We observe that in some cases not using the negative pairs for this purpose yields better results in comparison to when the negative pairs are used. However, the best results for almost all RoIs for both the datasets are obtained when using the negative pairs for the pre-training. The one exception being the case of using forehead alone as the RoI for PURE (MPEG-4) wherein the MAE are very similar.

In other cases, although not using the negative pairs do not give the best results,

Table 3.12: Impact of including and excluding negative pairs in pre-training (full RoI) for COHFACE.

Augmentation	w/ negative pairs			w/o negative pairs		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	2.96	4.44	0.90	2.54	4.68	0.89
Rot	2.78	4.84	0.88	2.78	4.51	0.91
Flip	<b>2.16</b>	<b>3.61</b>	<b>0.94</b>	<b>2.45</b>	<b>4.25</b>	<b>0.92</b>
Reverse	2.51	3.98	0.93	2.76	4.56	0.90
Reorder	2.59	4.32	0.91	2.59	4.28	0.91
Shuffle	2.22	3.67	0.94	2.82	5.26	0.88
Sup.	2.62	4.59	0.90	2.62	4.59	0.90

Table 3.13: Impact of including and excluding negative pairs in pre-training (full RoI) for PURE (MPEG-4).

Augmentation	w/ negative pairs			w/o negative pairs		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	0.83	1.22	0.99	<b>0.78</b>	<b>0.95</b>	0.99
Rot	0.81	1.05	0.99	0.99	1.35	0.99
Flip	0.88	1.20	0.99	0.92	1.13	0.99
Reverse	0.96	1.28	0.99	0.84	1.05	0.99
Reorder	1.18	1.79	0.99	1.09	1.85	0.99
Shuffle	<b>0.74</b>	<b>0.93</b>	0.99	0.93	1.40	0.99
Sup.	0.97	1.20	0.99	0.97	1.20	0.99

they still give improvements over the fully-supervised baselines. For COHFACE, while processing the combined RoI, using the self-supervised learning approach without negative pairs with the *flip* augmentation, MAE is reduced from 2.62 to 2.45 and RMSE from 4.59 to 4.25, while increasing  $R$  from 0.90 to 0.92. Likewise for PURE (MPEG-4), using the *crop* augmentation reduces MAE from 0.97 to 0.78 and RMSE from 1.20 to 0.95. A similar trend can be observed when using the cheek and the forehead as individual inputs. Overall, we observe in this experiment that the use of negative pairs generally benefits our solution. This can be due to the fact that for the problem of rPPG estimation, it is highly unlikely that two input clips would

Table 3.14: Impact of including and excluding negative pairs in pre-training (cheek as RoI) for COHFACE.

Augmentation	w/ negative pairs			w/o negative pairs		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	3.66	6.09	0.83	2.87	4.96	0.87
Rot	<b>2.55</b>	<b>3.92</b>	<b>0.90</b>	3.38	5.29	0.86
Flip	2.78	4.71	0.89	3.27	5.58	0.84
Reverse	2.88	4.81	0.89	<b>2.75</b>	<b>4.42</b>	<b>0.90</b>
Reorder	3.23	5.08	0.88	2.93	4.61	0.89
Shuffle	3.14	5.17	0.87	3.01	4.95	0.87
Sup.	3.03	5.17	0.87	3.03	5.17	0.87

Table 3.15: Impact of including and excluding negative pairs in pre-training (cheek as RoI) for PURE (MPEG-4).

Augmentation	w/ negative pairs			w/o negative pairs		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	1.93	3.00	0.97	1.27	1.65	0.99
Rot	1.13	1.56	0.99	<b>0.93</b>	<b>1.29</b>	0.99
Flip	<b>0.89</b>	<b>1.25</b>	0.99	1.47	2.09	0.99
Reverse	1.21	1.89	0.99	1.66	2.68	0.98
Reorder	1.58	2.20	0.98	1.60	2.50	0.98
Shuffle	1.48	2.10	0.98	1.18	1.70	0.99
Sup.	1.46	2.64	0.98	1.46	2.64	0.98

have identical PPG patterns, or in other words, identical labels (given that we have a regression problem). Thereby when we use negative pairs in our setup, the network essentially learns to distinguish even between those clips which might have some spatial similarities or even similar HR values, yet still different PPG patterns. This will allow for more effective representations to be learned to achieve better overall performance.

### 3.3.4 Impact of Different Facial Regions

Here, we study the effects of using different facial regions for rPPG estimation. To do so, we use different regions of the face, namely cheeks, forehead, and a combination of both (our original solution), as input to our model. Since the input dimensions will differ when the cheeks or the forehead alone are used, we perform spatial interpolation to scale the input to  $64 \times 64$  pixels (our original input spatial dimension). Tables 3.8, 3.9, 3.10 and 3.11 present the results where we observe (considering Encoder A) that when using only the cheek, we obtain results closer to the ones obtained when using the combined RoI of the cheek and forehead. However, this is not the case when we use the forehead as the sole input. This can be primarily due to the forehead region being partially occluded by hair, having wrinkles and other artifacts. Moreover, we observe that there is significant improvement while using the self-supervised pre-training over fully-supervised baselines when we use the facial regions separately. For COHFACE, when using cheek as the RoI, using the self-supervised learning approach with *rotation* augmentation, reduced MAE from 3.03 to 2.55, RMSE from 5.17 to 3.92, and increased  $R$  from 0.87 to 0.90 while for the forehead, the self-supervised learning approach using *rotation* augmentation, reduced the MAE from 4.96 to 4.01, RMSE from 7.32 to 5.55, and increased  $R$  from 0.71 to 0.84. Likewise for PURE (MPEG-4), when using cheek as the RoI, using the self-supervised learning approach with *flip* augmentation, reduced the MAE from 1.46 to 0.89, RMSE from 2.64 to 1.25, and increased  $R$  from 0.98 to 0.99 and while for the forehead, the self-supervised learning approach using *reorder* augmentation, reduced the MAE from 2.47 to 1.90, RMSE from 4.10 to 2.58, and increased  $R$  from 0.95 to 0.98. Nevertheless, there are considerable differences in the values of the metrics obtained whilst using the cheek and the forehead as standalone

Table 3.16: Impact of including and excluding negative pairs in pre-training (forehead as RoI) for COHFACE.

Augmentation	w/ negative pairs			w/o negative pairs		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	4.68	7.11	0.78	4.89	7.24	0.76
Rot	<b>4.01</b>	<b>5.55</b>	<b>0.84</b>	<b>4.46</b>	<b>6.65</b>	<b>0.77</b>
Flip	4.67	7.22	0.75	5.05	7.12	0.76
Reverse	4.54	6.55	0.77	4.82	7.04	0.74
Reorder	5.11	7.43	0.72	4.89	7.86	0.73
Shuffle	5.28	7.93	0.74	5.00	7.66	0.73
Sup.	4.96	7.32	0.71	4.96	7.32	0.71

Table 3.17: Impact of including and excluding negative pairs in pre-training (forehead as RoI) for PURE (MPEG-4).

Augmentation	w/ negative pairs			w/o negative pairs		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
Crop	2.55	3.62	0.96	1.91	<b>2.71</b>	<b>0.98</b>
Rot	2.23	3.37	0.97	<b>1.88</b>	2.75	0.97
Flip	2.55	3.36	0.97	2.50	3.41	0.97
Reverse	2.38	3.33	0.96	2.35	3.37	0.96
Reorder	<b>1.90</b>	<b>2.58</b>	<b>0.98</b>	2.48	3.77	0.96
Shuffle	2.36	3.88	0.95	2.23	3.28	0.97
Sup.	2.47	4.10	0.95	2.47	4.10	0.95

RoIs. A similar trend can be observed in Tables 3.14, 3.15, 3.16, and 3.17, wherein we do not use the negative pairs of the input RoI in the self-supervised pre-training.

### 3.3.5 Impact of Different Augmentations

Next, we explore the impact of different augmentations. Here, we only consider our best setups, i.e., Encoder A with negative pairs using the combined RoI. Revisiting Tables 3.6 and 3.7, we notice that for COHFACE, *flip* augmentation yields the best results, while for PURE (MPEG-4), *shuffle* results in the best performance. However, we observe that with the exception of *flip*, the temporal augmentations gave better

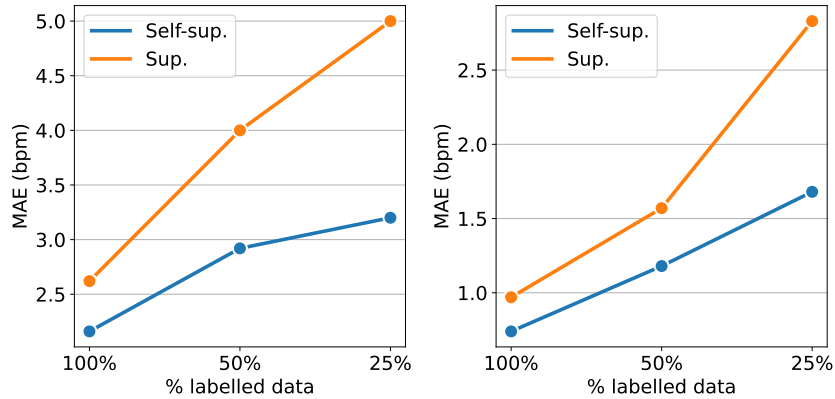


Figure 3.9: Performance of self-supervised and fully supervised approaches on reduced amounts of labelled data for COHFACE (left), and PURE (MPEG-4) (right).

results than the spatial augmentations for COHFACE. Similarly in PURE (MPEG-4), with the exception of *shuffle*, the spatial augmentations provide better results than the temporal ones. We should note that the subjects in COHFACE dataset have less spatial variations compared to the subjects in PURE, as the subjects in PURE were recorded with varying facial movements while the subjects in COHFACE were stationary. When using contrastive learning, robust feature representations are learnt which make the model invariant to the augmentation used to generate the pairs [157], which might be the reason for the better overall performance of spatial augmentations in PURE (MPEG-4). Nevertheless, the best performances are given by a spatial augmentation in COHFACE, and a temporal augmentation in the case of PURE (MPEG-4), demonstrating the need for exploration of a wide variety of augmentations for use in contrastive learning [158].

### 3.3.6 Performance on Reduced Labels

Lastly, to further illustrate the advantage of using self-supervised vs. fully supervised learning, we compare the two approaches on reduced amounts of *labeled* data. We first train the encoder (Encoder A) on *all* the video clips of the training data through contrastive learning. Next, for fine-tuning, we randomly select 50% and 25% of the video clips and their corresponding PPG signals from the training data, and fine-tune the network using the smooth L1 Loss. For the supervised learning method, we train *Sup. (3D)* from scratch on the same randomly selected video clips and PPG signals. Figure 3.9 presents the performance on the full and reduced (50%, 25%) training sets when the best augmentations are used to train the model. As we observe, the self-supervised approach leads to more robustness (suffers smaller drops in performance) when dealing with reduced labels on both the datasets.

## Chapter 4

### Privacy Preservation

#### 4.1 Method

##### 4.1.1 Problem Setup

Let  $D$  be a dataset comprising face videos  $F$  and corresponding PPG signals  $P$  for  $N$  subjects. Our goal is to design  $Q$ , a transformation with strong security, such that  $M_{\theta'}(Q(F)) \approx M_{\theta}(F) \approx P$ , where  $M$  is a deep learning model with learned parameters  $\theta$ , such that  $Q(\cdot)$  conceals the identities of the subjects in  $D$ , while  $M$  is able to appropriately estimate the rPPG signal from  $F$  or  $Q(F)$ .

##### 4.1.2 Perturbation Method

Let us assume a sample video  $F_i$  comprising a total of  $t$  frames  $f^1, f^2, \dots, f^t$ . These frames contain the face along with the background which serves no purpose in the rPPG estimation and might rather hinder it. Hence, we first detect the face in each frame using the Multi-Task Cascaded Convolutional Network face detector [159] and subsequently align and crop it. Next, we use Dlib [146] to detect facial landmarks which are subsequently used to crop the left cheek, right cheek, and forehead from

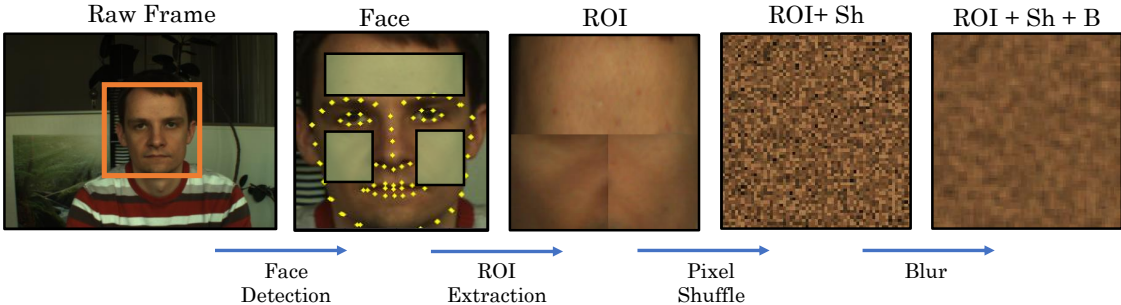


Figure 4.1: An overview of our proposed privacy-preserving data perturbation pipeline.

each frame. We crop these regions and use them instead of the entire face as the rPPG signals are stronger in these regions [160]. Moreover, we hypothesize that these regions contain less identity-related information, and are thus more suitable for privacy preservation. We crop the left and right cheeks first and then downsize the one with more height through interpolation such that the height of the two cheek regions becomes the same. After the resizing, we concatenate the cheek regions horizontally to form the whole cheek region. Next, we crop the forehead region and concatenate it (following resizing using interpolation) vertically with the cheek region to form the final ROI. Accordingly, we obtain frames  $f_{ROI}^1, f_{ROI}^2, \dots, f_{ROI}^t$ , which we then resize to a consistent size of  $H \times W$  pixels, where we set  $H$  and  $W$  to 64.

Next, we apply a windowing operation with a window length of  $T = 128$ , and a stride length of 8, to obtain smaller video clips which will be used for training and testing the approach. The same operation is applied to the corresponding PPG signals,  $P$ , to obtain  $(V, S)$  training samples, where  $V$  are the input clips and  $S$  are the output corresponding PPG signals.  $V$  comprises frames  $f_{ROI}^1, f_{ROI}^2, \dots, f_{ROI}^{128}$ , where  $f_{ROI}^i \in \mathbb{R}^{64 \times 64 \times 3}$ . We then flatten each frame into a  $4096 \times 3$  array comprising ordered pixels. Next, we shuffle the ordering of these pixels randomly and reshape the array

to obtain  $f_{RoI+Sh}^i$  with dimensions  $64 \times 64 \times 3$ . Next, we blur  $f_{RoI+Sh}^i$  by convolving  $f_{RoI+Sh}^i$  with a  $3 \times 3$  Gaussian kernel to obtain  $f_{RoI+Sh+B}^i$ . We perform this as we hypothesize that blurring helps in smoothing the shuffled pixels for relatively easier processing for rPPG extraction, while also causing loss of information for recovery of the original image, thereby providing a two-fold advantage for our use-case. We ensure that a set random order (key) is used for each particular sample while shuffling to maintain the pixel location coherence across all the frames of that particular sample. The process of RoI extraction from the face followed by shuffling and blurring comprises  $Q$  (our privacy-preserving transform) and illustrated in Figure 4.1. Here, we note that the search space of the key is  $4096!$  which results in the strong security of our method.

#### 4.1.3 rPPG Estimation Backbone

Finally, we train a deep learning model to estimate the rPPG signal  $S_r$  such that  $S_r \approx S$  from the shuffled and blurred RoI clips. To this end, we use a 3D CNN as our model  $M$ , with learnable weights  $\theta$ . This is the same encoder used in Section 3.1.3 where the model consists of 4 distinct convolutional blocks. The first block uses  $1 \times 5 \times 5$  convolutional filters that extract spatial information from each frame of the video clips. The following three blocks use  $3 \times 3 \times 3$  filters. Each convolution layer is followed by a ReLU activation and batch normalization. The detailed architecture of the model was presented earlier in Table 3.1.

#### 4.1.4 Loss Function

To optimize  $\theta$ , we use the smooth L1 loss [150]. By combining the L1 and L2 losses, this loss enables switching between the two depending on the disparity between the

amplitude values of the estimated rPPG signal  $S_r$ , and the ground-truth PPG signal  $S$  allowing for smoother and more effective learning of weights. The loss is given by:

$$\mathcal{L}(S_r, S) = \begin{cases} \frac{1}{2} \frac{(S_r - S)^2}{\beta}, & |S_r - S| < \beta \\ |S_r - S| - \frac{1}{2} * \beta, & \text{otherwise.} \end{cases} \quad (4.1)$$

where  $\beta$  is a hyperparameter set to 0.3 for our method.

## 4.2 Experiment Setup

### 4.2.1 Datasets

#### rPPG

To test our method in terms of performance toward rPPG estimation, we experiment with two rPPG datasets, the descriptions of which are given below.

**PURE [28]**. This dataset comprises 60 facial videos and their corresponding PPG signals. There are a total of 10 subjects with each subject contributing 6 videos performing different movements such as steady sitting, talking, face rotation, and others. The PPG signals were collected using a Pulox CMS50E finger pulse oximeter at a sampling rate of 60 Hz while the videos were recorded using an Eco274CVGE camera at 30 fps. The videos have been saved in PNG format with a  $640 \times 480$  pixel resolution using lossless compression.

**UBFC [29]**. This dataset comprises 42 facial videos from 42 subjects while they were playing a time-sensitive mathematical game. The videos have been recorded with a Logitech C920 HD Pro webcam at 30 fps while the PPG signals have been recorded using a finger pulse oximeter Pulox CMS50E at 60 Hz sampling rate. The videos have

Table 4.1: Overview of the facial recognition datasets.

Dataset	Identities	Total images
CASIA-Webface [161]	10,575	494,414
LFW [36]	5,749	13,233
CALFW [37]	5,749	12,174
AgeDB [38]	568	16,488

been stored in uncompressed 8-bit format with a resolution of  $640 \times 480$  pixels.

### Facial Recognition

To evaluate the ability of our method in reducing identification capability, we utilize a facial recognition system for benchmarking purposes. To this end, we also use four additional publicly available datasets **CASIA-Webface** [161], **LFW** [36], **CALFW** [37], and **AgeDB** [38] for facial recognition experiments which are summarized in Table 4.1.

#### 4.2.2 Evaluation Scheme and Metrics

**rPPG.** For PURE, we use a train-test split of 6-4 subjects, while for UBFC, we use a 30-12 subject train-test split as done in previous works [25, 59]. Upon estimation of rPPG signals, HR is calculated using the Welch power spectrum method, similar to other works in the area [25, 26]. We then take the average of the HR values to obtain an average HR for each test video and then compare it with the average ground-truth HR values to measure MAE, RMSE both in bpm along with  $R$ .

**Facial Identification.** We test the identifiability of the final perturbed images from both sets of datasets (rPPG datasets and facial recognition datasets). While the facial recognition datasets come with standard train-test protocols, this is not the case for the rPPG datasets. As a result, for PURE and UBFC, we randomly select



Figure 4.2: Sample pairs of images belonging to the same subject from CALFW dataset showing varying imaging conditions.

1000 facial frames for each subject from the dataset and apply the data perturbation scheme (RoI extraction, shuffling, and blurring) to generate the final perturbed images. Next, we use ArcFace [120], a widely used face recognition algorithm to generate 512-dimensional embeddings of the images. We use PCA to transform the high-dimensional embeddings into lower-dimensional embeddings of 32 dimensions for easier classification. We then perform a 5-fold cross-validation using a Support Vector Machine with a radial basis kernel. The final classification/identification accuracy (ID) serves as a utility measure for our privacy-preserving approach.

### 4.2.3 Training

For training the rPPG estimation model, we use a batch size of 8. We train the network for 15 epochs with  $5e-4$  as the learning rate for PURE, and  $2e-4$  for UBFC with Adam [153] optimizer. For ArcFace, we train the model on CASIA-Webface for 25 epochs with a batch size of 180, and an initial learning rate of  $1e-1$  which is divided by 10 at 11, and 16 epochs. We use the SGD optimizer with a momentum of 0.9

and weight decay of  $5e-4$ . All the codes were written in PyTorch [154] and run on an NVIDIA Quadro RTX 8000 GPU.

### 4.3 Results and Discussions

In this section we present our results. First, we present the results of our proposed method, along with an analysis of the different steps associated with it namely, the extraction of RoI, the shuffling of the pixels, and the blurring step. Thereafter, with all things being the same, we study the effect of grouping the pixels as patches. Next, we compare the results of our proposed method with other privacy-preserving methods, followed by a reconstruction attempt to test the security offered by our method. Finally, we present the impact of our method on public facial recognition datasets.

#### 4.3.1 Results on Pixel-wise Shuffling

In Tables 4.2, and 4.3, we present the results of our experiments across all the metrics of rPPG estimation and facial identification. In the first part of the tables, we observe that metrics for rPPG estimation (MAE, RMSE,  $R$ ) improve when we use the cropped RoI as input for rPPG extraction, in comparison to the full face. We then compare the results of varying the number of choices for shuffling keys. We select the key from a choice of 1, 10, 100, and 1000 different possibilities and ultimately remove the bound of key choices (represented as  $U$ ), thereby randomizing the key selection to the entire available key space. We observe that introducing our data perturbation method causes some degradation across MAE and RMSE compared to the RoI-only setting. However, the performance remains within acceptable limits, especially for the  $U$ -setting. In terms of identification, we observe that there is only a minimal

Table 4.2: Comparison of various parameters for our proposed method on PURE.

Input	Keys	MAE↓	RMSE↓	R↑	ID ↓
Face	-	0.65	0.95	0.99	99.35
RoI	-	<b>0.49</b>	<b>0.78</b>	0.99	98.99
RoI+Sh	U	1.23	1.71	0.99	60.20
RoI+Sh+B	1	1.23	1.68	0.99	98.79
RoI+Sh+B	10	1.08	1.46	0.99	90.67
RoI+Sh+B	100	1.69	2.84	0.98	59.96
RoI+Sh+B	1000	1.20	1.98	0.99	46.81
RoI+Sh+B	U	0.96	1.30	0.99	<b>46.19</b>
RoI+Sh <sub>2×2</sub> +B	1	1.65	2.47	0.98	98.02
RoI+Sh <sub>2×2</sub> +B	10	0.83	1.17	0.99	81.21
RoI+Sh <sub>2×2</sub> +B	100	1.11	1.63	0.99	48.98
RoI+Sh <sub>2×2</sub> +B	1000	0.83	1.29	0.99	44.36
RoI+Sh <sub>2×2</sub> +B	U	<b>0.79</b>	<b>1.11</b>	0.99	<b>43.57</b>
RoI+Sh <sub>4×4</sub> +B	1	1.45	2.35	0.99	97.91
RoI+Sh <sub>4×4</sub> +B	10	<b>0.65</b>	0.78	0.99	83.39
RoI+Sh <sub>4×4</sub> +B	100	0.81	1.03	0.99	48.64
RoI+Sh <sub>4×4</sub> +B	1000	0.69	<b>0.72</b>	0.99	38.71
RoI+Sh <sub>4×4</sub> +B	U	0.69	1.04	0.99	<b>36.98</b>
RoI+Sh <sub>8×8</sub> +B	1	1.18	2.32	0.98	97.50
RoI+Sh <sub>8×8</sub> +B	10	0.74	0.95	0.99	81.32
RoI+Sh <sub>8×8</sub> +B	100	0.80	1.15	0.99	45.67
RoI+Sh <sub>8×8</sub> +B	1000	<b>0.69</b>	<b>0.72</b>	0.99	32.67
RoI+Sh <sub>8×8</sub> +B	U	0.70	0.87	0.99	<b>32.55</b>

reduction in accuracy when the RoI is used instead of the face. Moreover, there is little to no change in the accuracy when we introduce the shuffling and blurring step in the single key setting. As we increase the key choices, we observe some reduction in accuracy when the key choices are increased from 1 to 10, but a significant reduction in the accuracies upon increasing the choices from 10 to 100 and subsequently 1000, with the lowest accuracy being achieved in the  $U$ -setting. To better visualize the relation between the HRs calculated from the rPPG estimated from our perturbed face representations ( $U$ -setting), and the ground-truth PPG, we explore the correlation and the Bland-Altman [156] plots in Figures 4.3 and 4.4. Further, we visualize the

Table 4.3: Comparison of various parameters for our proposed method on UBFC.

Input	Keys	MAE↓	RMSE↓	R↑	ID ↓
Face	-	0.52	0.76	0.99	99.97
RoI	-	<b>0.44</b>	<b>0.65</b>	0.99	99.93
RoI+Sh	U	0.79	1.14	0.99	43.49
RoI+Sh+B	1	0.61	0.71	0.99	99.86
RoI+Sh+B	10	0.91	1.34	0.99	98.49
RoI+Sh+B	100	0.88	1.25	0.99	78.31
RoI+Sh+B	1000	1.01	1.52	0.99	42.08
RoI+Sh+B	U	0.89	1.24	0.99	<b>36.14</b>
RoI+Sh <sub>2×2</sub> +B	1	0.66	0.74	0.99	99.79
RoI+Sh <sub>2×2</sub> +B	10	0.99	1.44	0.99	96.25
RoI+Sh <sub>2×2</sub> +B	100	0.65	0.81	0.99	55.53
RoI+Sh <sub>2×2</sub> +B	1000	0.79	1.05	0.99	34.63
RoI+Sh <sub>2×2</sub> +B	U	<b>0.53</b>	<b>0.69</b>	0.99	<b>30.80</b>
RoI+Sh <sub>4×4</sub> +B	1	0.96	1.36	0.99	99.81
RoI+Sh <sub>4×4</sub> +B	10	<b>0.66</b>	<b>0.89</b>	0.99	97.54
RoI+Sh <sub>4×4</sub> +B	100	1.03	1.61	0.98	68.02
RoI+Sh <sub>4×4</sub> +B	1000	0.72	1.02	0.99	33.58
RoI+Sh <sub>4×4</sub> +B	U	0.69	1.09	0.99	<b>27.17</b>
RoI+Sh <sub>8×8</sub> +B	1	1.04	1.74	0.98	99.75
RoI+Sh <sub>8×8</sub> +B	10	0.77	1.19	0.99	97.32
RoI+Sh <sub>8×8</sub> +B	100	0.81	1.28	0.99	70.78
RoI+Sh <sub>8×8</sub> +B	1000	0.63	<b>0.79</b>	0.99	29.10
RoI+Sh <sub>8×8</sub> +B	U	<b>0.61</b>	0.85	0.99	<b>20.83</b>

rPPG signals estimated from the perturbed face representations in Figure 4.5. As seen, our results agree well with the ground-truth for both the datasets, demonstrating the effectiveness of our proposed perturbations in preserving rPPG features.

The results of this experiment indicate that our approach significantly reduces the facial recognition performance, hence preserving privacy, while only resulting in a minor drop in performance for rPPG estimation. We also visualize the embeddings of ArcFace generated for the faces and the perturbed faces by our method in Figure 4.6. We clearly observe that our method disrupts the easily identifiable clusters of different subject classes. Finally, we experiment with the effect of blurring. First, as shown in

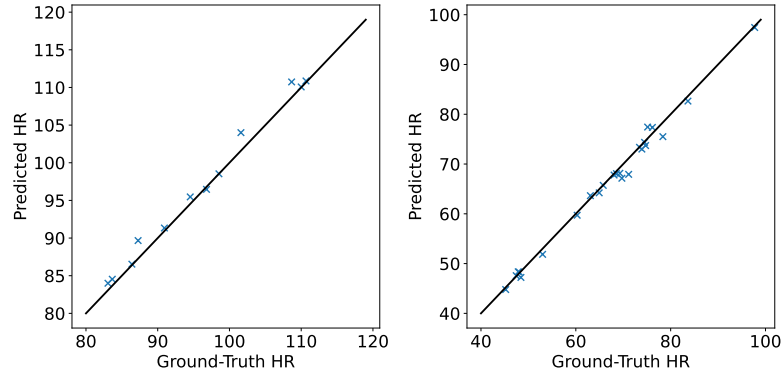


Figure 4.3: Correlation plots for UBFC (left), and PURE (right).

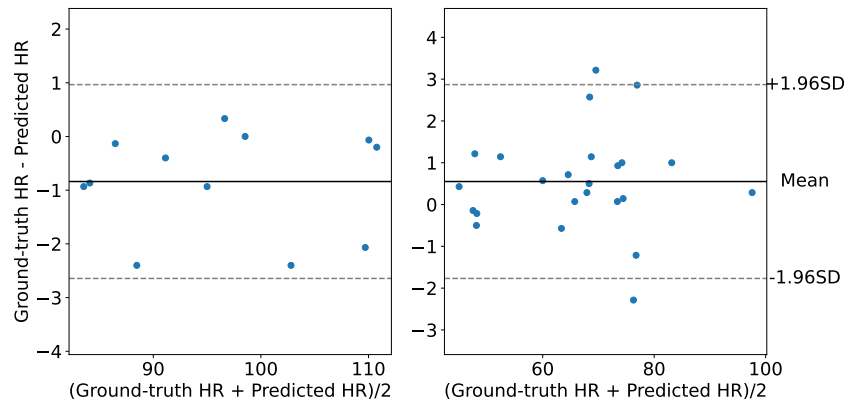


Figure 4.4: Bland-Altman plots for UBFC (left), and PURE (right).

the table, by comparing the performance of shuffling alone to shuffling plus blurring, we observe that in PURE, blurring has a positive impact on rPPG estimation as well as a reduction in ID, while in UBFC, there is a slight degradation in rPPG estimation, while still contributing to the reduction of identification capability. Moreover, when experimenting with different blurring window sizes, Figure 4.7 shows that  $3 \times 3$  is the optimal filter size for reducing ID.

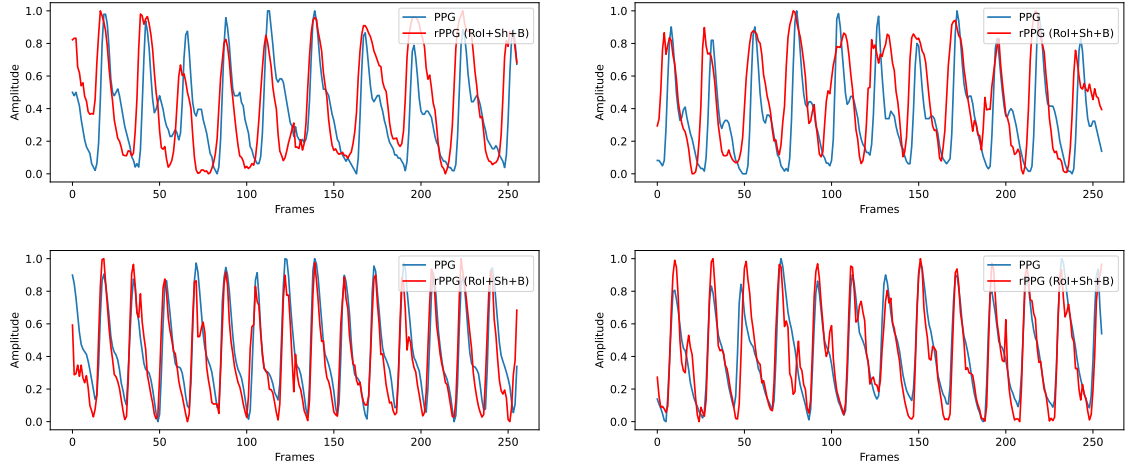


Figure 4.5: Visualization of predicted rPPG for PURE (top), and UBFC (bottom).

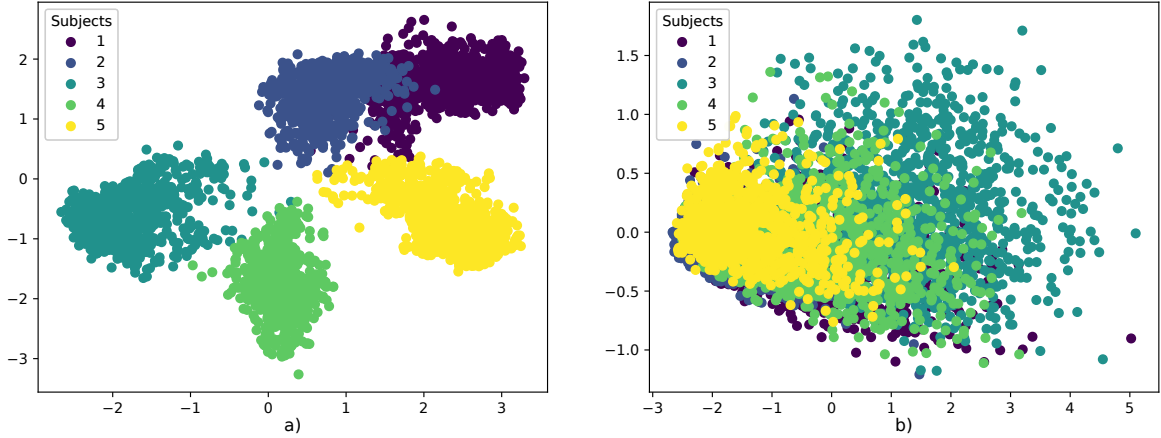


Figure 4.6: Visualization of ArcFace embeddings reduced to 2 dimensions with PCA for a) Face and b) RoI+Sh+B for 5 subjects in PURE.

### 4.3.2 Impact of Patch-wise Shuffling

In the next parts of Tables 4.2, and 4.3, we also compare the effect of shuffling patches instead of pixels, wherein we first divide the frame  $f_{RoI}^i$  into non-overlapping patches of size  $P \times P$ , flatten the image to obtain arrays of dimension  $\frac{64^2}{P^2} \times P \times P \times 3$ , and then shuffle the patches. Next, we reshape the outcome to obtain  $f_{RoI+Sh_{P \times P}}^i \in \mathbb{R}^{64 \times 64 \times 3}$  and blur it. We observe a similar trend for the rPPG and identification metrics

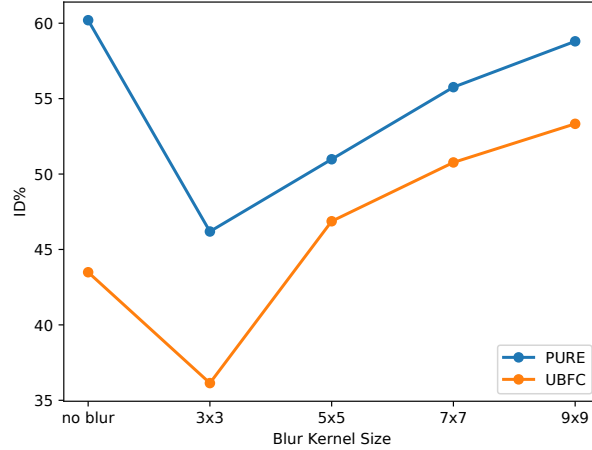


Figure 4.7: The impact of different kernel sizes used for blurring, on ID for both PURE and UBFC datasets.

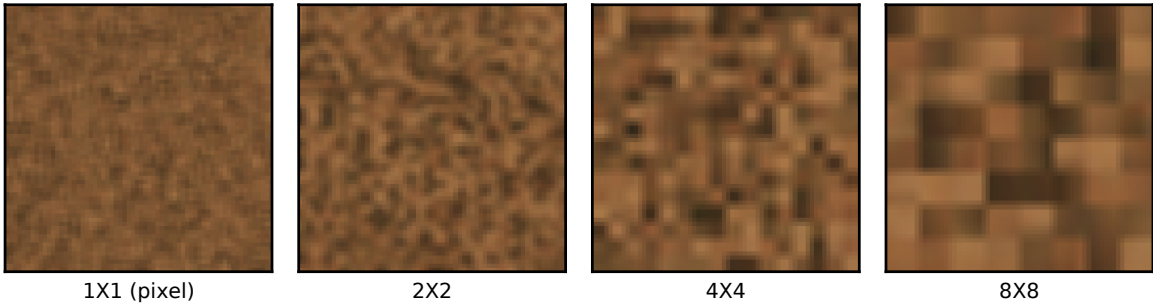


Figure 4.8: Visualization of pixel and patch-wise perturbed images.

as observed in pixel-level shuffle settings where the rPPG metrics remained within acceptable limits as compared to the RoI-only setting, while ID dropped considerably as the choice of shuffling keys were increased. However, we also observe that generally, the rPPG metrics improve and ID further drops upon increasing the patch size. While this may suggest that rather than pixel-level shuffling, we should favour patch-wise shuffling, it is important to note that the grouping of pixels as patches greatly reduces the key space from  $(64)^{2!}$  to  $(64/P)^{2!}$  where  $P \in \{2, 4, 8\}$ . Moreover, since patches

Table 4.4: Comparison with other privacy-preserving methods on rPPG estimation.

Method	PURE			UBFC		
	MAE↓	RMSE↓	R↑	MAE↓	RMSE↓	R↑
No perturbation	0.65	0.95	0.99	0.52	0.76	0.99
BDCT [139]	14.44	15.50	0.10	13.54	16.50	0.03
Noise [137]	9.25	10.75	0.60	8.98	10.82	0.34
LE [135]	9.15	11.60	0.44	5.89	9.14	0.67
InstaHide [134]	2.33	3.02	0.97	2.77	4.02	0.92
Ours	<b>0.96</b>	<b>1.30</b>	<b>0.99</b>	<b>0.89</b>	<b>1.24</b>	<b>0.99</b>

might be susceptible to jigsaw-puzzle solver attacks [162, 163], we believe that pixel-level shuffling ( $U$ -setting) may be a more secure approach. We visualize the effect of increasing patch sizes in our method in Figure 4.8. We can clearly observe that using patches can provide more information to the attacker and reduce the shuffle-key space.

### 4.3.3 Comparison with Other Methods

In Table 4.4 we compare our method with several privacy-preserving methods [139], [137], [135], and [134] as described in Section 2.4. First, we implement the Fast Face Image Masking utilizing BDCT proposed in [139] to encode the facial images before feeding them into our rPPG estimator. Next, we use Gaussian noise with a deviation of  $\sigma^2 = 0.5$  similar to the baseline used in [137]. We also use the encryption and adaptation strategy proposed in [135] to modify the input face images. For these three methods, we use the same data perturbation for both training and inference. And finally, we implement InstaHide [134] with  $k = 2$ . It is important to note that for InstaHide, during training, both the input as well as the output samples are mixed. However, since we do not want mixed/inseparable rPPG signals, we follow [134] and do not apply InstaHide during the inference process. The comparison results in Table 4.4 demonstrate that our method results in better maintenance of rPPG features as

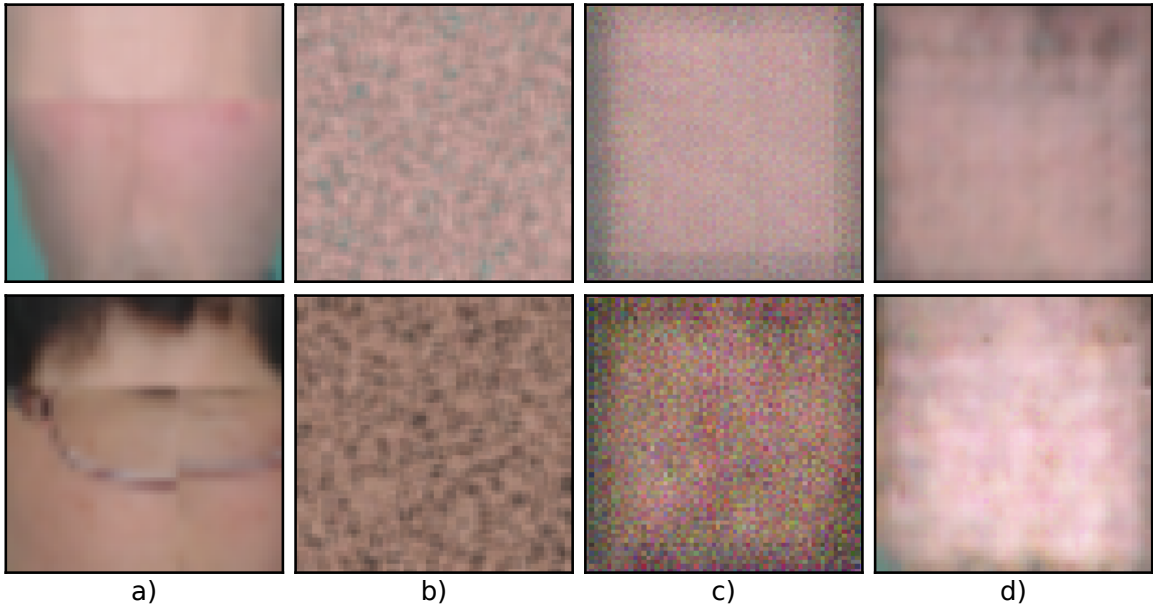


Figure 4.9: Reconstruction attempts. a) RoI, b) RoI+Sh+B, c) Reconstructed RoI from UNet, and d) Reconstructed RoI from Pix2Pix.

the other privacy-preserving techniques result in significant drops in rPPG estimation performance.

#### 4.3.4 Reconstruction Attempt

To test the reversibility of the shuffling and blurring operation of our data perturbation scheme, we follow similar approaches to [139, 140] and implement a UNet [164] (batch size=512, learning rate=1e-1, epochs=20), and Pix2Pix GAN [165] (learning rate=2e-4, epochs=200) to try to learn a mapping between RoI and RoI+Sh+B images. We train the networks with pixel-shuffled and blurred RoI images as inputs and the original RoI images as outputs. In Figure 4.9, we observe that recovering the RoI images is very difficult, further supporting our method.

Table 4.5: Results of using our method on facial recognition datasets in terms of verification accuracy %.

Testing Input	Keys	LFW	CALFW	AgeDB
No perturbation	-	99.35	93.03	93.53
RoI	-	65.78	55.65	51.78
RoI+Sh	U	58.30	52.28	<b>49.91</b>
RoI+Sh+B	1	57.25	53.33	50.11
RoI+Sh+B	10	<b>53.76</b>	52.58	50.93
RoI+Sh+B	100	55.10	52.58	50.53
RoI+Sh+B	1000	55.38	51.41	50.48
RoI+Sh+B	U	54.16	<b>51.36</b>	50.05

#### 4.3.5 Impact on Public Benchmarks

To further showcase the effect of our proposed data perturbation pipeline, we evaluate it on large-scale facial recognition tasks for which we use the ArcFace model trained on CASIA-Webface as described in Section 4.2.3 and test it on LFW, CALFW, and AgeDB. The original datasets contain images of faces across varying poses, lighting, background, and even age. We test the ArcFace on the given face images, the extracted RoI, and the perturbed images. In Table 4.5, we observe that using RoI instead of the face greatly reduces the verification accuracy, which further reduces upon the introduction of shuffling and blurring as part of our data perturbation scheme, and with increased randomness based on the key parameter for all the datasets.

## Chapter 5

### Conclusion and Future Work

#### 5.1 Conclusion

In this thesis, we addressed two major problems in the field of rPPG, and hence remote HR estimation, using deep learning. The first problem is the reliance of the methods on large amounts of labeled data for effective training. The second problem is the concern of privacy which arises by the use of facial videos to estimate rPPG. To address the problem of reliance on labeled data, we proposed an effective solution based on self-supervised contrastive pre-training and subsequent fine-tuning. To tackle the concern of privacy, we proposed a plug-and-play data perturbation scheme to extract a privacy-preserving facial representation from face images that causes minimal degradation in rPPG estimation. Following is a summary of our work in this thesis.

In **Chapter 3**, we proposed a two-stage method based on the use of self-supervised contrastive pre-training and fine-tuning for rPPG estimation and HR prediction from facial videos. In the first stage, a sample video clip of RoI, and its augmented counterpart were fed into an encoder to generate high-dimensional embeddings, which were then projected onto a lower dimension through a projection head. The contrastive

loss was used to maximize the similarity between the embeddings of the same original sample and minimize the similarity between those of different samples. Next, we discarded the projection head and fine-tuned the encoder for rPPG estimation. We showcased that introducing contrastive learning as a pre-training measure helps in more robust learning of the network for the downstream task of rPPG estimation. Our comprehensive experiments showed that our self-supervised approach outperforms many fully supervised techniques to approach the state-of-the-art, while also being less reliant on output labels during the training stage.

In **Chapter 4**, we proposed the detection of selected facial regions followed by pixel-shuffling and blurring as a means to conceal the identity of the subjects for rPPG extraction. Through comprehensive experiments, we validated the effect of different parameters of our proposed method on both the rPPG estimation as well as facial identification. We demonstrated that the introduction of our data perturbation scheme causes minimal trade-off in the performance of rPPG estimation from the new facial representation while causing major degradation in facial identification. Our comparison with other privacy-preserving methods showed that our proposed method is more suited for the task of rPPG extraction. We also illustrated that our data perturbation scheme is robust to reconstruction/reversing attempts, thereby providing strong security to the perturbed face images. Lastly, we also showcased that our method causes a significant reduction in the performance of facial recognition systems when used for testing on public datasets.

## 5.2 Future Work

For future work, our two approaches can be combined together to create an end-to-end privacy-preserving self-supervised learning pipeline. However, an interesting challenge to explore and study is the interaction between self-supervised augmentations and the perturbations required for privacy preservation. Additionally, different use cases of the obtained rPPG signals can be explored beyond HR estimation, for instance toward estimation of arousal, valence, stress, and cognitive load [5, 4, 166] to contribute to an all-encompassing system for deployment in smart environments.

Another orientation to further expand our work would be towards mitigating the bias and ensuring fairness in rPPG estimation. Since rPPG relies on the light intensities reflected from the surface of the skin, it is sensitive to a number of factors such as age, skin tones, makeup, cultural artifacts, and others [167, 168, 169] which differ across varying demographics. To this end, the proposed methods can be expanded to explicitly target such problems. Also, most of the datasets used in this context are collected in controlled settings, which although have some degrees of changes in illumination, pose, expressions, skin tones, among other deciding factors, do not represent the entirety of real-world scenarios. Accordingly, new datasets can be collected, which better represent and resembles real-life and in-the-wild scenarios. Also, given the newfound attention of the community towards generative artificial intelligence, methods such as GANs and diffusion models could be leveraged to synthesize facial rPPG data targeting all the above-mentioned issues and design algorithms on the same.

## Bibliography

- [1] K. Shelley, S. Shelley, and C. Lake, “Pulse oximeter waveform: photoelectric plethysmography,” *Clinical Monitoring*, vol. 2, 2001.
- [2] A. R. Kavsaoglu, K. Polat, and M. Hariharan, “Non-invasive prediction of hemoglobin level using machine learning techniques with the ppg signal’s characteristics features,” *Applied Soft Computing*, vol. 37, pp. 983–991, 2015.
- [3] P. Dehkordi, A. Garde, B. Molavi, J. M. Ansermino, and G. A. Dumont, “Extracting instantaneous respiratory rate from multiple photoplethysmogram respiratory-induced variations,” *Frontiers in Physiology*, vol. 9, p. 948, 2018.
- [4] M. S. Lee, Y. K. Lee, D. S. Pae, M. T. Lim, D. W. Kim, and T. K. Kang, “Fast emotion recognition based on single pulse ppg signal with convolutional neural network,” *Applied Sciences*, vol. 9, no. 16, p. 3355, 2019.
- [5] F. Gasparini, A. Grossi, and S. Bandini, “A deep learning approach to recognize cognitive load using ppg signals,” *PERvasive Technologies Related to Assistive Environments Conference*, pp. 489–495, 2021.
- [6] F. P. Wieringa, F. Mastik, and A. F. van der Steen, “Contactless multiple wavelength photoplethysmographic imaging: A first step toward “spo 2 camera”

- technology,” *Annals of Biomedical Engineering*, vol. 33, no. 8, pp. 1034–1041, 2005.
- [7] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, “Internet of things (iot) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5g-iot scenarios,” *IEEE Access*, vol. 8, pp. 23 022–23 040, 2020.
- [8] S. Bhowmick, P. K. Kundu, and D. D. Mandal, “Iot assisted real time ppg monitoring system for health care application,” *IEEE International Conference on Control, Measurement and Instrumentation*, pp. 122–127, 2021.
- [9] R. K. Nath and H. Thapliyal, “Ppg based continuous blood pressure monitoring framework for smart home environment,” *IEEE World Forum on Internet of Things*, pp. 1–6, 2020.
- [10] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [11] B. Hammi, S. Zeadally, R. Khatoun, and J. Nebhen, “Survey on smart homes: Vulnerabilities, risks, and countermeasures,” *Computers & Security*, vol. 117, p. 102677, 2022.
- [12] J. Al-Dulaimi, J. Cosmas, and M. Abbod, “Smart health and safety equipment monitoring system for distributed workplaces,” *Computers*, vol. 8, no. 4, p. 82, 2019.

- [13] S. Majumder, E. Aghayi, M. Noferesti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. J. Deen, "Smart homes for elderly healthcare—recent advances and research challenges," *Sensors*, vol. 17, no. 11, p. 2496, 2017.
- [14] M. Jones, F. DeRuyter, and J. Morris, "The digital health revolution and people with disabilities: perspective from the united states," *International Journal of Environmental Research and Public Health*, vol. 17, no. 2, p. 381, 2020.
- [15] J. Wang, J. M. Warnecke, M. Haghi, and T. M. Deserno, "Unobtrusive health monitoring in private spaces: The smart vehicle," *Sensors*, vol. 20, no. 9, p. 2442, 2020.
- [16] A. Kumar, A. Sharma, V. Bharti, A. K. Singh, S. K. Singh, and S. Saxena, "Mobihisnet: a lightweight cnn in mobile edge computing for histopathological image classification," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 778–17 789, 2021.
- [17] D. K. Dewangan and S. P. Sahu, "Deep learning-based speed bump detection model for intelligent vehicle system using raspberry pi," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3570–3578, 2020.
- [18] J.-h. Park, M. M. Salim, J. H. Jo, J. C. S. Sicato, S. Rathore, and J. H. Park, "Ciot-net: a scalable cognitive iot based smart city network architecture," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–20, 2019.

- 
- [19] A. C. de Araujo and A. Etemad, “End-to-end prediction of parcel delivery time with deep learning for smart-city applications,” *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 17 043–17 056, 2021.
- [20] M. Burhan, R. A. Rehman, B. Khan, and B.-S. Kim, “Iot elements, layered architectures and security issues: A comprehensive survey,” *Sensors*, vol. 18, no. 9, p. 2796, 2018.
- [21] S. Elzeiny and M. Qaraqe, “Blueprint to workplace stress detection approaches,” *International Conference on Computer and Applications*, pp. 407–412, 2018.
- [22] S. Zepf, J. Hernandez, A. Schmitt, W. Minker, and R. W. Picard, “Driver emotion recognition for intelligent vehicles: A survey,” *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–30, 2020.
- [23] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, “A survey on federated learning: The journey from centralized to distributed on-site learning and beyond,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2020.
- [24] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, “Security and privacy in smart city applications: Challenges and solutions,” *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.
- [25] R. Špetlík, V. Franc, and J. Matas, “Visual heart rate estimation with convolutional neural network,” *British Machine Vision Conference*, pp. 3–6, 2018.

- [26] S.-Q. Liu and P. C. Yuen, “A general remote photoplethysmography estimator with spatiotemporal convolutional network,” *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 481–488, 2020.
- [27] W. Verkrusse, L. O. Svaasand, and J. S. Nelson, “Remote plethysmographic imaging using ambient light.” *Optics Express*, vol. 16, no. 26, pp. 21 434–21 445, 2008.
- [28] R. Stricker, S. Müller, and H.-M. Gross, “Non-contact video-based pulse rate measurement on a mobile service robot,” *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 1056–1062, 2014.
- [29] S. Bobbia, R. Macwan, Y. Benezeth, A. Mansouri, and J. Dubois, “Unsupervised skin tissue segmentation for remote photoplethysmography,” *Pattern Recognition Letters*, vol. 124, pp. 82–90, 2019.
- [30] G. Heusch, A. Anjos, and S. Marcel, “A reproducible study on remote heart rate measurement,” *arXiv preprint arXiv:1709.00962*, 2017.
- [31] M. Wang and W. Deng, “Deep face recognition: A survey,” *Neurocomputing*, vol. 429, pp. 215–244, 2021.
- [32] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, “Cafe: Catastrophic data leakage in vertical federated learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 994–1006, 2021.
- [33] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos, “Privacy and security issues in deep learning: A survey,” *IEEE Access*, vol. 9, pp. 4566–4593, 2020.

- [34] G. De Haan and V. Jeanne, “Robust pulse rate from chrominance-based rppg,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2878–2886, 2013.
- [35] W. Wang, A. C. den Brinker, S. Stuijk, and G. De Haan, “Algorithmic principles of remote ppg,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1479–1491, 2016.
- [36] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” *Workshop on faces in ‘Real-Life’ Images: detection, alignment, and recognition*, 2008.
- [37] T. Zheng, W. Deng, and J. Hu, “Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments,” *arXiv preprint arXiv:1708.08197*, 2017.
- [38] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, “Agedb: the first manually collected, in-the-wild age database,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 51–59, 2017.
- [39] D. Gupta and A. Etemad, “Self-supervised remote monitoring of heart rate from videos,” *AAAI Workshop on Human-Centric Self-Supervised Learning*, 2022.
- [40] X. Li, J. Chen, G. Zhao, and M. Pietikainen, “Remote heart rate measurement from face videos under realistic situations,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4264–4271, 2014.

- [41] M.-Z. Poh, D. J. McDuff, and R. W. Picard, "Advancements in noncontact, multiparameter physiological measurements using a webcam," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 7–11, 2010.
- [42] W. Wang, S. Stuijk, and G. De Haan, "A novel algorithm for remote photoplethysmography: Spatial subspace rotation," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 9, pp. 1974–1984, 2015.
- [43] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, 2012.
- [44] Z. Yu, X. Li, and G. Zhao, "Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks," *arXiv preprint arXiv:1905.02419*, 2019.
- [45] M. Hu, F. Qian, X. Wang, L. He, D. Guo, and F. Ren, "Robust heart rate estimation with spatial-temporal attention network from facial videos," *IEEE Transactions on Cognitive and Developmental Systems*, 2021.
- [46] M. Hu, F. Qian, D. Guo, X. Wang, L. He, and F. Ren, "Eta-rppgnet: Effective time-domain attention network for remote heart rate measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.
- [47] P. Zhang, B. Li, J. Peng, and W. Jiang, "Multi-hierarchical convolutional network for efficient remote photoplethysmograph signal and heart rate estimation from face video clips," *arXiv preprint arXiv:2104.02260*, 2021.

- [48] M. Hu, D. Guo, X. Wang, P. Ge, and Q. Chu, "A novel spatial-temporal convolutional neural network for remote photoplethysmography," *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, pp. 1–6, 2019.
- [49] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [50] W. Chen and D. McDuff, "Deepphys: Video-based physiological measurement using convolutional attention networks," *European Conference on Computer Vision*, pp. 349–365, 2018.
- [51] Y. Zhao, B. Zou, F. Yang, L. Lu, A. N. Belkacem, and C. Chen, "Video-based physiological measurement using 3d central difference convolution attention network," *IEEE International Joint Conference on Biometrics*, pp. 1–6, 2021.
- [52] Z. Yu, B. Zhou, J. Wan, P. Wang, H. Chen, X. Liu, S. Z. Li, and G. Zhao, "Searching multi-rate and multi-modal temporal enhanced networks for gesture recognition," *IEEE Transactions on Image Processing*, 2021.
- [53] Y. Ren, B. Syrnyk, and N. Avadhanam, "Dual attention network for heart rate and respiratory rate estimation," *IEEE International Workshop on Multimedia Signal Processing*, pp. 1–6, 2021.
- [54] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," *IEEE International Conference on Computer Vision*, pp. 7083–7093, 2019.

- [55] X. Liu, W. Wei, H. Kuang, and X. Ma, "Heart rate measurement based on 3d central difference convolution with attention mechanism," *Sensors*, vol. 22, no. 2, p. 688, 2022.
- [56] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," *European Conference on Computer Vision*, pp. 3–19, 2018.
- [57] J. Li, Z. Yu, and J. Shi, "Learning motion-robust remote photoplethysmography through arbitrary resolution videos," *arXiv preprint arXiv:2211.16922*, 2022.
- [58] M. Hu, D. Guo, M. Jiang, F. Qian, X. Wang, and F. Ren, "rppg-based heart rate estimation using spatial-temporal attention network," *IEEE Transactions on Cognitive and Developmental Systems*, 2021.
- [59] R. Song, H. Chen, J. Cheng, C. Li, Y. Liu, and X. Chen, "PulseGAN: Learning to generate realistic pulse waveforms in remote photoplethysmography," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 5, pp. 1373–1384, 2021.
- [60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [61] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [62] H. Wang, Y. Zhou, and A. El Saddik, "Vitasi: A real-time contactless vital signs estimation system," *Computers and Electrical Engineering*, vol. 95, 2021.

- [63] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, “Phase-based video motion processing,” *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–10, 2013.
- [64] X. Niu, S. Shan, H. Han, and X. Chen, “Rhythmnet: End-to-end heart rate estimation from face via spatial-temporal representation,” *IEEE Transactions on Image Processing*, vol. 29, pp. 2409–2423, 2019.
- [65] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.
- [66] E. Magdalena Nowara, T. K. Marks, H. Mansour, and A. Veeraraghavan, “Sparseppg: Towards driver monitoring using camera-based vital signs estimation in near-infrared,” *IEEE IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1272–1281, 2018.
- [67] E. M. Nowara, T. K. Marks, H. Mansour, and A. Veeraraghavan, “Near-infrared imaging photoplethysmography during driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3589–3600, 2020.
- [68] G. Nagamatsu, E. M. Nowara, A. Pai, A. Veeraraghavan, and H. Kawasaki, “Ppg3d: Does 3d head tracking improve camera-based ppg estimation?” *International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1194–1197, 2020.
- [69] D. Alexey, P. Fischer, J. Tobias, M. R. Springenberg, and T. Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks,”

- IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1734–1747, 2016.
- [70] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [71] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” pp. 1422–1430, 2015.
- [72] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1422–1430, 2015.
- [73] N. Komodakis and S. Gidaris, “Unsupervised representation learning by predicting image rotations,” *International Conference on Learning Representations*, 2018.
- [74] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.
- [75] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” *European Conference on Computer Vision*, pp. 649–666, 2016.
- [76] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and learn: unsupervised learning using temporal order verification,” *European Conference on Computer Vision*, pp. 527–544, 2016.

- [77] B. Fernando, H. Bilen, E. Gavves, and S. Gould, “Self-supervised video representation learning with odd-one-out networks,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3636–3645, 2017.
- [78] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, “Self-supervised spatiotemporal learning via video clip order prediction,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10 334–10 343, 2019.
- [79] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, “Learning and using the arrow of time,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8052–8060, 2018.
- [80] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, “Tracking emerges by colorizing videos,” *European Conference on Computer Vision*, pp. 391–408, 2018.
- [81] U. Ahsan, R. Madhok, and I. Essa, “Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition,” *IEEE Winter Conference on Applications of Computer Vision*, pp. 179–189, 2019.
- [82] L. Jing, X. Yang, J. Liu, and Y. Tian, “Self-supervised spatiotemporal feature learning via video rotation prediction,” *arXiv preprint arXiv:1811.11387*, 2018.
- [83] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

- [84] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 16 000–16 009, 2022.
- [85] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [86] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [87] C. Feichtenhofer, Y. Li, K. He *et al.*, “Masked autoencoders as spatiotemporal learners,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 946–35 958, 2022.
- [88] S. R. Taghanaki and A. Etemad, “Self-supervised wearable-based activity recognition by learning to forecast motion,” *arXiv preprint arXiv:2010.13713*, 2020.
- [89] P. Sarkar and A. Etemad, “Self-supervised ecg representation learning for emotion recognition,” *IEEE Transactions on Affective Computing*, 2020.
- [90] P. Sarkar and A. Etemad, “Self-supervised learning for ecg-based emotion recognition,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3217–3221, 2020.

- 
- [91] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, “Fsce: Few-shot object detection via contrastive proposal encoding,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7352–7362, 2021.
- [92] S. Roy and A. Etemad, “Spatiotemporal contrastive learning of facial expressions in videos,” *arXiv preprint arXiv:2108.03064*, 2021.
- [93] S. Roy and A. Etemad, “Self-supervised contrastive learning of multi-view facial expressions,” *arXiv preprint arXiv:2108.06723*, 2021.
- [94] Z. Mahmud, P. Hungler, and A. Etemad, “Gaze estimation with eye region segmentation and self-supervised multistream learning,” *arXiv preprint arXiv:2112.07878*, 2021.
- [95] S. R. Taghanaki, M. Rainbow, and A. Etemad, “Self-supervised human activity recognition with localized time-frequency contrastive representation learning,” *arXiv preprint arXiv:2209.00990*, 2022.
- [96] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” *International Conference on Machine Learning*, pp. 1597–1607, 2020.
- [97] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.
- [98] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, “With a little help from my friends: Nearest-neighbor contrastive learning of visual

- representations,” *IEEE International Conference on Computer Vision*, pp. 9588–9597, 2021.
- [99] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020.
- [100] X. Chen and K. He, “Exploring simple siamese representation learning,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 15 750–15 758, 2021.
- [101] P. Sarkar and A. Etemad, “Self-supervised audio-visual representation learning with relaxed cross-modal synchronicity,” *arXiv preprint arXiv:2111.05329*, 2021.
- [102] S. Soltanieh, J. Hashemi, and A. Etemad, “In-distribution and out-of-distribution self-supervised ecg representation learning for arrhythmia detection,” *arXiv preprint arXiv:2304.06427*, 2023.
- [103] M. D. Kelly, “Visual identification of people by computer,” 1971.
- [104] T. Kanade, “Picture processing by computer complex and recognition of human faces,” 1973.
- [105] L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces,” *Josa a*, vol. 4, no. 3, pp. 519–524, 1987.
- [106] M. Kirby and L. Sirovich, “Application of the karhunen-loeve procedure for the characterization of human faces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, 1990.

- 
- [107] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, “Face recognition using hog–ebgm,” *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1537–1543, 2008.
- [108] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [109] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, “On the use of sift features for face authentication,” *Conference on Computer Vision and Pattern Recognition Workshop*, pp. 35–35, 2006.
- [110] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.
- [111] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” *British Machine Vision Association*, 2015.
- [112] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [113] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [114] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

- 
- [115] R. Ranjan, C. D. Castillo, and R. Chellappa, “L2-constrained softmax loss for discriminative face verification,” *arXiv preprint arXiv:1703.09507*, 2017.
- [116] F. Wang, J. Cheng, W. Liu, and H. Liu, “Additive margin softmax for face verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [117] Y. Zheng, D. K. Pal, and M. Savvides, “Ring loss: Convex feature normalization for face recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5089–5097, 2018.
- [118] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 212–220, 2017.
- [119] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.
- [120] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.
- [121] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, “Circle loss: A unified perspective of pair similarity optimization,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6398–6407, 2020.
- [122] H. Uppal, A. Sepas-Moghaddam, M. Greenspan, and A. Etemad, “Depth as attention for face representation learning,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2461–2476, 2021.

- 
- [123] H. Uppal, A. Sepas-Moghaddam, M. Greenspan, and A. Etemad, "Teacher-student adversarial depth hallucination to improve face recognition," *IEEE International Conference on Computer Vision*, pp. 3671–3680, 2021.
- [124] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, "Capsfield: Light field-based face and expression recognition in the wild using capsule routing," *IEEE Transactions on Image Processing*, vol. 30, pp. 2627–2642, 2021.
- [125] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, "Long short-term memory with gate and state level fusion for light field-based face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1365–1379, 2020.
- [126] V. Pavez, G. Hermosilla, F. Pizarro, S. Fingerhuth, and D. Yunge, "Thermal image generation for robust face recognition," *Applied Sciences*, vol. 12, no. 1, p. 497, 2022.
- [127] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 253–261, 2020.
- [128] R. Panchendraran and S. Bhoi, "Dataset reconstruction attack against language models," *CEUR Workshop*, 2021.
- [129] Z. Zhang, Q. Liu, Z. Huang, H. Wang, C.-K. Lee, and E. Chen, "Model inversion attacks against graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

- [130] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace, “Extracting training data from diffusion models,” *arXiv preprint arXiv:2301.13188*, 2023.
- [131] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A survey on homomorphic encryption schemes: Theory and implementation,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.
- [132] Q. Zhang, C. Xin, and H. Wu, “Privacy-preserving deep learning based on multiparty secure computation: A survey,” *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 412–10 429, 2021.
- [133] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” *ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.
- [134] Y. Huang, Z. Song, K. Li, and S. Arora, “Instahide: Instance-hiding schemes for private distributed learning,” *International Conference on Machine Learning*, pp. 4507–4518, 2020.
- [135] M. Tanaka, “Learnable image encryption,” *IEEE International Conference on Consumer Electronics-Taiwan*, pp. 1–2, 2018.
- [136] K. Madono, M. Tanaka, M. Onishi, and T. Ogawa, “Block-wise scrambled image recognition using adaptation network,” *arXiv preprint arXiv:2001.07761*, 2020.
- [137] Z. Ren, Y. J. Lee, and M. S. Ryoo, “Learning to anonymize faces for privacy preserving action detection,” *European Conference on Computer Vision*, pp. 620–636, 2018.

- [138] Y. Hu, Y. Wang, and J. Zhang, “Dear-gan: Degradation-aware face restoration with gan prior,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [139] Y. Wang, J. Liu, M. Luo, L. Yang, and L. Wang, “Privacy-preserving face recognition in the frequency domain,” *AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, pp. 2558–2566, 2022.
- [140] J. Ji, H. Wang, Y. Huang, J. Wu, X. Xu, S. Ding, S. Zhang, L. Cao, and R. Ji, “Privacy-preserving face recognition with learnable privacy budgets in frequency domain,” *European Conference on Computer Vision*, pp. 475–491, 2022.
- [141] M. A. P. Chamikara, P. Bertok, I. Khalil, D. Liu, and S. Camtepe, “Privacy preserving face recognition utilizing differential privacy,” *Computers & Security*, vol. 97, p. 101951, 2020.
- [142] Y. Li, Q. Lu, Q. Tao, X. Zhao, and Y. Yu, “Sf-gan: face de-identification method without losing facial attribute information,” *IEEE Signal Processing Letters*, vol. 28, pp. 1345–1349, 2021.
- [143] Y. Wu, F. Yang, and H. Ling, “Privacy-protective-gan for face de-identification,” *arXiv preprint arXiv:1806.08906*, 2018.
- [144] D. Datcu, M. Cidota, S. Lukosch, and L. Rothkrantz, “Noncontact automatic heart rate analysis in visible spectrum by specific face regions,” *International Conference on Computer Systems and Technologies*, pp. 120–127, 2013.

- [145] R. Irani, K. Nasrollahi, and T. B. Moeslund, “Improved pulse detection from head motions using dct,” *International Conference on Computer Vision Theory and Applications*, vol. 3, pp. 118–124, 2014.
- [146] D. E. King, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [147] Y. W. Lee and K. R. Park, “Recent iris and ocular recognition methods in high-and low-resolution images: A survey,” *Mathematics*, vol. 10, no. 12, p. 2063, 2022.
- [148] D. P. Chowdhury, R. Kumari, S. Bakshi, M. N. Sahoo, and A. Das, “Lip as biometric and beyond: a survey,” *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 3831–3865, 2022.
- [149] A. Elmahmudi and H. Ugail, “Deep face recognition using imperfect facial data,” *Future Generation Computer Systems*, vol. 99, pp. 213–225, 2019.
- [150] R. Girshick, “Fast r-cnn,” *IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.
- [151] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.
- [152] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, “Self-supervised spatiotemporal learning via video clip order prediction,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10 334–10 343, 2019.

- [153] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [154] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [155] V. Bhaskaran and K. Konstantinides, “Image and video compression standards: algorithms and architectures,” 1997.
- [156] J. M. Bland and D. Altman, “Statistical methods for assessing agreement between two methods of clinical measurement,” *The Lancet*, vol. 327, no. 8476, pp. 307–310, 1986.
- [157] L. Ericsson, H. Gouk, and T. M. Hospedales, “Why do self-supervised models transfer? investigating the impact of invariance on downstream tasks,” *arXiv preprint arXiv:2111.11398*, 2021.
- [158] S. Soltanieh, A. Etemad, and J. Hashemi, “Analysis of augmentations for contrastive ecg representation learning,” *arXiv preprint arXiv:2206.07656*, 2022.
- [159] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [160] S. Kwon, J. Kim, D. Lee, and K. Park, “Roi analysis for remote photoplethysmography on facial video,” *International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4938–4941, 2015.

- [161] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” *arXiv preprint arXiv:1411.7923*, 2014.
- [162] T. Chuman and H. Kiya, “A jigsaw puzzle solver-based attack on block-wise image encryption for privacy-preserving dnns,” *International Workshop on Advanced Imaging Technology*, vol. 12592, pp. 335–340, 2023.
- [163] T. Chuman, W. Sirichotedumrong, and H. Kiya, “Encryption-then-compression systems using grayscale-based image encryption for jpeg images,” *IEEE Transactions on Information Forensics and security*, vol. 14, no. 6, pp. 1515–1525, 2018.
- [164] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.
- [165] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.
- [166] A. Bhatti, B. Behinaein, P. Hungler, and A. Etemad, “Attx: Attentive cross-connections for fusion of wearable signals in emotion recognition,” *arXiv preprint arXiv:2206.04625*, 2022.
- [167] E. M. Nowara, D. McDuff, and A. Veeraraghavan, “A meta-analysis of the impact of skin tone and gender on non-contact photoplethysmography measurements,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 284–285, 2020.

- 
- [168] W. Wang and C. Shan, “Impact of makeup on remote-ppg monitoring,” *Biomedical Physics & Engineering Express*, vol. 6, no. 3, p. 035004, 2020.
- [169] A. Dasari, S. K. A. Prakash, L. A. Jeni, and C. S. Tucker, “Evaluation of biases in remote photoplethysmography methods,” *NPJ digital medicine*, vol. 4, no. 1, p. 91, 2021.