Towards Accessible, Equitable, Generalizable and Useful Camera Health Sensing

Xin Liu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Shwetak N. Patel, Chair

Daniel McDuff

Christopher Altoff

Luis Ceze

Program Authorized to Offer Degree:

Paul G. Allen School of Computer Science & Engineering

University of Washington

Abstract

Towards Accessible, Equitable, Generalizable and Useful
Camera Health Sensing

Xin Liu

Chair of the Supervisory Committee:
Shwetak N. Patel
Paul G. Allen School of Computer Science & Engineering

The COVID-19 pandemic has prompted a shift in the delivery of healthcare globally, with a growing emphasis on scalable health sensing. Currently, biomedical contact sensors are considered the gold standard for measuring vital signals, but they are not widely accessible, particularly in under-resourced areas. Camera-based health sensing offers the potential to reach a wider population by using regular RGB cameras to detect changes in electromagnetic radiation (light) reflected from the body that result from physiological processes. However, existing camera-based health sensing methods are inaccessible due to their high computational costs, inequitable due to poor generalizability across skin tones, lighting, and movements, and not fully validated for use in clinical settings. To address these challenges, this thesis explores the development of on-device neural networks, few-shot adaptation, federated learning, and data augmentation systems and algorithms for camera-based health sensing. A transnational clinical study is also conducted to evaluate the usefulness of these methods in real-world clinical settings and to advance the field of camera-based health sensing beyond well-studied physiological signals. Finally, this research introduces an open-source toolbox to promote reproducibility and fair benchmarking comparisons.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to extend my deepest thanks to my parents and sister, who have supported me tirelessly and made it possible for me to start my undergraduate education in the United States. Their unwavering love and support have played a vital role in shaping who I am today. I would also like to express my heartfelt thanks to my partner, Xinbei, for being there every step of the way during my PhD journey. Your presence has made this journey the best part of my life. Thank you for sharing this journey with me and for bringing joy and comfort during the challenges we faced together.

I would like to express my sincere gratitude to my advisor, Shwetak, who has served as a role model to me in many aspects of my PhD journey, including as a highly-regarded researcher, an effective team leader, a loving husband, and a devoted father. Thank you for invaluable guidance in not only teaching me the skills to conduct impactful research but also instilling in me the ability to balance work, life and the art of multitasking. I am also grateful for the freedom you granted me to pursue the projects I am passionate about, without worrying about funding, and for supporting my desire to share my time between academia and industry throughout my PhD.

To Daniel, thank you for introducing me into the world of camera health sensing and shaping my PhD. Your passion, persistence and encouragement have been instrumental in enabling me to achieve my goals as a multidisciplinary researcher in machine learning, ubiquitous computing and biomedical engineering. I also appreciate that you have been an amazing co-advisor, friend, research collaborator, and colleague at UW, Microsoft, and Google. I sincerely look forward to our continuing collaboration in the future.

To my labmates in the Ubicomp Lab, thanks for creating such a wonderful and supportive

environment in our lab. Your support have made my time here truly memorable. I really enjoyed brainstorming research ideas, talking about career plans and building the new lab space with all of you. Matt, we joined the lab at the same year and both working on deep learning based health sensing. Thank you for being a such supportive labmate and friend. I have no doubt that your startup will reach great heights. Ishan, Joe, Alvin, Jason, Richard, Girish, Matt, Anand, Shirley, I am confident that each of you will have a bright future in research.

I also would like express my gratitude to many lab alumnus who have left a lasting impact on me. Josh, thank you for introducing me to the world of on-device machine learning and for your endless patience and guidance during my early years.I am grateful for the opportunity to collaborate with you at OctoML and delve into TVM. Alex, thanks for not only being like a big brother taking care of the lab and my questions during the first two years in the lab, but also providing invaluable assistance during my first PhD paper submission. Edward, thank you for your mentorship during my first couple years especially when I felt completely lost in the beginning of my PhD. Eric, you truly defined what is a good labmmate and taught me how to deliver high-quality research and presentation. Your examples have been a source of inspiration to me. Sidhant, I miss the time we spent working on the baby project and I am grateful for everything I learned from you about entrepreneurship and industry. CJ, your kindness and willingness to share resources and provide answers to my random questions have been invaluable. Gabe, thank you for your patience to provide feedback to my various fellowship applications. Finally, I would like to express my appreciation for every single lab alumini who has provided help and advice along the way. I will always look up to you all and hope that our paths will cross again in the future.

I would like to extend my gratitude to my friends at UW and the Allen School. Ziheng and Chengqian, our discussions on the latest technology trends have always been a source of enjoyment, and our debates and analysis of listed companies and investments have been a

highlight of our friendship. I hope our bond continues to thrive. Orson, you truly exemplify what is a great researcher. Your passion for research is inspiring and your dedication has motivated me to strive for excellence. I look forward to our future collaboration as you embark on your path as a professor. Marrisa, our friendship dates back to our time at Amherst, and I am grateful for your unwavering support and for helping me with my applications. Xiaoyi, as the first student I met at the Allen School, I appreciate your patience in answering all of my questions about graduate school, research, and career. You are a perfect example of what it means to be a successful industry researcher to me. Yuntao, I am grateful for your exceptional collaboration on our research projects and for consistently providing the resources I needed throughout my PhD journey. I am honored to have been a part of the ASSP summer programs that you led over the past three years and proud of the impact you have made in the field.

Additionally, I would like to extend my appreciation to my research collaborators, including Dr. Yang, Teddy, Brian, Yuzhe, Roni and Akshay. Your expertise and guidance have been invaluable, as I have learned so much from you across various areas such as cardiology and machine learning. A special thanks to my undergraduate research advisor, Ivan Lee, who introduced me to the field of mobile health and taught me the fundamentals of research. Without your guidance and mentorship, I wouldn't be where I am today.

I also want to thank to all my internship mentors: Vu, Josh, Bin, Daniel, Jiang, Silviu, Ming, Jake, Zaid, and Samy. Thanks to all of you, I was able to undertake this unique joint industry-academia PhD program, dedicating 60% of my time to industry. I am appreciative of the knowledge and perspectives you shared regarding industry R&D and how to strike a balance between research and engineering to produce world-class products. My time at industry had a profound impact on me and changed my mind to pursue a career in the industry.

Lastly, I would like thank the funding agencies, corporations, and foundations that have

supported my PhD research. This includes the Google PhD Fellowship, Microsoft, Cisco, and the Bill & Melinda Gates Foundation.

DEDICATION

to my family.

Chapter 1

INTRODUCTION

1.1 Overview

The SARS-CoV-2 (COVID-19) pandemic is transforming the face of healthcare around the world [130, 125]. One example of this is the sharp increase (by more than 10x) in the number of medical appointments held via telehealth platforms because of the increased pressures on healthcare systems, the desire to protect healthcare workers and restrictions on travel [125]. Telehealth includes the use of telecommunication tools, such as phone calls and messaging, and online health portals that allow patients to communicate with their providers. The Center for Disease Control and Prevention is recommending the "use of telehealth strategies when feasible to provide high-quality patient care and reduce the risk of COVID-19 transmission in healthcare settings"[1]. Performing primary care visits from a patient's home reduces the risk of exposing people to infections, increases the efficiency of visits and facilitates care for people in remote locations or who are unable to travel. These are longstanding arguments for telehealth and will still be valid after the end of the current pandemic. Healthcare systems are likely to maintain a high number of telehealth appointments in the future [113].

However, despite the longstanding promise of telehealth, it is difficult to provide a similar level of care on a video call as during an in-person visit. The physician *can* diagnose a patient based on visual observations and self-reported symptoms; however, in most cases they *cannot* objectively assess the patient's physiological state. This means that physicians have to make decisions (e.g., recommending a trip to the emergency department) without important data. In the case of COVID-19, there are severe cardiopulmonary (heart and lung related) symptoms that are difficult to evaluate remotely. The symptoms seen in patients

[1] https://www.cdc.gov/coronavirus/2019-ncov/hcp/ways-operate-effectively.html

have drawn links to acute respiratory distress syndrome [156], myocardial injury, and chronic damage to the cardiovascular system. Experts suggest that particular attention should be given to cardiovascular protection during treatment [164]. The development of more accurate and efficient camera cardiopulmonary measurement technology would give remote physicians access to the data to make more informed decisions. Beyond telehealth, the same technology could impact passive health monitoring, improving the standard of care for infants in neonatal intensive care units [146].

Cameras can be used to measure physiological signals, including heart and respiration rates, and blood oxygenation levels [112, 46, 27], based on facial videos [134, 144]. camera cardiopulmonary measurement involves capturing subtle changes in light reflected from the body caused by physiological processes. Imaging methods can be used to measure volumetric changes of blood in the surface of the skin cause changes in light absorption (↑ volume of hemoglobin = ↑ light absorption). This in turns affects the amount of visible light reflected from the skin, which is the source of the photoplethysmogram (PPG). The mechanical force of blood pumping around the body also causes subtle motions and these are the source of the ballistocardiogram (BCG). These color and motion changes in the video help us extract the pulse signal and heart rate frequency. More optical background and details are provided in Chapter 2. The PPG and BCG signals provide complementary information to one another and also contain information about breathing due to respiratory sinus arrhythmia [111]. Respiratory signals can also be recovered from motion-based analyses of the head and torso as the subjects breathes in and out [135].

1.2 Challenges in Camera Health Sensing

Although there are a number of strengths to camera health sensing, I argue that there remain several technical challenges that need to be overcome before it reaches its potential. Firstly, accuracy of measurements is critical to avoid false alarms or misdiagnoses. The US Federal Drug Administration (FDA) mandates that testing of a new device for cardiac monitoring should show "substantial equivalence" in accuracy with a legal predicate device (e.g., a contact

sensor)[2]. This standard has not been obtained in camera approaches.

Secondly, designing models that run on-device helps reduce the need for high-bandwidth Internet connections making telehealth more practical and accessible. Thirdly, camera health sensing is a highly privacy sensitive application. These data are personally identifiable, combining videos of a patient's face with sensitive physiological signals. Therefore, streaming and uploading data to the cloud to perform analysis is not ideal. The ability to run at a high frame rates enables opportunistic sensing (e.g., obtaining measurements each time you look at your phone) and helps capture waveform dynamics that could be used to detect arterial fibrillation [20], hypertension [51], and heart rate variability [88] where high-frame rates (at least 100Hz) are a requirement to yield precise measurements to capture beat-to-beat variability in millisecond level.

Thirdly, ensuring that machine learning-based camera health sensing systems are equitable and useful in real-world deployments is crucial. However, existing optical sensors and algorithms introduce significant bias when evaluating minority populations, such as individuals with darker skin tones. Achieving equitable camera health sensing remains a challenge.

Fourthly, previous research has primarily focused on evaluating the healthy population in controlled lab settings and has not adequately investigated the performance of camera health sensing in real-world clinical settings or in individuals with irregular vital signs, such as arrhythmia. Furthermore, previous studies mainly focused on estimating well-studied vital signs including heart rate and respiration rate. camera sensing has been moving slowly to explore new clinically valuable measures.

Lastly, standardization in camera health sensing is still severely lacking. Replication of results and benchmarking of new models is critical for scientific progress. However, as with many other applications of deep learning, reliable codebases are not easy to find.

---

[2]https://www.fda.gov/regulatory-information/search-fda-guidance-documents/cardiac-monitor-guidance-including-cardiotachometer-and-rate-alarm-guidance-industry#6_1

1.3 Thesis Contributions and Outline

In this thesis, I argue that **Novel machine learning techniques can help enable accessible, equitable, generalizable and useful camera health sensing systems.** As below, I define the terms 'accessible', 'equitable', 'generalizable', and 'useful' as they apply to this thesis.

In this thesis, **accessible** implies that the camera health sensing technology I aim to develop should be within reach of all potential populations, irrespective of the type of mobile device they possess. It acknowledges the need for these tools to be inclusive, accommodating a wide range of devices from high-end smartphones to more affordable, low-end mobile devices. Accessible also embodies the requirement for the technology to operate in real-time or near-real-time speeds, delivering outputs at a minimum frame rate of 30-60 frames per second. This aspiration is in line with the performance of modern medical devices, which frequently run at 100Hz or more to capture high-fidelity signals for advanced applications such as heart rate variability monitoring and waveform analysis. Overall, the goal is to develop deep learning models that optimize computational resources without compromising their efficacy or accuracy, making accessibility not just about reach, but also about efficient performance.

Equitability in this thesis refers to a uniform performance of our camera health sensing systems across diverse skin tones and physical attributes, reducing potential biases. Alongside, **generalizability** refers to the robustness of the proposed methods to various noises such as diverse backgrounds or differing lighting conditions. The models should be capable of effective learning from a broad dataset, thereby adapting to a wide range of situations, even those that are unforeseen. Together, equitability and generalizability stress on constructing a camera health sensing system that is fair across user demographics and robust to environmental noise.

The **usefulness** of the camera health sensing systems I aim to develop will be measured not just by its performance in lab settings or its validation against healthy populations.

Rather, it will be assessed by its practical utility in clinical settings, its ability to contribute to actual health outcomes. A truly useful technology is one that can withstand the complexities and challenges of real-world clinical environments and contribute meaningfully to health outcomes. Thus, usefulness here is synonymous with clinical relevance, emphasizing the need for our technology to be validated by health practitioners, to provide actionable insights, and to demonstrate its value in real-life healthcare scenarios.

To support this thesis, I propose to address the following research questions:

RQ.1: How to design an on-device neural networks to enable accessible camera health sensing?

In Chapter 3, I present two generations of on-device efficient neural networks to enable real-time camera health sensing on mobile devices while achieving state-of-the-art accuracy and making the networks simple to deploy. Published in `NeurIPS 2020` [70] and `WACV 2023` [71].

RQ.2: How to make camera health sensing more equitable and generalizable in real-world deployments?

In Chapter 4, I first present MetaPhys, an algorithm that uses unsupervised labels for few-shot personalization and adaptation. Building on top of MetaPhys, I propose MobilePhys, a dual-camera mobile system that generates pseudo labels for few-shot personalization. I then explore the use of weighed federated learning to enable collaborative learning with imperfect data. Finally, I present a dataset of synthetic data for camera health sensing. The goal of this proposed work is to promote equitable camera health sensing systems that can function effectively in diverse skin tones, contexts, and with varying levels of motion. Published in `CHIL 2021` [72], `IMWUT 2022` [74], `NeurIPS 2022` [92] and `CVPR-W 2022` [75].

RQ.3: How to push the limit of camera health sensing beyond cardiac measurement in clinical settings?

In Chapter 5, I first validate the robustness of heart rate measurements using the data collected in our transnational clinical study. I then present the results of how I have expanded the potential of camera health sensing by estimating blood pressures from facial videos in a non-contact manner. This is the first clinical study exploring blood pressure estimation from facial videos. An abstract published in `American Cardiology College 2022` [31] and a journal paper is under preparation.

RQ.4: How to design a toolbox to standardize the training and evaluation regime in camera health sensing?

In Chapter 6, I present an open-source toolbox that can assist the community in standardizing the processes of preprocessing, training, and evaluation for camera health sensing. The toolbox supports popular public datasets, as well as supervised neural networks and unsupervised methods. A paper is under submission.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Optical Basis

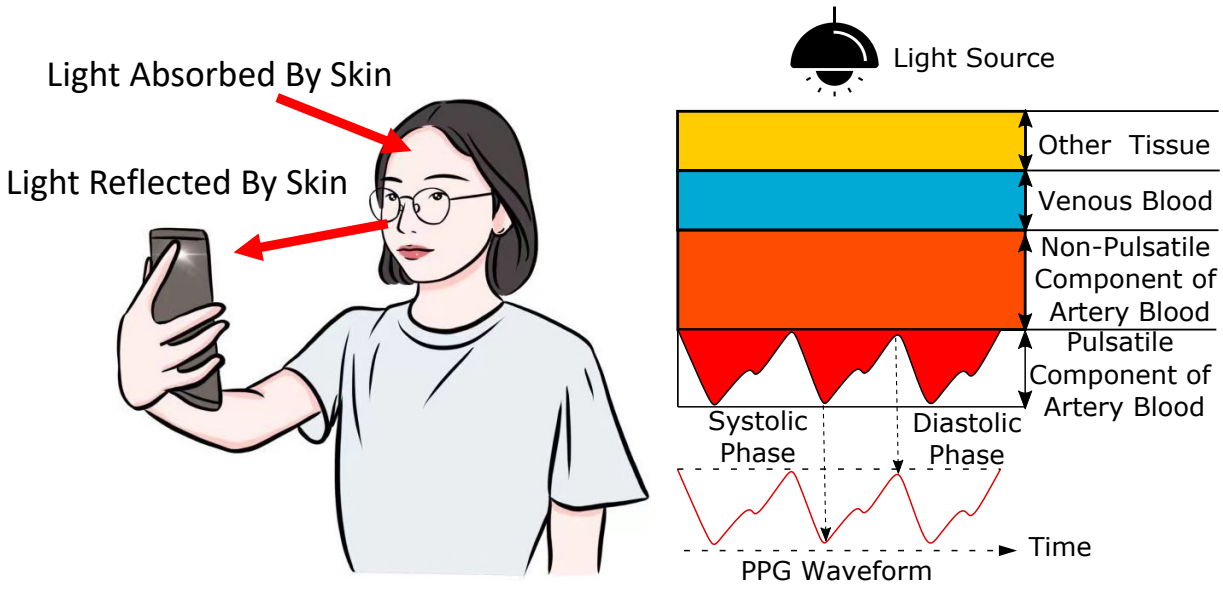

Figure 2.1: Optical Basis of Camera Health Sensing

In the field of camera health sensing, we work on developing computational methods for extracting physiological signals (e.g., pulse rate, respiration rate, blood oxygenation, blood pressure) based on videos of the human body. In principle, as Figure 2.1 is illustrated, these methods use pixel information to quantify visible light, or other electromagnetic radiation (e.g., infrared or thermal), reflected from the body. This reflected radiation is modulated by

motions of the body and absorption characteristics of the skin [11, 134, 145, 112, 63, 155, 148].

As visible light penetrates between 4 to 5 mm below the skin's surface, it is modulated by the volume of oxygenated and deoxygenated hemoglobin enabling the measurement of the peripheral blood volume pulse (BVP) via photoplethysmography (PPG). The frequency channels offered by multiband (e.g., RGB) cameras enable the composition of blood, including the oxygen saturation to be measured. In addition, these pixels are affected by the motion as a person breaths in and out and by the mechanical effects of the heart beating, enabling the measurement of breathing signals and the ballistocardiogram (BCG). Analyzing the morphology of these signals, and combining them together, offers the possibility of measuring correlates of blood pressure.

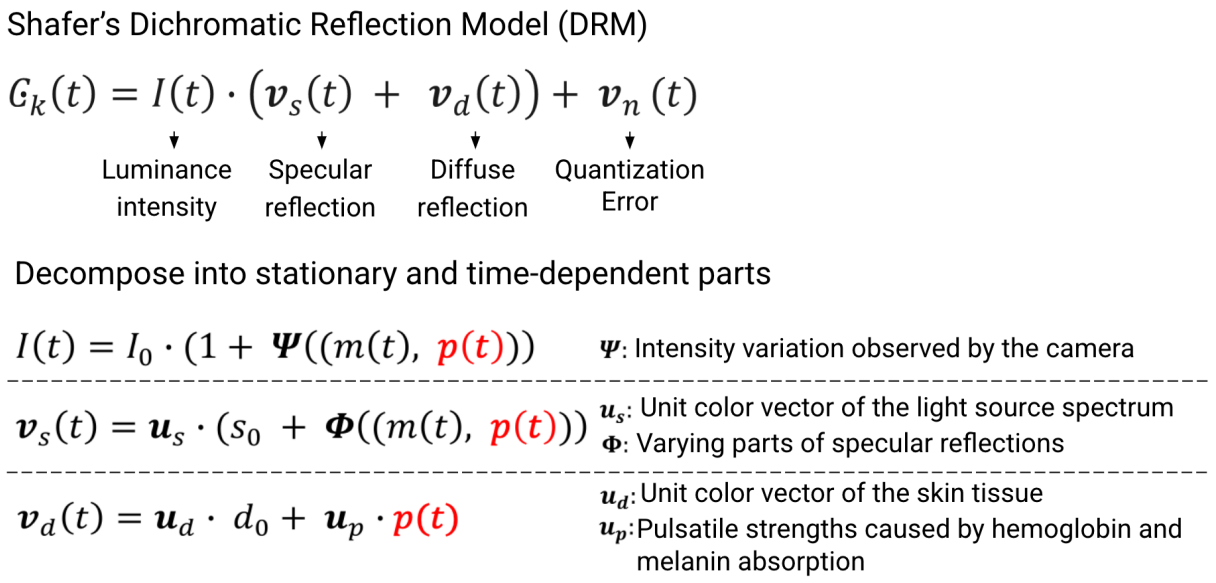

Figure 2.2: Mathematical Optical Basis

To delve deeper into the optical basis of camera health sensing, let's start with Shafer's Dichromatic Reflection Model (DRM) (see 2.2), as in prior work [148, 27]. Specifically, we aim to capture both spatial and temporal changes and the relationship between multiple

physiological processes. Let us start with the RGB values captured by the cameras as given by:

$$\boldsymbol{C}_k(t) = I(t) \cdot (\boldsymbol{v}_s(t) + \boldsymbol{v}_d(t)) + \boldsymbol{v}_n(t) \tag{2.1}$$

where $I(t)$ is the luminance intensity level, modulated by the specular reflection $\boldsymbol{v}_s(t)$ and the diffuse reflection $\boldsymbol{v}_d(t)$. The quantization noise of the camera sensor is captured by $\boldsymbol{v}_n(t)$. Following [148] we can decompose $I(t)$, $\boldsymbol{v}_s(t)$ and $\boldsymbol{v}_d(t)$ into stationary and time-dependent parts:

$$\boldsymbol{v}_d(t) = \boldsymbol{u}_d \cdot d_0 + \boldsymbol{u}_p \cdot p(t) \tag{2.2}$$

where $\boldsymbol{u}_d$ is the unit color vector of the skin-tissue; d_0 is the stationary reflection strength; $\boldsymbol{u}_p$ is the relative pulsatile strengths caused by hemoglobin and melanin absorption; $p(t)$ represents the physiological changes.

$$\boldsymbol{v}_s(t) = \boldsymbol{u}_s \cdot (s_0 + \Phi(m(t), p(t))) \tag{2.3}$$

where $\boldsymbol{u}_s$ denotes the unit color vector of the light source spectrum; s_0 and $\Phi(m(t), p(t))$ denote the stationary and varying parts of specular reflections; $m(t)$ denotes all the non-physiological variations such as flickering of the light source, head rotation, and facial expressions.

$$I(t) = I_0 \cdot (1 + \Psi(m(t), p(t))) \tag{2.4}$$

where I_0 is the stationary part of the luminance intensity, and $I_0 \cdot \Psi(m(t), p(t))$ is the intensity variation observed by the camera. As in [27] we can disregard products of time-varying components as they are relatively small:

$$\begin{aligned}\boldsymbol{C}_k(t) \approx \boldsymbol{u}_c \cdot I_0 \cdot c_0 + \boldsymbol{u}_c \cdot I_0 \cdot c_0 \cdot \Psi(m(t), p(t)) + \\ \boldsymbol{u}_s \cdot I_0 \cdot \Phi(m(t), p(t)) + \boldsymbol{u}_p \cdot I_0 \cdot p(t) + \boldsymbol{v}_n(t)\end{aligned} \tag{2.5}$$

However, unlike in previous work which modeled pulse and respiration signals as independent [26], we also leverage the fact that *p(t)* actually captures a complex combination of both pulse and respiration information. Specifically, both the specular and diffuse reflections

are influenced by related physiological processes. Respiratory sinus arrhythmias (RSA) are rhythmical fluctuations in heart periods at the respiration frequency [9]. Furthermore, the respiration and pulse signals both cause outward motions of the body in the form of chest and head motions. We can say that the physiological process *p(t)* is a complex combination of both the blood volume pulse, *b(t)*, and the respiration wave, *r(t)*. Thus, $p(t) = \Theta(b(t), r(t))$ and the following equation gives a more accurate representation of the underlying process:

$$\boldsymbol{C}_k(t) \approx \boldsymbol{u}_c \cdot I_0 \cdot c_0 + \boldsymbol{u}_c \cdot I_0 \cdot c_0 \cdot \Psi(m(t), \Theta(b(t), r(t))) + \boldsymbol{u}_s \cdot I_0 \cdot \Phi(m(t), \Theta(b(t), r(t))) + \boldsymbol{u}_p \cdot I_0 \cdot p(t) + \boldsymbol{v}_n(t) \quad (2.6)$$

Since *b(t)* and *r(t)* are so closely intertwined, a temporal multi-task learning approach would seem suitable for this problem and at very least could leverage redundancies between the two signals.

2.2 Fundamentals of Algorithmic Basis

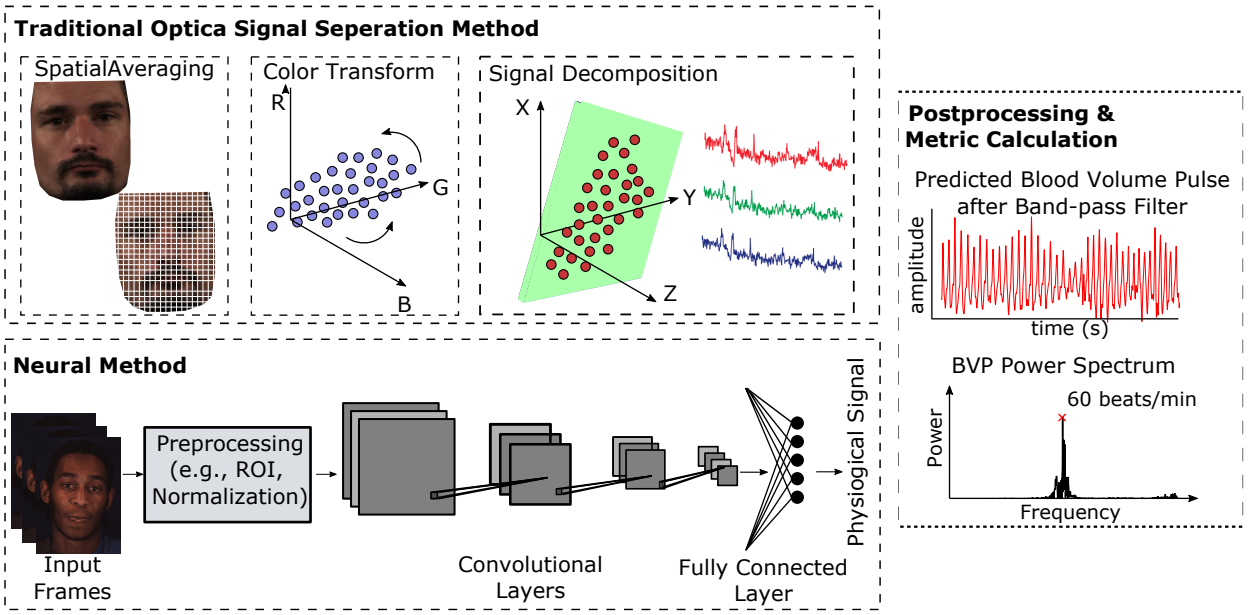

Figure 2.3: A general overview of traditional unsupervised and neural network based methods

2.2.1 Unsupervised Signal Separation Algorithms

Early work established that the blood volume pulse can be extracted by analysing skin pixel intensity changes over time [134, 144]. These methods are grounded by optical models (e.g., the Lambert-Beer law (LBL) and Shafer's dichromatic reflection model (DRM)) that provide a framework for modeling how light interacts with the skin. However, traditional signal processing techniques are quite sensitive to noise from other sources in video data, including head motions and illumination changes [112, 111]. To help address these issues, some approaches incorporate prior knowledge about the physical properties of the patient's skin [33, 148]. Plane-Orthogonal-to-Skin (POS) [148] is the current state-of-the-art for demixing in this context. POS calculates a projection plane orthogonal to the skin-tone, derived based

on optical and physiological principles, that is then used for pulse extraction. The POS method can be summarized as follows: 1) spatially averaging pixel values for each frame with the region-of-interest, 2) temporally normalizing the resulting signals within a certain window size calculated relative to the frame rate, 3) applying a fixed matrix projection to offset the specular reflections and other noise, 4) band-pass filtering of the resulting pulse waveform. Although effective, these handcrafted signal processing pipelines make it difficult to capture the complexity of the spatial and temporal dynamics of physiological signals in video.

2.2.2 Supervised Algorithms

Since the underlying relationship between the pulse and skin pixels is complex, deep convolutional neural networks [27, 131, 159, 129, 97, 98, 128, 82] have shown superior performance over the traditional source separation unsupervised algorithms. DeepPhys [27] was the first paper that demonstrated that a deep neural network outperforms all the traditional signal processing approaches. Yu et al. [159] have shown that applying 3D convolutional neural networks (CNNs) significantly improves performance and achieves better accuracy compared to using a combination of 2D CNNs and recurrent neural networks. The benefit of 3D CNNs implies that incorporating temporal data in all layers of the model is necessary for high accuracy. More recently, an adversarial learning approach, called Dual-GAN, has also been studied to learn noise-resistant mappings from video frames to pulse waveform and noise distributions [82]. With two generative-adversarial networks, they can promote each adversarial network's representation and further improve the feature disentanglement between pulse and various noise sources.

Although convolutional neural networks have been widely studied and used in many computer vision applications, vision transformers were first successfully used on the task of image classification. By training on larger datasets, vision transformer (ViT) attains excellent performance and can be used in downstream fine-tuning with fewer amounts of data [37]. More recently, the state-of-the-art Swin vision transformer was proposed as a way to construct hierarchical feature maps and improve computational efficiency by using a

hierarchical representation and limiting self-attention computation to non-overlapping local windows while allowing for cross-window connection [77]. However, transformer architectures have been barely studied in the field of camera vitals measurement. The closest work used transformers to detect remote photoplethysmography (rPPG) for attack/spoofing detection [158]. However, this paper did not evaluate the proposed vision transformer in the task of heart rate estimation using any public datasets, which is considered as the gold-standard benchmark for the field of camera vital measurement. More recently, Yu *et al.* recently proposed Physformer [161] which is also a visual transformer based architecture.

2.2.3 Efficient Camera Health Sensing

Although supervised neural networks have shown superior performance, these methods still struggle with effectively combining spatial and temporal information while maintaining a low computational budget. More recently, researchers have investigated on-device remote camera heart rate variability measurement using facial videos from smartphone cameras [54]. However, their proposed architecture requires approximately 200ms per frame inference, which is insufficient for real-time performance, and was not evaluated on public datasets.

As [159] shows, spatial-temporal modeling is the key of accurate camera phhsyiologcal measurement. However, direct temporal modeling with 3D CNNs requires dramatically significant computational resources and make real-world deployment challenging. In addition to reducing computational cost, there are several reasons that it is highly desirable to be able to have efficient non-contact physiological measurement models that run on-device. Temporal Shift Modules [69] provide a clever mechanism that can be used to replace 3D CNNs without reducing accuracy and requiring only the computational budget of a 2D CNN. This is achieved by shifting the tensor along the temporal dimension, facilitating information exchange across multiple frames. TSM has been evaluated on the tasks of video recognition and video object detection and achieved superior performance in both latency and accuracy. Xiao et al. [154] used pretrained TSM-based residual networks as a backbone followed by two attention modules for reasoning about human-object interactions. The differences between

this aforementioned work and ours is they applied attention modules as the head followed by pretrained TSM-based residual feature maps while our work applies two attention modules to the intermediate feature maps generated from regular 2D CNNs with TSM.

Moreover, current neural methods require complex preprocessing procedures. For example, DeepPhys [27] requires a few preprocessing steps including calculating difference frames and performing image normalization. Dual-GAN has a even more complex preprocessing module called MSTMaps proposed by [98]. The MSTMaps are a set of multi-scale spatial temporal maps created by 1) cropping the facial region, 2) extracting facial landmarks, 3) performing average pooling for every color channel and every ROI combination for each frame, 4) generating ROI combinations using all the detected ROI regions and landmarks, 5) multiplying each item in all ROI combinations with six channels. The final size of the MSTMap is $(2^n - 1) \times T \times 6$ where T is the number of frames and n is the number of ROI regions. Such a preprocessing module not only consumes large amounts of memory but also introduces a large computational burden to the entire pipeline. Moreover, stacking all of these extra procedures makes development and deployment much more difficult. Unlike these methods, the goal of our proposed method is to create a preprocessing-free neural architecture that is simple to use and deploy, efficient on mobile devices, and accurate on settings with various types of noise. gbbn,.

2.2.4 Few-shot Adaption and Personalization for Camera Health Sensing

The ability to learn from a small number of samples or observations is often used as an example of the unique capabilities of human intelligence. However, machine learning systems are often brittle in a similar context. Meta-learning approaches tackle this problem by creating a general learner that is able to adapt to a new task with a small number of training samples, inspired by how humans can often master a new skill without many observations [52]. However, most of the previous work in meta-learning focuses on supervised vision problems [165, 126] and in the computer vision literature has mainly been applied to image analysis [147, 68]. Supervised regression in video settings has received less attention. One

of few examples is object or face tracking [29, 104]. In these tasks, the learner needs to adapt to the individual differences in appearance of the target and then track it across frames, even if the appearance changes considerably over time in the video. Choi et al. [29] present a matching network architecture providing the meta-learner with information in the form of loss gradients obtained using the training samples. Lee et al. [64] recognized the potential for meta-learning applied to imaging-based cardiac pulse measurement. Their method (Meta-rPPG) focuses on using transductive inference based meta-learning and a LSTM encoder-decoder architecture which to our knowledge was not validated in previous work. Instead, our proposed meta-learning framework is built on top of a state-of-the-art on-device network [70] and aims to explore the potential of both supervised and unsupervised on-device personalized meta-learning. More specifically, Meta-rPPG uses a synthetic gradient generator and a prototypical distance minimizer to perform transductive inference to enable self-supervised meta-learning. This learning mechanism requires a number of rather complex steps with transductive inference. We propose a somewhat simpler mechanism that is physiologically and optically grounded [148, 70] and achieves greater accuracy.

The property of fast adaptation makes meta-learning a good candidate for personalizing models, it has been used in various applications such as dialogue agents [83], gaze estimation [48], sleep stage classification [8], activity recognition [44], and video retargeting [65]. For example, Banluesombatkul et al. proposed a MAML-based meta-learning system to perform fast adaption of a sleep stage classification model using biosignals [8]. More recently, MetaPix [65] leveraged a meta-learning training schema with a small amount of video to adapt a universal generator to a particular background and human in the problem of video retargeting. Similarly, our proposed meta-learning framework is also capable of personalizing a universal remote physiological model to a new person or an environmental setting.

2.2.5 Federated Learning in Healthcare

Federated learning enables training machine learning models from a set of distributed remote devices (e.g., mobile devices) while storing data only on the individual clients. Early work

established optimization principals on how to perform non-convex optimization on distributed client's model weights [93]. Due to federated learning's unique characteristics in protecting privacy, it has been used and studied in healthcare applications. The volume of training data in healthcare applications is often smaller than in many traditional machine learning tasks. Therefore, aggregating as much data as possible from decentralized clients' could help boost the performance of machine learning applications in healthcare while reducing the chances of leaking sensitive information or violating HIPAA guidelines [157, 118]. Brisimi et al. [16] proposed to use federated learning to train a supervised classification model for cardiac events. More specifically, they develop a federated learning based framework to enable multiple data holders (i.e., hospitals) to collaborate and converge to a centralized model. More recently, [28] proposed a framework that leveraged federated learning to perform transfer learning for wearable sensors called FedHealth. In this framework, when the clients receive the updated model weights from the server all the layers in the neural network are frozen except for the last two fully connected dense layers. They claim that fine-tuning the last two layers on the client side can help build personalized models for each user or organization. FedHealth was evaluated on a Parkinson's disease dataset. The application of federated learning in COVID-19 has also been investigated. Qayyum et al. [114] explored the use of federated learning in automatic diagnosis of COVID-19. They demonstrated improvements on results of X-ray and Ultrasound datasets after using federated learning. In the field of physiological measurement, Brophy et al. [17] investigated the use of federated learning and generative adversarial networks to estimate continuous blood pressure from the PPG signal. This work is quite distinct from ours as it uses contact sensor based PPG measurements while our work is focused on deriving the PPG signal and heart rate from facial videos.

2.2.6 Training on Synthetics

One of the most notable properties of neural models is how they scale efficiently with the number of training examples. A large amount of engineering and research efforts have been invested in scaling learning infrastructures so that models with vast numbers (millions or

billions) of parameters can be trained with time efficiency. However, it is becoming increasingly difficult to collect sufficient volumes of labeled data to exploit this scale, especially for video-based applications.

Using parameterized graphics simulations to augment existing datasets have been extensively explored in different computer vision domains [124, 141, 142, 143, 140, 47] such as training pose recognition [124], scene segmentation for self-driving cars [120], improving object recognition [107], detecting pedestrians under different conditions [140], and for performance evaluation of learned models [47]. AirSim is a graphics-based simulation environment [122] that has been successfully used in the context of training autonomous drone navigation [15]. In the context of physiological sensing; however, synthetic data has been mostly used for evaluation purposes of different algorithms considering other modalities (e.g., [36, 102, 22]). To the best of our knowledge, our work is the first example of using high-fidelity physiological simulations to train rPPG methods. Fortunately, the importance of human bodies and faces, especially in the entertainment industry (e.g., for CGI movies), has fueled a large amount of innovation towards the creation of high-fidelity human graphical simulations.

Synthetics have proven particularly valuable for face and body analyses. In training, synthetics have been used successfully to create models for landmark localization and face parsing [152], body pose estimation [124] and eye tracking [153]. Although not completely representative of real observations, synthetics are also valuable in testing (e.g., for face detection or eye tracking [133]).

Our work is made possible thanks to the ability to render high-fidelity frames/videos with an optical basis for manipulating blood volume in the skin. Creating realistic blood flow simulations is achieved by modeling the appearance of multiple translucent skin layers [35, 57, 4]. These dynamic appearance models usually capture the subsurface scattering that occurs when light interacts with the outer layers of the skin, and are motivated by in-vivo measurements of melanin and hemoglobin concentrations [56]. In this thesis, we propose that synthetic data can be successfully used for training rPPG systems and leverage these innovations in rendering. We create synthetic data to show how non-contact vital sign

measurement can be improved using synthetic data.

2.2.7 Open-Source Toolbox for Camera Health Sensing

Open source code allow researchers to compare novel approaches to consistent baselines without ambiguity regarding the implementation or parameters used. This transparency is important as subsequent research invariably builds on prior state-of-the-art. Implementing a prior method from a paper, even if clearly written, can be difficult. Furthermore, it is an inefficient use of time for many researcher to re-implement all baseline methods. To address this, several open source toolboxes have been released for camera physiological sensing. These toolboxes have been a significant contribution to the community and provide implementations of methods and models [85, 13, 108]. However, these toolboxes are incomplete. McDuff and Blackford [85][1] implemented a set of source separation methods (Green, ICA, CHROM, POS) and Pilz [108] published the PPGI-Toolbox[2] containing implementations of Green, SSR, POS, Local Group Invariance (LGI), Diffusion Process (DP) and Riemannian-PPGI (SPH) models. These toolboxes are implemented in MATLAB (e.g., [85]); however, Python is now the language of choice for a large majority of computer vision and deep learning research. There are several implementations of popular signal processing methods in Python: Bob.rrpg.base[3] includes implementations of CHROM, SSR and Boccignone et al. [13] released code for Green, CHROM, ICA, LGI, PBV, PCA, and POS. Several published papers have included links to code; however, often this is only inference code and not training code for neural models. Without providing training code for neural networks, it is challenging for researchers to conduct end-to-end reproducible experiments and build ideas on top of it.

---

[1] https://github.com/danmcduff/iphys-toolbox

[2] https://github.com/partofthestars/PPGI-Toolbox

[3] https://pypi.org/project/bob.rppg.base/

2.3 Benchmark Datasets

AFRL [39]: There is a total of 300 videos from 17 male participants and 8 female participants. The resolution of each video is 658 x 492 and the sampling rate is 120 fps. A fingertip reflectance medical-grade photoplethysmograms (PPG) device was provided to record ground-truth PPG signal for training the network and for evaluating the performance of our proposed system. During the data collection, every participant was asked to keep stationary for the first two tasks and perform head motion tasks in the subsequent four tasks. These motion tasks include rotating their head along the vertical axis, horizontal axis as well as orienting their head randomly to one of nine predefined locations. For the vertical and horizontal rotations, participants were asked to rotate in an angular velocity of 10 degrees/second, 20 degrees/second, 30 degrees/second, respectively. The six recording were repeated twice with two backgrounds. This data collection protocol was approved by the institutions IRB.

MMSE-HR [163]: 102 videos of 40 participants were recorded at 1040x1392 resolution and 25 fps during spontaneous emotion elicitation experiments. The gold standard contact signal was measured via a Biopac2 MP150 system[4] which provided pulse rate at 1000 fps and was updated after each heart beat. These videos feature smaller but more spontaneous motions than those in the AFRL dataset including facial expressions. Respiration measurements were not provided.

UBFC [12]: A total of 42 videos from 42 participants were recorded at resolution of 640 x 480 and sampling rate of 30 fps. UBFC has a similar volume as MMSE, which is also smaller than AFRL. All the videos are recorded at uncompressed 8-bit RGB format. The medical-grade pulse oximeter (CMS50E transmissive pulse oximeter) was used to record PPG signal for evaluation. All the participants were asked to keep stationary during the experiments. This data collection protocol was approved by the institutions IRB.

PURE [132]. A total of 60 one-minute videos comprises of 10 individuals (8 male, 2 female) who were recorded in six different settings. The videos were captured using an

[4] https://www.biopac.com/

eco274CVGE camera at 30fps with a resolution of 640x480 pixels. The PPG data, including pulse rate wave and SpO2 readings, were acquired in parallel by a CMS50E transmissive pulse oximeter at a sampling rate of 60Hz. The PURE dataset is widely used due to its diversity of motions, including talking, translation, and head rotation. However, the lack of variety in skin tones, real-world motion tasks and lighting conditions makes it less suitable for handling complex scenarios.

SCAMPS [92]: A total of 2,800 videos (1.68M frames) of synthetic avatars with aligned cardiac and respiratory signals. The waveforms and videos were synthesized using a sophisticated facial pipeline that produces high fidelity, almost photo-realistic renderings. The videos have a range of confounders including head motions, facial expressions, and ambient illumination changes.

Chapter 3

ON-DEVICE NEURAL NETWORKS FOR ACCESSIBLE CAMERA HEALTH SENSING

3.1 Introduction

On-device neural models running help reduce the need for high-bandwidth Internet connections making camera health sensing more practical and accessible. camera health sensing is a highly privacy sensitive application. These data are personally identifiable, combining videos of a patient's face with sensitive physiological signals. Therefore, streaming and uploading data to the cloud to perform analysis is not ideal. The ability to run at a high frame rates enables opportunistic sensing (e.g., obtaining measurements each time you look at your phone) and helps capture waveform dynamics that could be used to detect arterial fibrillation [20], hypertension [51], and heart rate variability [88] where high-frame rates (at least 100Hz) are a requirement to yield precise measurements.

While prior research has framed architectures as "end-to-end" methods, those that achieve state-of-the-art performance actually require several preprocessing steps before data is used as input to the network. For example, [27] uses hand-crafted normalized difference frames and normalized appearance frames as input to their convolutional attention network. [98] and [82] use a complex schema to create feature maps called "MSTmaps", their process includes facial landmark detection, extraction of several regions of interest (ROI) using these landmarks, and then averaging pixel values in both the RGB and YUV color spaces. These preprocessing steps are computationally costly and in many cases add a significant number of operations to the video processing pipeline. There are several reasons why running camera physiological sensing on-device is desirable: privacy preservation, the ability to use raw (i.e., uncompressed) video and data cost and bandwidth savings. Many of these steps are also non-trivial to implement

and optimize in and of themselves. This makes it harder to deploy real-time systems and to replicate the implementation on different platforms. For instance, implementing existing methods on Android, iOS, or in JavaScript requires a significant amount of effort. Some libraries, such as facial landmark detection, are not even available on every platform. Thus, the last mile engineering using the existing methods becomes especially challenging.

In this chapter, we introduce two generations on-device neural networks for camera health sensing. First, we propose a novel multi-task temporal shift convolutional attention network (MTTS-CAN) to address the challenges of privacy, portability, and precision in contactless cardiopulmonary measurement. MTTS-CAN leverages temporal shift modules to perform efficient temporal modeling and remove various sources of noise without any additional computational overhead. An attention module improves signal source separation, and a multi-task mechanism shares the intermediate representations between pulse and respiration to jointly estimate both simultaneously. By combining these three techniques, our proposed network can run on an ARM CPU and achieve the state-of-the-art accuracy and inference speed. To summarize, the contributions of MTTS-CAN are to 1) present the first accurate and efficient approach to perform on-device real-time spatial-temporal modeling of vitals signal, 2) evaluate our system and show state-of-the-art performance on two large public datasets, 3) provide an implementation of core tensor operations required for MTTS-CAN using a modern deep learning compiler and an on-device latency evaluation across different architectures showing MTTS-CAN can run at more than 150 frame per second. Our code are available in `https://github.com/xin71/MTTS-CAN`.

Building on top of MTTS-CAN, we introduce a truly end-to-end on-device network, EfficientPhys, for which the input is unprocessed video frames without requiring accurate face cropping (see Fig. 3.3). Due to recent advancements in visual transformers, we propose both a convolutional and visual transformer architecture and compare and contrast the performance of these two. In summary, our key contributions of EfficientPhys are to: 1) propose two novel one-stop neural architectures, a visual transformer and a convolutional network, which do not require any preprocessing steps, 2) evaluate the proposed methods on

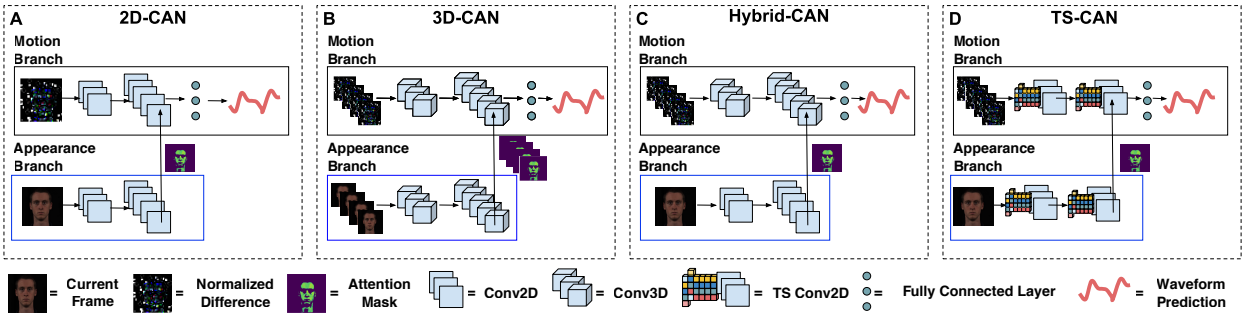

Figure 3.1: We perform a systematic comparison of several convolutional attention network (CAN) architectural designs. Starting from previous work that presented a 2D-CAN [26], we introduce a fully 3D-CAN, a 2D-3D Hybrid CAN in which the appearance branch takes a single frame, and our proposed temporal shift CAN. Each of these models can be applied in a single or multi-task manner.

three popular benchmark datasets, 3) evaluate on-device latency across both state-of-the-art machine learning-based approaches as well as signal processing-based techniques. To the best of our knowledge, this is the first paper that explores the visual transformer in camera physiological measurement and its comparison with convolutional networks. This is also the first paper exploring a completely end-to-end on-device neural architecture for mobile devices.

3.2 End-to-End On-Device Networks

3.2.1 MTTS-CAN: Multi-Task Temporal Shift Convolutional Attention Network

Efficient Spatial-Temporal Modeling: To achieve state-of-the-art performance in on-device optical cardiopulmonary measurement, an architecture should have the ability to: 1) efficiently learn spatial features that map raw RGB values to latent representations corresponding to the pulse and respiratory signals as well as temporal features that offset various sources of noise (e.g., head motion, ambient illumination changes, etc.), 2) learn the relationships between associated physiological processes, 3) work in real-time to support

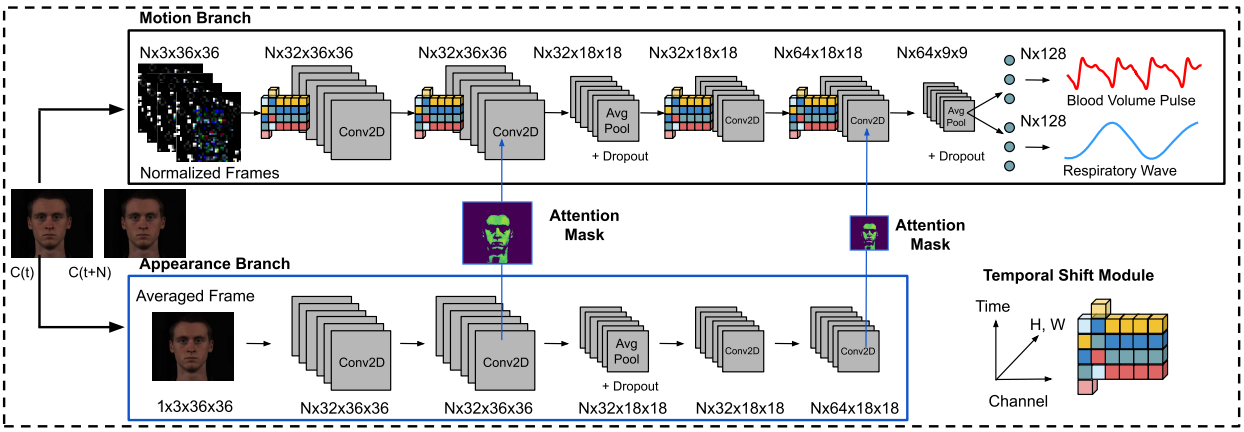

Figure 3.2: We present a multi-task temporal shift convolutional attention network for camera physiological measurement.

various telehealth deployments. Our solution is a novel temporal shift convolutional attention architecture (Fig. 3.1D) which we systematically compare to its variants (Fig. 3.1A-C) to illustrate its benefits.

Because of the strong performance shown in prior work [27], our architecture leverages a two-branch structure with a spatial attention module (Fig. 3.1A). One branch is used for motion modeling, and the other branch for extracting meaningful spatial (i.e., facial) features. However, it fails to capture temporal dependencies beyond consecutive frames and thus is still vulnerable to many sources of noise. Perhaps the simplest way to introduce a strong temporal dependency is a *3D-CAN* that leverages 3D convolutions to model temporal relationships (Fig. 3.1B) which is similar to the model used in [159] but adds an attention module. However, since 3D convolutions incur quadratic computational cost compared to 2D convolutions, it is not feasible to achieve real-time on-device performance using a primarily 3D architecture. Therefore, we present a *Hybrid-CAN* architecture that is more computationally efficient than a purely 3D model. *Hybrid-CAN* combines a 2D-CAN and a 3D-CAN to maintain temporal modeling while leveraging more efficient 2D convolutions where possible.

Since spatial position changes between adjacent frames are subtle, using 3D convolutions in the appearance branch is unnecessary. As Fig. 3.1C illustrates, the input of the appearance branch is a single frame generated by averaging N (window size) adjacent frames. Although Hybrid-CAN reduces computational cost significantly, the computational overhead from 3D convolutions in the motion branch is still not tolerable if we want to achieve real-time inference on low-end mobile platforms (i.e., ideally at least 60 FPS).

Therefore, we introduce *TS-CAN* to remove the 3D convolution operations from the architecture entirely while preserving spatial-temporal modeling. TS-CAN has two major additional components: the temporal shift module (TSM) [69] and the attention module. TSM performs tensor shifting before the tensor is fed into the convolutional layers as visualized in Fig.3.2. More specifically, TSM splits the input tensor into three chunks across the channel dimension. Then, it shifts the first chunk to the left by one place (advancing time by one frame) and shifts the second chunk to the right by one place (delaying time by one frame). Both shifting operations are along the temporal axis, and the third chunk remains unchanged. It is worth noting that tensor shifting does not add any additional parameters to the network, but does enable information exchange among neighbouring frames. We used TSM in the motion branch to mimic the effects of 3D convolution, while the appearance branch in the TS-CAN is the same as Hybrid-CAN and only takes a single averaged frame. By doing so, the model not only significantly reduces computational time by only calculating the attention mask once, but also captures most of the pixels that contain skin and reduces camera quantization error.

Attention on Temporal Shift: Given there are already many different sources of noise described in the previous section, naively shifting an input tensor in time will introduce extra temporal information to our representation. It is then important that we pay attention to the pixels with physiological signals or risk amplifying noise. Therefore, we propose inserting an attention module in TSM to minimize the negative effects introduced by tensor shifting as well as to enable the network to focus on the target signals. The spatial and temporal distribution of physiological signals are not uniform on human skin. Soft-attention masks can assign

higher weights to certain shifted pixels with stronger signals in intermediate representations from the convolutional operations. More concretely, our attention modules are the bridges between the appearance branch and the motion branch (See Fig. 3.2). Softmax attention masks are generated via 1×1 convolutions before pooling layers. The attention mask is calculated as in Equation 3.5 where k is the index of a layer, ω^k is the 1×1 convolution and followed by a sigmoid activate function $\sigma(\cdot)$. l_1 normalization was applied to soften the extreme values in the mask to make sure the network avoided pixel anomalies. Finally, we perform an element-wise product to the corresponding representation $\mathbb{X}^k$ from the motion branch.

$$\mathbb{X}^k \odot \frac{H_k W_k \cdot \sigma(\omega^k \mathbb{X}_\alpha^k + b^k)}{2 \parallel \sigma(\omega^k \mathbb{X}_\alpha^k + b^k) \parallel_1} \tag{3.1}$$

Multi-Task TS-CAN: We now have an efficient on-device architecture to predict physiological signals in real-time. However, we still have two independent networks, one for estimating the blood volume pulse and another for the respiration signals. Thus, the computational cost is doubled while preventing the possibility for information sharing across these related physiological processes. As we know that pulse and respiration are linked, we propose a multi-task variant of our network (see Fig. 3.2). This shrinks the computational budget by approximately 50% and the tasks of estimating BVP and respiration can share an intermediate representation. The loss function of this multi-task TS-CAN (MTTS-CAN) is described in Eqn. 3.2 where b(t) is the gold-standard BVP waveform and r(t) is gold-standard respiration waveform. b(t)' and r(t)' are the respective predictions from the model.

$$L = \frac{1}{T}\sum_{t=1}^{T}|b(t) - b(t)'| + \alpha\frac{1}{T}\sum_{t=1}^{T}|r(t) - r(t)'| \tag{3.2}$$

3.2.2 EfficientPhys: Enabling Simple, Fast and Accurate camera Vitals Measurement

Convolution-based EfficientPhys

To enable simple, fast and accurate real-time on-device camera vitals measurement, we propose a one-stop solution architecture that takes raw video frames as the input to the

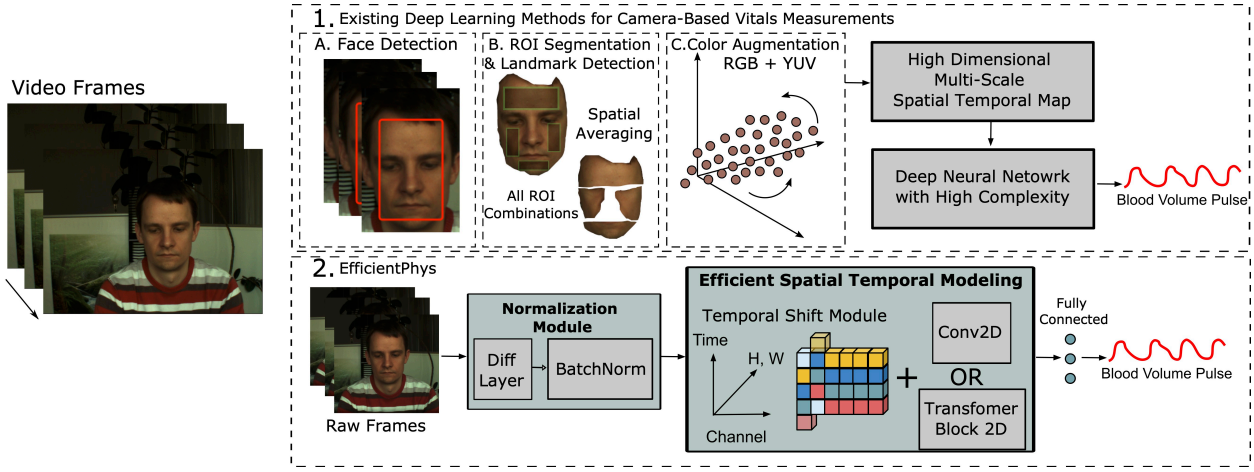

Figure 3.3: A high-level comparison of EfficientPhys and existing deep learning approaches for camera vitals measurement

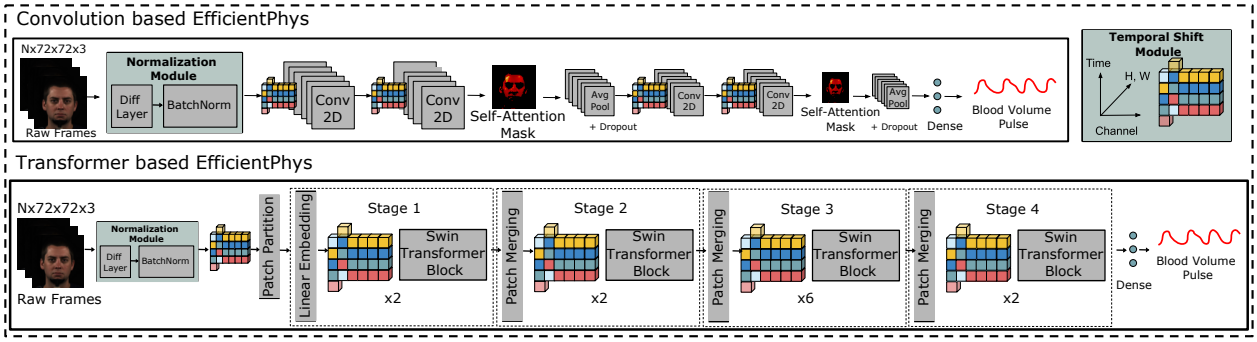

Figure 3.4: We present two novel architectures to enable simple, fast, and accurate camera vitals measurement: Convolution-based EfficientPhys and Transformer-based EfficientPhys. N is the number frames of video clip inputting to the network.

network and outputs a first-derivative PPG signal. The convolution-based EfficientPhys is a one-branch network that contains a custom normalization layer, self-attention module, tensor-shift module and 2D convolution operation to perform efficient and accurate spatial-temporal

modeling while making it simple to deploy.

Normalization Module. Existing neural methods all require different levels of preprocessing before providing the visual representation to the network to learn the underlying relationship between skin pixels and cardiac pulse signal. For instance, The state-of-the-art networks Dual-GAN [82] and CVD [98] proposed a hand-crafted spatial-temporal representations called STMaps. These preprocessed representations are generated for each video frame and includes steps of detecting 81 facial landmark points, extracting a set of region of interest (ROI) combinations ($2^n - 1$ where n is the number of ROIs, n=6) using these landmarks, and averaging pixel values in both the RGB and YUV color spaces, multiplying the 63 ROI combinations with the six channels. These modules not only add significant computational burden (Table 3.7 shows that Dual-GAN's preprocessing module takes 275ms per frame) but also make the system more challenging to implement and deploy on real-world computing systems such as mobile devices.

One of the goals of EfficientPhys is to remove these preprocessing modules completely and provide a one-stop solution. To achieve such simplicity and deployability, we propose a custom normalization module, which can perform motion modeling between every two consecutive RGB raw frames and normalization to reduce the lighting and motion noise. More specifically, the proposed normalization module includes a difference layer and a batchnorm layer. The difference layer (e.g., torch.diff) computes the first forward difference along the temporal axis of the raw video frames, by subtracting every two adjacent frames. Performing motion modeling between every two consecutive frames and normalization is more like a high-pass filtering and can help reduce the global noise from lighting and motion noise, while maintaining the subtle changes from PPG. To provide optical basis in our work, equation 3.3 illustrates the optical grounding of difference frame where $\boldsymbol{D}_k(t)$ of every two consecutive frames. $I(t)$ is the luminance intensity which is modulated by the specular reflection $\boldsymbol{v}_s(t)$ and the diffuse reflection $\boldsymbol{v}_d(t)$ as well as the optical sensor's quantization noise $\boldsymbol{v}_n(t)$.

$$\begin{aligned}\boldsymbol{D}_k(t) = (I(t) \cdot (\boldsymbol{v}_s(t) + \boldsymbol{v}_d(t)) + \boldsymbol{v}_n(t)) - (I(t-1) \\ \cdot(\boldsymbol{v}_s(t-1) + \boldsymbol{v}_d(t-1)) + \boldsymbol{v}_n(t-1))\end{aligned} \tag{3.3}$$

However, difference frames could be dramatically different in scale and make it hard for the network to learn meaningful feature representations, especially when the signal of interest is hidden in subtle pixel changes along the temporal axis and noise artifacts can cause significantly larger relative changes. To address this, we add a batch-normalization (batchnorm) layer following the difference layer. Adding a batchnorm layer provides two benefits: 1) it normalizes the difference frames to the same scale within the batch during training, 2) unlike fixed normalization in previous work [27, 70], batchnorm provides two learnable parameters β and γ for scaling (to a different variance) and shifting (to a different mean) and two non-trainable parameters which are the mean μ and the standard deviation σ. Through the learning process, the batchnorm layer can learn the best parameters for amplifying the pixel changes while minimizing the noise as Equation 3.4 and Fig.3.5 show. Without a batchnorm layer, directly applying a difference layer means the frames appear "black"; because the subtle changes of skin pixels in every two consecutive frames are relatively very small. On the other hand, adding a follow-up batchnorm layer will help it learn the normalization function to magnify the subtle changes of skin pixels substantially. The result is not simply a magnification of values but a normalization and magnification. Moreover, we also compare the output batchnorm layer to the hand-crafted normalized frame as shown in Fig.3.5. The output of batchnorm layer contains more information and qualitative analysis suggests it should be a better tool for skin segmentation after the learning process.

$$\boldsymbol{N}_k(t) = \frac{(\beta_t * \boldsymbol{D}_k(t) + \gamma_t) - \mu_{\boldsymbol{D}_k}}{\sigma_{\boldsymbol{D}_k}} \tag{3.4}$$

Self-Attention-Shifted Network. To efficiently capture the rich spatial-temporal information, we propose a self-attention-shifted network (SASN). SASN is built on top of the previous state-of-the-art method for on-device spatial-temporal modeling in optical cardiac

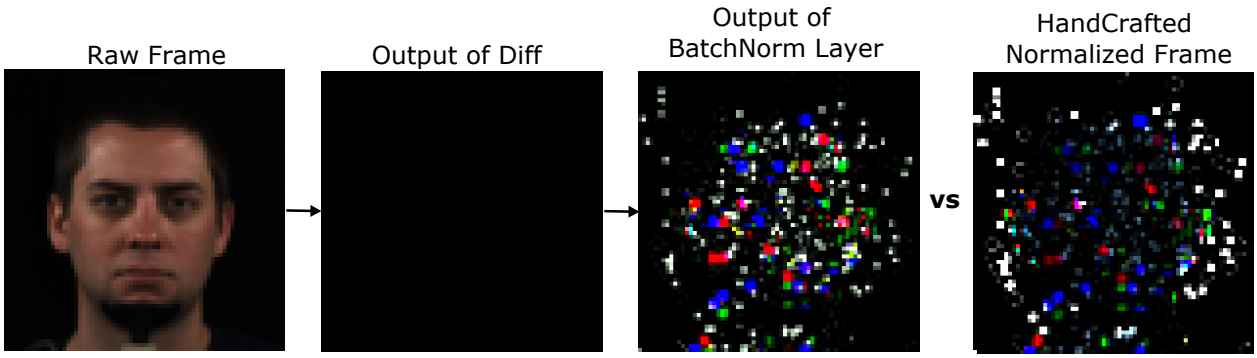

Figure 3.5: Outputs of diff and batchnorm layers and comparison with normalized frames generated via the hand-crafted process in prior work [27]. The output from the diff layer is almost black because the difference in skin pixels of consecutive frames is very subtle.

measurement - temporal-shift convolutional attention network (TS-CAN) [70]. TS-CAN has two convolutional branches, one of which takes a preprocessed difference frame representation and one of which takes a normalized appearance frame. The motion branch performs the main spatial-temporal modeling and estimation, and the appearance branch provides attention masks to guide the motion branch to better isolate the pixels of interest (e.g., skin pixels). However, we argue that the attention masks do not have to be obtained through a separate appearance branch and they can be also learned with a single branch end-to-end network. As Fig. 3.4 illustrates, our proposed self-attention-shifted network starts with the custom normalization module discussed in the previous section and then continues with two tensor-shifted convolutional operations. After the second and fourth tensor-shifted 2D convolutional layers, we add a self-attention module to help the network minimize the negative effects introduced by temporal shifting as well as motion and lighting noise. The self-attention layers are softmax attention layers with 1D convolutions followed by a sigmoid activation function. Then, normalization is applied to remove the outlier values in the attention mask, and the final normalized attention mask is element-wise multiplied with the output from the

tensor-shifted convolution. Equation 3.5 summarizes how our self-attention mechanism works where $ts(.)$ denotes temporal shift operation, ω_c^t denotes the 2D convolutional kernel followed by the temporal shift module, σ is the sigmoid activation and ω_a^t is the 1×1 convolutional kernel for self attention.

$$(\omega_c^t ts(\boldsymbol{N}_k(t)) + b_c^t) \odot \frac{H_t W_t \cdot \sigma(\omega_a^t \mathbb{X}_\alpha^t + b_a^t)}{2 \parallel \sigma(\omega_a^t \mathbb{X}_\alpha^t + b_a^t) \parallel_1} \tag{3.5}$$

Transformer-based EfficientPhys

Efficient Spatial-Temporal Video Transformer. Due to the recent success of visual transformers for image and video understanding and the importance of attention mechanisms for this task [159, 27, 115, 70], we also present a visual transformer version of EfficientPhys. For this task, we need a visual transformer to learn both spatial and temporal representations. Several existing video-based visual transformers are based on 3D-embedding tokens and input all the frames into 3D encoder and spatial-temporal attention modules [5, 79]. However, the computational complexity makes these unfavourable for real-time efficient modeling on mobile devices. In the convolutional version we used tensor-shifted 2D convolutions which have been shown to achieve comparable performance as 3D convolutions [70]. Inspired by this, our proposed transformer-based EfficientPhys is based on a 2D visual transformer, Swin transformer [78], but with additional components that we will describe below.

Since the 2D Swin transformer is only able to learn spatial features that map raw RGB values to latent representations between a single frame and the target signal (pulse), it does not have ability to model temporal relationships beyond consecutive frames. One of the main contributions of the Swin transformer is the shifted window module which has linear computation complexity and allows cross-window connection by shifting the window partition and limiting self-attention computation to non-overlapping local windows. Inspired by the idea of shifting of spatial window partitions, we propose to add a tensor-shift module (TSM) [69] before every Swin transformer block to facilitate information exchange across the temporal axis. The TSM first splits the input tensor into three chunks, shifts the first

chunk to the left by one place (advancing time by one frame) and shifts the second chunk to the right by one place (delaying time by one frame). All shifting operations are along temporal axis and are performed before the tensor is fed into each transformer block as shown in Fig. 3.4. By adding the TSM module to the Swin transformer, the new transformer architecture now has the ability to perform efficient spatial-temporal modeling and attention by combining shifting window partitions spatially and shifting frames temporally. It is worth noting that TSM does not introduce any learnable parameters, thus the proposed transformer architecture has the same number of parameters as the original Swin transformer. Finally, to enable truly end-to-end inference and learning, we also add the same normalization module proposed in the convolution EfficientPhys to this architecture.

In summary, the transformer-based EfficientPhys is the first end-to-end transformer architecture for camera cardiac pulse measurement that leverages tensor-shift modules and window-partition shift modules to perform efficient spatial-temporal modeling and attention to learn the underlying physiological signal from skin pixels.

3.2.3 Experiments

We compare our methods to four approaches for pulse measurement: POS[148], CHROM[33], ICA[111], 2D-CAN[27], Dual-GAN [82], Pulse-GAN [128] and two for respiration measurement: 2D-CAN and ARM [135]. For MTTS-CAN, other than DeepPhys [27], we run our experiments using AFRL and MMSE-HR datasets. For EfficientPhys, We trained on AFRL and SCAMPS, and test on three popular benchmark datasets (UBFC, MMSE-HR and PURE) to evaluate the accuracy.

To calculate the performance metrics, we post-processed the outputs of all methods in the same way using a 2nd-order Butterworth filter (cut-off frequencies of 0.75 and 2.5 Hz for HR and 0.08 and 0.5 Hz for BR). For the AFRL data, we divided the dataset into 30-second windows with no overlap. For the MMSE-HR dataset we used a window size equal to the number of frames in each video. We then computed four standard metrics for each window: mean absolute error (MAE), root mean squared error (RMSE) and correlation (ρ)

in heart/breathing rate estimations and the corresponding BVP/respiration signal-to-noise ratio (SNR) [33]. Details of the calculation for these metrics, training code, architecture and the trained models are available in the supplementary material.

Implementation Details for MTTS-CAN: At a high-level all our proposed networks share a similar two-branch architecture. Each branch has four convolutional layers. There is an averaging pooling layer and dropout layer placed after the second and fourth convolutional layers as shown in Fig. 3.2. Different architectures in Fig. 3.1 require different convolutional operations (e.g., 3D-CAN requires 3D CNNs). To preprocess the input of the appearance branch, we downsample each frame $c(t)$ to 36×36, which balances maintaining spatial resolution quality and suppressing camera noise [151]. For the motion branch, we calculate normalized frames using every two adjacent frames as $(c(t+1) - c(t))/(c(t) + c(t+1))$. The normalized frames are less vulnerable to changes in brightness and skin appearance compared to the raw frames $c(t)$ and reduce the chance of over-fitting to certain datasets.

Our system is implemented in TensorFlow [3]. We trained our proposed MTTS-CAN architectures using the Adadelta optimizer [162] with a learning rate of 1.0, batch size of 32, kernel size of 3×3, pooling size of 2×2, and dropout rates of 0.25 and 0.5. The final model was chosen after the training converged (12 epochs on the respiration task and 24 epochs on the pulse task). We implemented 2D-CAN, 3D-CAN and Hybrid-CAN as baselines to compare against our proposed architectures. For the 3D and Hybrid models the training schema is similar to TS-CAN, but we use a kernel size of 3×3×3 and a pooling size of 2×2×2. We used a window size of 10 frames in all temporal models to provide a fair comparison for our proposed architectures. We picked $\alpha = 0.5$ for the multi-tasking loss function in the MTTS-CAN to force estimations of pulse and respiration treated equally (pulse and respiration signals were both normalized in amplitude).

Our proposed architectures were deployed on an open-source embedded system called Firefly-RK3399[1] for latency evaluation. This embedded system has two large Cortex-A72

[1] http://en.t-firefly.com/product/rk3399.html

cores and four small Cortex-A53 cores. Although RK3399 also has a mobile Mali GPU, we focus our evaluation on CPU such that our proposed end-to-end architecture can be generalized to any ARM based mobile platform and IoT device. In this work, we extend a deep learning compiler stack - TVM [24] to support the core temporal shift operation required for TS-CAN. TVM takes a high-level description of a function and generates highly optimized low-level code for a targeted device. More specifically, our TVM-based on-device system first converts a TensorFlow graph to a Relay graph [119] and complies the code to Firefly-RK3399 using LLVM. We take advantage of TVM's scheduling primitives to generate efficient low-level LLVM code that accelerates expensive operations such as 2D and 3D convolutions.

Implementation Details for EfficientPhys: We implemented both convolution-based and transformer-based EfficientPhys in PyTorch [106]. We used an AdamW optimizer to train both networks instead of Adam by introducing additional regularization to reduce the effects of over-fitting through weight decay [80]. The learning rate we used for Convolutional model was 0.001 while the rate for transformer model was 0.0001. Based on empirical studies, we used the mean squared error (MSE) loss for training the transformer models and negative Pearson loss [138] for the convolutional model. We trained both models for ten epochs with a fixed random seed. We implemented TS-CAN based on the open-sourced code [73, 70] and used the Deep Physiological Sensing Toolbox [76] for the experiments on the UBFC and PURE datasets. To calculate the performance metrics, we first applied a band-pass filter to the signal with a cutoff frequency of 0.75 and 2.5 Hz (45 beats/minute to 150 beats/minute). We then followed Dual-GAN's evaluation scheme using peak detection and FFT to get estimated heart rate on each video of UBFC and PURE datasets [81] and MetaPhys's evaluation scheme on MMSE [73]. We conducted video-level evaluation where we calculated an averaged heart rate for each single video. We calculated three standard metrics for each video: mean absolute error (MAE), root mean squared error (RMSE) and Pearson correlation (ρ) in heart rate estimations and the corresponding ground-truth heart rates from the blood volume pulse collected via contact oximeter sensor.

To explore the efficiency of different architectures on mobile devices, we also conducted

experiments on a quad-core Cortex-A72 Raspberry Pi 4B to evaluate the model's performance on an edge device. We performed inference 10 times to get a reliable averaged on-device inference latency for EfficientPhys and TS-CAN. Due to the lack of open-source implementation of Dual-GAN, we were only able to find the implementation of STMaps which is the preprocessing module of Dual-GAN. Thus, we only evaluated the on-device latency for the preprocessing module in Dual-GAN. We also evaluated the latency of POS, CHROM, and ICA as they are traditional signal processing methods and don't have a separate preprocessing module.

Table 3.1: Benchmark performance of pulse measurement on the AFRL [39] and MMSE-HR [163] datasets.

	Heart Rate								**Respiration Rate**			
	AFRL (All Tasks)				**MMSE-HR**				**AFRL (All Tasks)**			
Method	MAE	RMSE	ρ	SNR	MAE	RMSE	ρ	SNR	MAE	RMSE	ρ	SNR
MTTS-CAN (Ours)	1.46	3.71	0.94	8.59	3.17	6.11	0.91	1.98	2.25	4.40	0.44	18.6
TS-CAN (Ours)	1.54	4.04	0.92	8.75	4.53	10.4	0.72	1.74	2.30	4.51	0.41	19.2
3D-CAN	1.28	3.00	0.96	11.4	2.85	5.01	0.94	5.12	2.29	4.41	0.43	19.4
Hybrid-CAN	1.21	2.79	0.97	11.5	2.84	5.04	0.94	6.76	2.02	4.06	0.49	20.0
2D-CAN (DeepPhys)	2.32	5.82	0.85	6.23	4.72	8.68	0.82	2.06	2.86	5.16	0.34	16.3
POS	2.48	5.07	0.89	2.32	3.90	9.61	0.78	2.33				
CHROM	6.42	12.4	0.60	-4.83	3.74	8.11	0.82	1.90	Not Applicable			
ICA	4.36	7.84	0.77	3.64	5.44	12.00	0.66	3.03				
Poles	Not Applicable								3.68	5.52	0.29	-6.22

MAE = Mean Absolute Error in HR estimation, RMSE = Root Mean Squared Error in HR estimation, SNR = BVP Signal-to-Noise Ratio, ρ = Pearson Correlation in HR estimation, WMAE = Waveform MAE.

3.3 Results of Accuracy and On-Device Efficiency

Comparison with the State-of-the-Art: For the AFRL dataset all 25 participants were randomly divided into five folds of five participants each (same folds as in [27]). The learning

Table 3.2: Pulse and respiration measurement on the AFRL and MMSE-HR datasets.

	Heart Rate								Respiration Rate				
	AFRL (All Tasks)				MMSE-HR				AFRL (All Tasks)				Time
Method	MAE	RMSE	ρ	SNR	MAE	RMSE	ρ	SNR	MAE	RMSE	ρ	SNR	(ms)
MTTS-CAN	1.45	3.72	0.94	8.64	3.00	5.66	0.92	2.37	2.30	4.52	0.40	18.7	6
MT-Hyb.-CAN	1.15	2.69	0.97	10.2	3.43	6.98	0.88	4.70	2.17	4.24	0.45	19.1	13
TS-CAN	1.32	3.25	0.95	8.86	3.41	7.82	0.84	2.92	2.25	4.47	0.41	18.9	12
Hyb.-CAN	1.12	2.60	0.97	10.6	2.55	4.16	0.96	5.47	2.06	4.17	0.46	19.8	26
3D-CAN	1.18	2.83	0.97	10.5	2.78	5.08	0.94	4.73	2.31	4.42	0.44	19.3	48
2D-CAN [27]	2.32	5.82	0.85	6.23	4.72	8.68	0.82	2.06	2.86	5.16	0.34	16.3	20
POS [148]	2.48	5.07	0.89	2.32	3.90	9.61	0.78	2.33					-
CHROM [33]	6.42	12.4	0.60	-4.83	3.74	8.11	0.82	1.90	Not Applicable				-
ICA [111]	4.36	7.84	0.77	3.64	5.44	12.00	0.66	3.03					-
ARM [135]	Not Applicable								3.68	5.52	0.29	-6.22	-

MAE = Mean Absolute Error, RMSE = Root Mean Squared Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

models were trained and tested via five-fold cross-validation using data from all tasks. The evaluation metrics are averaged over five folds and shown in Table 4.1. All of our proposed models outperform the 2D-CAN and other baselines. Hybrid-CAN and 3D-CAN achieve similar accuracy, reducing MAE by 50% on pulse and 20% on respiration measurement. The hybrid model has lower computational cost and is therefore preferable. TS-CAN also surpasses the 2D-CAN by more than 43% on pulse and 20% on respiration measurement. We also evaluated a multi-tasking version of TS-CAN and Hybrid-CAN, and call them MTTS-CAN and MT-Hybrid-CAN respectively. We observe that there is no accuracy benefit from the multi-tasking model variants relative to the single task versions because the network must use almost all the same parameters for both tasks. However, the MT models require half the computation and half as many parameters compared to running pulse and respiration models separately which is a considerable benefit.

In Table 3.4 and Table 3.5, we present results from our proposed EfficientPhys models and the current state-of-the-art neural and signal processing methods. The learning models

Table 3.3: Pulse and respiration measurement MAE on the AFRL by motion task.

	Heart Rate						Respiration Rate					
Method	T1	T2	T3	T4	T5	T6	T1	T2	T3	T4	T5	T6
MTTS-CAN	1.08	1.23	0.94	1.27	1.07	3.12	0.68	0.98	2.12	3.81	3.31	2.89
MT-Hybrid-CAN	1.04	1.24	0.95	1.23	0.88	1.53	0.77	0.89	2.23	3.28	3.03	2.80
TS-CAN	1.07	1.25	0.96	1.24	1.01	2.36	0.69	1.14	2.27	3.70	3.18	2.53
Hybrid-CAN	1.04	1.21	0.94	1.22	0.89	1.39	0.77	1.03	1.83	3.19	2.96	2.60
3D-CAN	1.06	1.19	0.92	1.23	0.89	1.77	0.96	0.98	2.58	3.80	2.87	2.65
2D-CAN [27]	1.08	1.21	1.02	1.43	2.15	7.05	1.25	1.11	3.35	4.63	3.77	3.08
POS [148]	1.50	1.53	1.50	1.84	2.05	6.11				|		
CHROM [33]	4.53	4.59	4.35	4.84	6.89	10.3				Not Applicable		
ICA [111]	1.17	1.70	1.70	4.00	5.22	11.8				|		
ARM [135]				Not Applicable			2.51	2.53	3.19	4.85	4.22	4.78

are all trained on the same datasets (AFRL + Synthetic) and tested on three dataset (UBFC, PURE and MMSE) to test if the model can generalize to videos with a different facial appearance, background, and lighting. To investigate how the depth of the network impacts the Transformer architecture, we created two version of Transformer-based EfficientPhys: T1 and T2. T1 uses the same depth as the Swin Transformer reported in [78] ([2, 2, 6, 2]). Each number indicates the number of Swin Transformer blocks as illustrated in Fig. 3.4. T2 is a more lightweight architecture to enable real-time on-device inference and has a depth of [2, 1]. EfficientPhys-C denotes the Convolution-based EfficientPhys as shown in the Fig.3.4. For UBFC and PURE, as Table 3.4 illustrates, EfficientPhys-C and EfficientPhys-T1 outperform all the existing methods. As Table 3.5 demonstrates, all the neural methods outperform the signal processing methods. EfficientPhys-T1 and TS-CAN achieved slightly better results than EfficientPhys-C and EfficientPhys-T2. Unfortunately, due to the lack of open source implementation or released models (e.g., Dual-GAN [82]), we could not successfully replicate

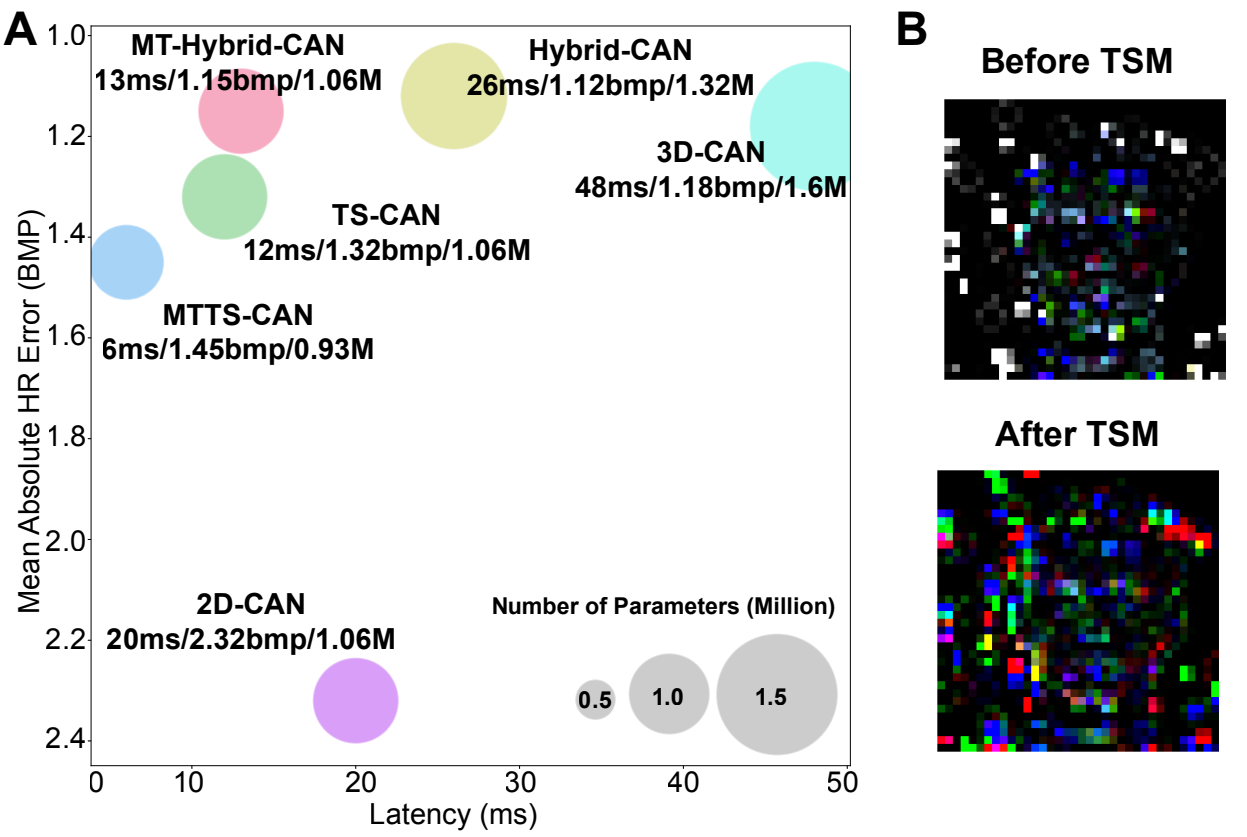

Figure 3.6: (A) On-Device latency evaluation across six models; (B) An visualization of TSM on a normalized frame from motion branch.

their complicated model architecture and conduct cross-dataset evaluation on this comparison.

Cross-Dataset Generalization:

To test whether our model can generalize to videos with a different resolution, background, and lighting, we trained our proposed models on the AFRL dataset and tested on the MMSE-HR dataset. Our proposed TS-CAN, Hybrid-CAN and 3D-CAN reduce errors by 25-50% compared to 2D-CAN (see Table 4.1). Furthermore, MTTS-CAN and MT-Hybrid-CAN both perform strongly, showing that it is possible to share the representations between pulse and respiration.

Table 3.4: Cross-dataset heart rate evaluation on UBFC and PURE (beats per minute).

	UBFC [12]				PURE [132]			
Method	MAE↓	MAPE↓	RMSE↓	ρ ↑	MAE↓	MAPE↓	RMSE↓	ρ ↑
EfficientPhys-C	1.14	1.16%	1.81	0.99	1.33	1.71%	5.99	0.97
EfficientPhys-T1	2.08	2.53%	4.91	0.96	1.11	1.30%	5.94	0.97
EfficientPhys-T2	3.07	3.41%	4.78	0.96	2.67	3.22%	9.08	0.92
TS-CAN[70]	1.70	1.99%	2.72	0.99	2.23	2.25%	3.71	0.98
POS[149]	3.52	3.36%	8.38	0.90	1.68	1.56%	9.60	0.92
CHROM[33]	3.10	3.83%	6.84	0.93	6.23	10.04%	17.18	0.71
ICA[111]	4.39	4.30%	11.60	0.82	5.70	5.69%	18.10	0.70

MAE = Mean Absolute Error in HR estimation, MAPE = Mean Absolute Error Percentage in HR estimation, RMSE = Root Mean Square Error in HR estimation, ρ = Pearson Correlation in HR estimation.

Table 3.5: Cross-dataset heart rate evaluation on MMSE (beats per minute).

	MMSE [163]			
Method	MAE↓	MAPE↓	RMSE↓	ρ ↑
EfficientPhys-C	3.48	4.02%	7.21	0.86
EfficientPhys-T1	3.04	3.91%	5.91	0.92
EfficientPhys-T2	3.51	3.96%	6.98	0.88
TS-CAN[70]	3.04	3.41%	6.55	0.89
POS[149]	3.79	4.28%	8.47	0.82
CHROM[33]	3.61	4.50%	7.43	0.85
ICA[111]	7.96	9.20%	14.02	0.51

MAE = Mean Absolute Error in HR estimation, MAPE = Mean Absolute Error Percentage in HR estimation, RMSE = Root Mean Square Error in HR estimation, ρ = Pearson Correlation in HR estimation.

To conduct a fair cross-dataset evaluation for EfficientPhys, we followed Dual-GAN [82] to train our models only on PURE and to test on UBFC as Table 3.6 shows. Although Dual-GAN outperforms all of other methods, we argue that the margin is relatively small as both

Table 3.6: Cross dataset evaluation with models trained on PURE only and tested on UBFC (beats per minute).

Method	MAE↓	MAPE↓	RMSE↓	ρ ↑
EfficientPhys-C	2.13	2.35 %	3.00	0.99
EfficientPhys-T1	3.83	4.32%	5.62	0.87
EfficientPhys-T2	3.97	4.35%	5.91	0.94
TS-CAN[70]	1.16	1.42%	2.78	0.99
Dual-GAN[82]	0.74	0.73%	1.02	0.99
PulseGAN[128]	2.09	2.23%	4.42	0.99

MAE = Mean Absolute Error in HR estimation, MAPE = Mean Absolute Error Percentage in HR estimation, RMSE = Root Mean Square Error in HR estimation, ρ = Pearson Correlation in HR estimation.

Dual-GAN and EfficientPhys-C achieve a Pearson correlation of 0.99. Moreover, according to American National Standards Institute (ANSI) and Consumer Technology Association's standard [6], MAPE of ± 5 is an acceptable error rate. Various studies have also used this standard to validate FDA approved sensors and systems [95, 123, 117, 19]. All the methods in Table 3.6 have met this recommended bar.

Computation Cost and Latency:

Fig. 3.6A and the last column of Table 4.1 show that MTTS-CAN and TS-CAN are the fastest architectures of those evaluated, taking 6 ms and 12 ms per frame for inference respectively. It is worth noting that TS-CAN is 40% faster than the 2D-CAN because the unique design of the appearance branch that only executes once and provides the generated attention mask to all the frames in the motion branch.

MT-Hybrid-CAN and Hybrid-CAN also achieve 13ms and 26ms inference times respectively, this is approximately double that of our TS-based methods due to the cost of 3D convolutions relative to 2D convolutions. The 2D-CAN not only has a higher latency compared to TS-CAN, but the accuracy is significantly lower. It is not surprising that the 3D-CAN achieved the worst inference speed because it has costly 3D convolutions in both branches. Latency is

Table 3.7: On-Device data preprocessing latency and model inference latency per frame (ms).

Method	Preprocessing (ms) ↓	Model (ms) ↓	Total (ms) ↓
EfficientPhys-C	0	40	40
EfficientPhys-T1	0	300	300
EfficientPhys-T2	0	40	40
TS-CAN[70]	3	60	63
Dual-GAN[82]	**275**	N/A	> 275
POS[149]	0	27	27
CHROM[33]	0	28	28
ICA[111]	0	31	31

ms = Preprocessing and model latency on Raspberry Pi 4B per frame.

important because we want our models to run at as high a frame rate as possible, 30 fps is the bare minimum required to accurately measure heart rate variability and subtle waveform dynamics and 100 fps would be preferable. Therefore, faster inference increases the precision at which we can measure inter-beat and systolic-diastolic intervals [88] and could help with non-invasive blood pressure measurement [51] and detecting arterial fibrillation (AFib) [20].

Fig. 3.7 and the Table 3.7 summarize the computational cost of the existing neural methods and proposed EfficientPhys models. Again, due to the lack of open source implementation and complex algorithm design, we were not able to replicate every architecture to benchmark its on-device latency. The results show that EfficientPhys-C only takes 40ms to process a single frame and it does not take any extra computational time to perform preprocessing. On the other hand, due to the complex model architecture and additional time for calculating hand-crafted normalized raw and difference frames, TS-CAN takes 63ms per frame. As mentioned earlier, Dual-GAN has a complicated preprocessing procedure for facial landmark detection, segmentation, color transformation and augmentation. We implemented this and

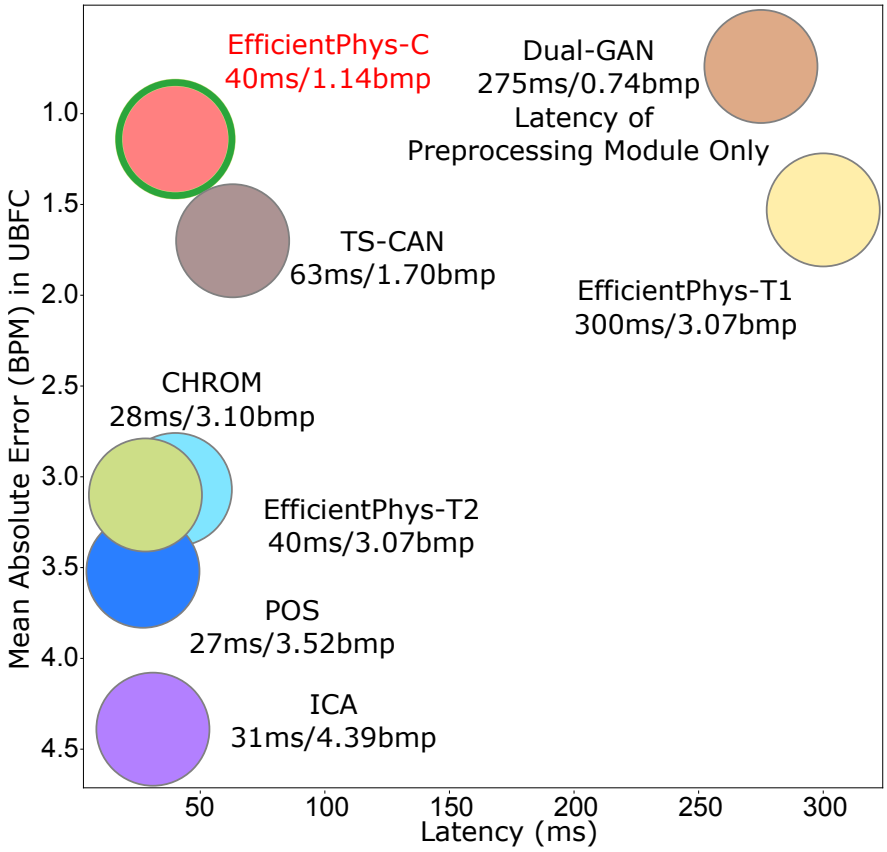

Figure 3.7: Accuracy-Latency Trade-off in eight different methods. Y-axis denotes the MAE error, and X-axis denotes the latency. The methods in the left-top corner have the best accuracy-latency Trade-off.

benchmarked the preprocessing module on our platform, and it took 275ms per frame, which is already 7x than the entire computational time of EfficientPhys-C. The estimation network in Dual-GAN also includes 12 2D convolution operations and numerous 1D convolution operations. Thus, we believe it would add a significant amount of computational time on top of the 275ms preprocessing time per frame. The default Transformer-based EfficientPhys (T1) has an unfavorable inference time due to its deep architecture design and takes 300ms to process every single frame. After reducing the depth to EfficientPhys-T2, it can achieve the same inference time as the EfficientPhys-C. However, EfficientPhys-T2 has the poorest

performance on all three benchmark datasets.

3.4 Discussion

Temporal Modeling for MTTS-CAN: Capturing such waveform dynamics requires good temporal modeling, therefore we compared several designs to help improve this. Our proposed MTTS-CAN, TS-CAN, MT-Hybrid-CAN, Hybrid-CAN and 3D-CAN all outperform the 2D-CAN and other baseline methods. This is consistent with prior work that found a 3D-CNN without attention outperformed a 2D-CNN (without attention) [159]. We would anticipate that the focus on modeling the temporal aspects of the physiological waveform would lead to greater resilience to noise. We perform a systematic evaluation on videos with varying velocities of angular (rotational) head motion. The results are shown in Table 3.3. As expected, all the proposed temporal models perform particularly strongly on tasks with greater velocity head motion; reducing the error on the most challenging task (6) by over 75%. Moreover, as Fig. 3.6B illustrates, although tensor shifting provides important temporal information, it also introduces extra noise. The results in Table 4.1 indicate that our attention module is effective at separating the signal from the added noise.

Multi-task Learning for MTTS-CAN: Comparing our MT models with the non-MT models, we observe that the MT models do not reduce the error in pulse and respiration rate estimates. But they do significantly improve the efficiency of inference as shown in Fig. 3.6A which is critical in resource constrained mobile platforms. Moreover, in order to estimate heart beat and respiration rate from a video, there is a number of mandatory pre-processing and post-processing steps to be included in the pipeline such as down-sampling images, computing averaged frames, calculating the number of peaks etc. Since MTTS-CAN only takes 6ms for inference on each fraem, even with the pre-processing overhead real-time inference is still eminently feasible. Also, memory is a valuable resource on edge devices, and MTTS-CAN only requires half of the memory to store the parameters compared to TS-CAN. We believe MTTS-CAN can be deployed and especially useful in resource constrained settings.

Applications of MTTS-CAN: The low latency and high accuracy of our system

opens the door for many other applications. For example, it could be used to improve the measurement of heart rate variability which is a measure of the variation in the time between each heartbeat. Tracking the subtle changes between consecutive heart beats requires low latency like that provided by MTTS-CAN. Contactless and on-device HRV tracking could enable numerous novel applications in mental health and personalized health. Besides health applications, MTTS-CAN is also potentially be applied to various computer vision tasks that require on-device computation such as activity recognition and video understanding.

Convolution vs. Transformer for EfficientPhys:

Although visual transformers have begun to achieve state-of-the-art performance in some vision tasks, it is not the case for the task of video-based vitals measurement. Based on the results shown in Table 3.4 and Table 3.5, Efficient-C outperforms both Efficient-T1 by 45% of MAE in UBFC and similar performance in MMSE and PURE, while Efficient-C is more than 7x faster in terms of latency. When we shrink the Transformer-based EfficientPhys to a similar complexity as Convolution-based EfficientPhys, the performance is significantly diminished. The errors from the lightweight Transformer-based EfficientPhys-T2 increased 48% of MAE in UBFC, 141% of MAE in PURE and 15% of MAE in MMSE. These results indicate a shallow transformer architecture struggles to model subtle changes of skin pixels in the video. These finding suggest two potential insights. First, further optimizations will be necessary for transformers to outperform, even relatively shallow, convolutional models in this domain, this is possibly especially true when there is not a large amount of high-quality data available. As previous studies have shown [37], Transformers usually require more pre-training samples to obtain state-of-the-art accuracy. Unfortunately, currently the amount of data in the field of camera vital measurement is limited compared to other visual tasks. Our experiments in Table 3.6 also support this hypothesis where EfficientPhys-C surpasses both EfficientPhys-T1 and T2 with training only on PURE. We believe synthetic data is one way to help address this issue. Second, the good accuracy-efficiency trade-off for visual transformer might not be scaled to on-device architectures without further work. Since many on-device neural networks require significantly less amount of computing resources to perform real-time

Table 3.8: Ablation study on EfficientPhys-C (Top) an EfficientPhys-T1 (Down). Models are trained only on PURE and tested on UBFC.

Self-Attention	Diff	BatchNorm	MAE
✓	✓	✓	2.13
✗	✓	✓	2.43
✓	✓	✗	16.06
✓	✗	✗	16.06

TSM	Normal. Module	MAE
✓	✓	3.83
✓	✗	16.10
✗	✓	11.52

operations, scaling the Transformer architecture down is not ideal as our experimental results of EfficientPhys-T2 have shown.

Ablation Study for EfficientPhys: We provide ablation studies on various parameters in EfficientPhys-C and EfficientPhys-T1 in Table 3.8. Without the self-attention module, MAE of EfficientPhys-C is increased by 14%. Without the Normalization Module, in both EfficientPhys-C and EfficientPhys-T1, the MAEs increased by 753% and 420%. As Figure 3.5 illustrates, the output of the difference layer contains almost black pixels and these results indicate that neural methods are sensitive to the magnitude of pixel values and whether they are zero centered. Finally, without the tensor shift module (TSM) in transformer-based models, the error increased by 300% which indicates TSM plays an important role in exchanging temporal information and dynamics.

Simplifying Last-Mile ML Deployment: Numerous real-world applications are driven by novel machine learning algorithms. However, deploying these algorithms on different computing platforms has been extremely challenging for various reasons. One of these is that researchers sometimes only pay attention to the accuracy of the model and ignore the complexity of the last-mile engineering efforts. In this paper, we address this important issue through our one-stop architecture that takes the unprocessed raw frames and directly outputs the desired signal. This elegant and simple design will not only reduce the burden

of engineering required for cross-platform implementations, but also will help the research community to replicate and reproduce results.

Extensible to Other Signal: Finally, as another potential upside of our end-to-end design and the low latency, we envision EfficientPhys being applied to various other video-based applications. Since the input of our model is raw frames, we believe EfficientPhys can be easily extended to other tasks such as video-based blood pressure measurement and video understanding & recognition etc. On the other hand, most of the baseline methods we compared with (e.g., Dual-GAN, PulseGAN) require many custom preprocessing operations for video-based measurement which are less useful in other applications.

3.5 Broader Impact

Non-contact camera vital sign monitoring has great potential as a tool for telehealth. Our proposed system can promote global health equity and make healthcare more accessible for those in rural areas or those who find it difficult to travel to clinics and hospitals in-person (perhaps because of age, mobility issues or care responsibilities). These needs are likely to be particularly acute in low-resource settings. Non-contact sensing has other potential benefits for measuring the vitals of infants who ideally would not have contact sensors attached to their delicate skin. Furthermore, due to the exceptionally fast inference speed, the computational budget required for our proposed system is minimal. Therefore, people who cannot afford high-end computing devices still will be able to access the technology. While low-cost, ubiquitous sensing democratizes physiological measurement, it presents other challenges. If measurement can be performed from only a video, what happens if we detect a health condition in an individual when analyzing a video for other purposes. When and how should that information be disclosed? If the system fails in a context where a person is in a remote location, it may lead them to panic. We contextualize our contributions within the scope of democratizing technology for social good and helping to reduce health disparities with advanced AI technology. However, we are aware that machine learning systems are biased and can propagate inequalities. Before technology such as that presented in this thesis

is ready for deployment we need to make sure that that is not the case.

It is also important to consider how such technology could be used by "bad actors" or applied with negligence and without sufficient forethought for the implications. Non-contact sensing could be used to measure personal physiological information without the knowledge of the subject. Law enforcement might be tempted to apply this in an attempt to detect individuals who appear "nervous" via signals such as an elevated heart rate or irregular breathing, or an employer may surreptitiously screen prospective employees for health conditions without their knowledge during an interview. These applications would set a very dangerous precedent and would be illegal in many cases. Just as is the case with traditional contact sensors, it must be made transparent when these methods are being used and subjects should be required to consent before physiological data is measured or recorded. There should be no penalty for individuals who decline to be measured. Ubiquitous sensing offers the ability to measure signals in more contexts, but that does not mean that this should necessarily be acceptable. Just because cameras may be able to measure these signals in a new context, or with less effort, it does not mean they should be subject to any less regulation than existing sensors, in fact quite the contrary.

In the United States, the Health Insurance Portability and Accountability Act (HIPAA) and the HIPAA Privacy Rule sets a standard for protecting sensitive patient data and there should be no exception with regard to camera sensing. In the case of videos there should be particular care in how videos are transferred, given that significant health data can be contained with the channel. That was one of the motivations for designing our methods to run on-device, as it can minimize the risks involved in data transfer.

3.6 Conclusions

Telehealth and the SARS-CoV-2 pandemic have acutely highlighted the specific need for accurate and computationally efficient cardiovascular and pulmonary sensing. We have presented a novel multi-task temporal shift convolutional attention network (MTTS-CAN) that improves on the state-of-the-art in both of these dimensions. On top of MTTS-CAN, we

present a novel end-to-end method called EfficientPhys to enable simple, fast, accurate camera contactless vitals measurement. We achieved strong performance with using significant less computational power. With the simple and elegant one-stop design, EfficientPhys also helps address the issue of last-time machine learning deployment and reduces health disparity.

Chapter 4

ALGORITHMS AND DATASETS FOR EQUITABLE CAMERA HEALTH SENSING

4.1 Introduction

While there are many compelling advantages of camera health sensing, the approach also presents unsolved challenges. The use of ambient illumination means camera measurement is sensitive to **environmental differences** in the intensity and composition of the incident light. Camera **sensor differences** mean that hardware can differ in sensitivity across the frequency spectrum and auto adjustments (e.g., white balancing) and video compression codecs can further impact pixel values [86]. People (the subjects) exhibit large **individual differences** in appearance (e.g., skin type, facial hair) and physiology (e.g, pulse dynamics). Finally, **contextual differences** mean that motions in a video at test time might be different from those seen in the training data. One specific example of challenges to generalization is biases in performance across skin types [99]. This problem is not isolated to physiological measurement as studies have found systematic biases in facial gender classification, with error rates up to 7x higher on women than men and poorer performance on people with darker skin types [18].

Collecting large corpora of high-quality physiological data presents challenges: 1) recruiting and instrumenting participants is often expensive and requires advanced technical expertise, 2) the data can reveal the identity of the subjects and/or sensitive health information meaning it is difficult for researchers to share such datasets. Therefore, training supervised models that generalize well across environments and subjects is often difficult. For these reasons, we observe that performance on cross-dataset evaluation is significantly worse than within-dataset evaluation using current state-of-the-art methods [27, 70].

Calibration of consumer health sensors is often performed in a clinic, where a doctor or nurse will collect readings from a high-end sensor to calibrate a consumer-level device the patient owns. The reason for this is partly due to the variability within readings from consumer devices across different individuals. Ideally, we would be able to train a personalized model for each individual; however, standard supervised learning training schemes require large amounts of labeled data. Getting enough physiological training data of each individual is difficult because it requires using medical-grade devices to provide reliable labels. Being able to generate a personalized model from a small amount of training samples would enable customization based on a few seconds or minutes of video captured while visiting a clinic where people have access to a gold-standard device. Furthermore, if this process could be achieved without the need for these devices (i.e., in an unsupervised manner), that would have even greater impact. Integrating remote physiological measurement into telehealth systems could provide patients' vital signs for clinicians during remote diagnosis. Given that requests for telehealth appointments have increased more than 10x during COVID-19, and that this is expected to continue into the future [125], robust personalized models are of growing importance.

In this chapter, we introduce a series of methods to help bridge the gaps of equitable camera health sensing. First, we introduce a novel unsupervised few-shot adaptation algorithm called MetaPhys in Section 4.2. MetaPhys is the first algorithm exploring few-shot adaption and personalization in person appearance and backgrounds. Second, building on top of MetaPhys, we introduce a dual-camera mobile system in Section 4.3 to provide high-quality pseudo labels for robust few-shot adaptation and personalization. Third, we explore a privacy-preserving federated learning algorithm in Section 4.4 with imperfect data to enable potential large-scale collaborative training without using users' raw data. Finally, we introduce a computer graphics based synthetics framework in Section 4.5 to generate diverse synthesized video data from different skin tones, backgrounds and lighting conditions with synchronized various vital signal.

4.2 Few-Shot Adaptation and Personalization

4.2.1 Introduction

Meta-learning, or learning to learn, has been extensively studied in the past few years [52]. Instead of learning a specific generalized mapping, the goal of meta-learning is to design a model that can adapt to a new task or context with a small amount of data. Due to the inherent ability for fast adaption, meta-learning is a good candidate strategy for building personalized models (e.g., personalization in dialogue and video retargeting [83, 65].) However, we argue that meta-learning is underused in healthcare. The goal of this work is to develop a meta-learning based personalization framework in remote physiological measurement whereby we can use a limited amount of data from a previously unseen individual (task) to mimic how a clinician might manually calibrate sensor readings for a specific patient. When meta-learning is applied to remote physiological measurement, there are two kinds of scenarios: 1) supervised adaptation with a few samples of labeled data from a clinical grade sensor and 2) unsupervised adaptation with unlabeled data. We hypothesize that supervised adaptation is more likely to yield a robust personalized model with only a few labels, while unsupervised adaptation may personalize the model less effectively but require much lower effort and complexity in practice.

In this thesis, we propose a novel meta-learning approach to address the aforementioned challenges called MetaPhys. Our contributions are: 1) A meta-learning based deep neural framework, supporting both **supervised and unsupervised** few-shot adaptation, for camera vital sign measurement; 2) A systematic cross-dataset evaluation showing that our system considerably outperforms the state-of-the-art (42% to 52% reduction in heart rate error); 3) To perform an ablation experiment, freezing weights in the temporal and appearance branches to test sensitivity during adaptation; 4) To analyze performance for subjects with different skin types. To our best knowledge, MetaPhys is the first work that leverages pseudo labels in training a physiological sensing model and the first unsupervised deep learning method in remote physiological measurement. Our code, example models, and video results can be

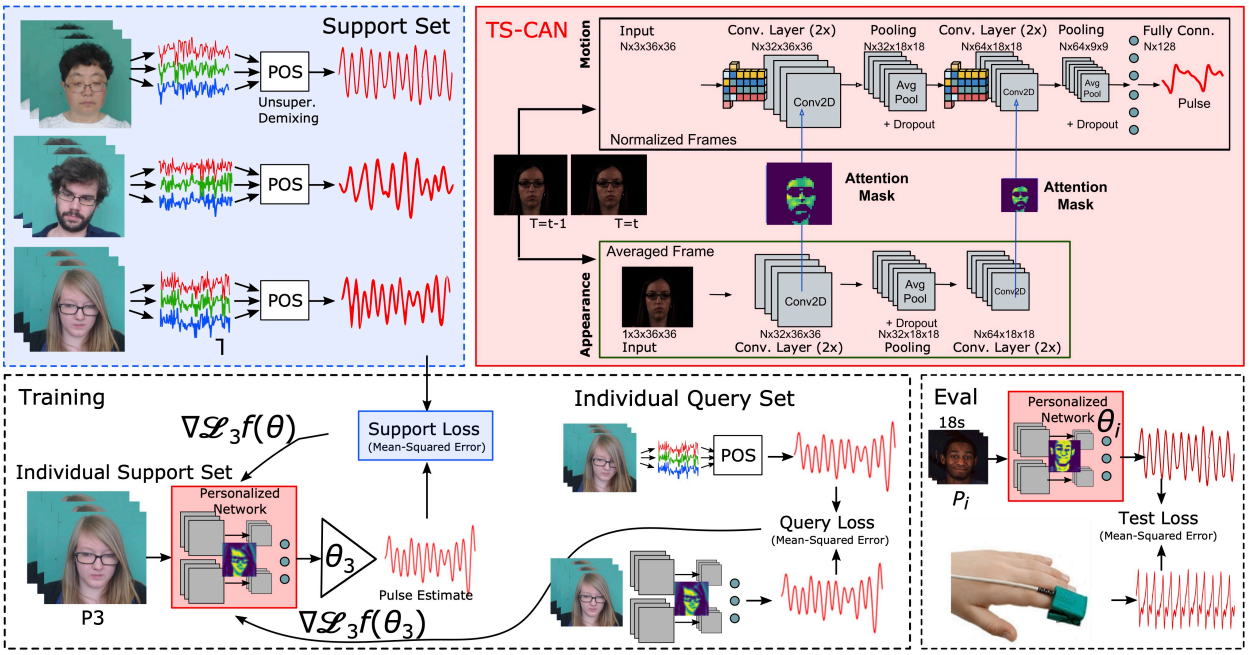

Figure 4.1: We present MetaPhys, an approach for few-shot unsupervised adaptation for personalized camera physiological measurement models.

found on our project page.[1].

4.2.2 Method

Physiological Meta-Learning

In camera cardiac measurement, the goal is to separate pixel changes due to pulsatile variations in blood volume and body motions from other changes that are not related to the pulse signal. Examples of "noise" in this context that might impact the performance on the task include: changes in the environment (illumination) and changes in appearance of the subject and motions (e.g., facial expressions, rigid head motions). A model trained within a traditional

[1] https://github.com/xliucs/MetaPhys

supervised learning regime might perform well if illumination, non-pulsatile motions, and facial appearances in the test set are similar to those in the training set. However, empirical evidence shows that performance usually significantly degrades from one dataset to another, suggesting that traditional training is likely to overfit to the training set to some extent [27]. Therefore, to achieve state-of-the-art performance in remote physiological measurement on cross-dataset evaluation, the system should have: 1) a good initial representation of the mapping from the raw video data to the pulse signal, and 2) a strategy for adapting to unseen individuals and environments.

To achieve this, we propose a system called MetaPhys (Fig. 4.1), an adaptable meta-learning based on-device framework aimed at efficient and personalized remote physiological sensing. MetaPhys uses a pretrained convolutional attention network as the backbone (described below) and leverages a novel personalized meta-learning schema to overcome the aforementioned limitations. We adopt Model-Agnostic Meta-Learning (MAML) [41] as our personalized parameter update schema. MAML produces a general initialization as the starting point for fast adaptation to a diverse set of unseen tasks with only a few training samples. However, applying MAML to the task of camera physiological measurement has differences to many previously explored meta-learning problems. Existing meta-learning approaches are often evaluated on classification or some toy regression tasks due to the lack of regression benchmark datasets [52]. Our problem is a non-trivial vision-based regression task due to the subtle nature of the underlying physiological signal. Algorithm 1 outlines the training process for MetaPhys, we first pretrain the backbone network to get an initial spatial-temporal representation. Then we treat each individual as a task τ_i. At training time, we split the data into a support set (K video frames) and a query set (K' video frames) for each individual (task). The support set is used to update the task's parameters and yield a personalized model θ_i. The query set is used to assess the effectiveness of the personalized model and further update the global initialization θ to make future adaptation better. A robust personalized model θ_i aims to provide a more accurate attention mask to the corresponding motion branch and to preform precise physiological measurement for the

target individual as well as the target's environment. During the testing stage, MetaPhys has the updated global initialization $\hat{\theta}$, and can generate $\hat{\theta}_i$ for each test individual (task) by optimizing the test support set as $\hat{\theta_{\tau_i}} \leftarrow \hat{\theta} - \alpha \nabla_{\hat{\theta}} \mathcal{L}_{\tau_i} f(\hat{\theta})$. With this training and testing schema, the robust global initialization $\hat{\theta}$ generated from MetaPhys not only leverages the pretrained representation but also learns how to adapt to new individuals and environmental noise quickly.

Spatial and Temporal Model Architecture Backbone

Our ultimate goal is a computationally efficient on-device meta-learning framework that offers inference at 150 fps. Therefore, we adopt the state-of-the-art architecture (TS-CAN) [70] for remote cardiopulmonary monitoring. TS-CAN is an end-to-end neural architecture with appearance and motion branches. The input is a series of video frames and the output is the first-derivative of the pulse estimate at the corresponding time points. Tensor shifting modules (TSM) [69] are used that shift frames along the temporal axis allowing for information exchange across time. This helps capture temporal dependencies beyond consecutive frames. The appearance branch and attention mechanism help guide the motion branch to focus on regions with high pulsatile signal (e.g., skin) instead of others (e.g., clothes, hair) (see Fig. 4.1). However, we discover empirically that this network does not necessarily generalize well across datasets with differences in subjects, lighting, backgrounds and motions (see Table 4.1). One of the main challenges when employing TS-CAN is that the appearance branch may not generate an accurate mask while testing on unseen subjects or environments because of the differences in appearance of skin pixels. Without a good attention mask, motions from other sources are likely to be given more weight, thus damaging the quality of our physiological estimate.

Supervised or Unsupervised Learning

We explore both supervised and unsupervised training regimes for MetaPhys. Supervised personalization may be suitable in clinical settings that require highly precise adaptation and

where there is access to reference devices. Unsupervised personalization may be preferable for consumer measurement when convenience and scalability is of a greater priority and calibration with a clinical grade device might be difficult.

For the supervised version of MetaPhys we use the gold standard reference signal from a fingertip PPG or blood pressure wave (BPW) to train the meta-learner and perform few-shot adaptation when testing. In contrast to the supervised version, in the unsupervised case we use pseudo labels during the training of the MetaPhys meta-learner and parameter updates rather than the ground-truth signal from the medical-grade devices. We use a physiologically-based unsupervised remote physiological measurement model to generate pseudo pulse signal estimates without relying on gold standard measurements. More specifically, we leverage the Plane-Orthogonal-to-Skin (POS) [148] method, which is the current state-of-the-art for demixing in this context. POS calculates a projection plane orthogonal to the skin-tone, derived based on optical and physiological principles, that is then used for pulse extraction. The POS method can be summarized as follows: 1) spatially averaging pixel values for each frame with the region-of-interest, 2) temporally normalizing the resulting signals within a certain window size calculated relative to the frame rate, 3) applying a fixed matrix projection to offset the specular reflections and other noise, 4) band-pass filtering of the resulting pulse waveform.

We observe that even though our unsupervised model uses the POS signal for meta-training, MetaPhys's performance significantly outperforms POS once trained. As Algorithm 1 illustrates, the pseudo label generator G produces pseudo labels for both K support frames and K' query frames for adaptation and parameter updates. We used pseudo labels for the query set (K') at training time, as we observed similar empirical results in preliminary testing whether we used pseudo labels or ground-truth labels.

Algorithm 1 MetaPhys: Meta-learning for physiological signal personalization

Require: S: Subject-wise video data

Require: A batch of personalized tasks τ where each task τ_i contains N data points from S_i

Require: A pseudo label generator G for unsupervised meta-learning

$\theta \leftarrow$ **Pre-training** TS-CAN on AFRL dataset

for each $\tau_i \in \tau$ **do**

    if Supervised **then**

        $K \leftarrow$ Sample K support frames from videos of τ_i with ground truth labels

        $K' \leftarrow$ Sample K' query frames from videos of τ_i with ground truth labels

    else

        $K \leftarrow$ Sample K support frames from videos of τ_i with pseudo labels from G

        $K' \leftarrow$ Sample K' query frames from videos of τ_i with pseudo labels from G

    end if

    $\theta_{\tau_i} \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i} f(K, \theta)$, Update the personalized params. based on indiv. support loss

end for

$\hat{\theta} \leftarrow \theta - \beta \nabla_\theta \sum_{\tau_i} \mathcal{L}_{\tau_i} f(K'_{\tau_i}, \theta_{\tau_i})$, Update the global params. based on individuals' query loss

Table 4.1: Pulse Measurement (Heart Rate) on the MMSE-HR and UBFC datasets.

Method	Train / Test Datasets	MAE	RMSE	SNR	ρ
Pretrain + Unsupervised **MetaPhys**	(AFRL & UBFC) / MMSE	**1.87**	**3.12**	5.04	**0.89**
Pretrain + Supervised **MetaPhys**	(AFRL & UBFC) / MMSE	2.98	4.86	3.81	0.72
MetaPhys (No pretrain)	(AFRL & UBFC) / MMSE	3.67	5.50	2.41	0.70
Supervised Pretrain + FT on Test Support Set	(AFRL & UBFC) / MMSE	4.05	5.68	2.76	0.76
Supervised Pretrain [70]	(AFRL & UBFC) / MMSE	3.78	5.75	2.67	0.77
(Unsupervised) CHROM [33]	None / MMSE	3.2	5.71	5.42	0.75
(Unsupervised) POS [148]	None / MMSE	3.98	6.66	**5.74**	0.67
(Unsupervised) ICA [111]	None / MMSE	4.12	6.46	6.09	0.67

Method	Train / Test Datasets	MAE	RMSE	SNR	ρ
Pretrain + Unsupervised **MetaPhys**	(AFRL & MMSE) / UBFC	2.46	3.12	**4.28**	**0.96**
Pretrain + Supervised **MetaPhys**	(AFRL & MMSE) / UBFC	**1.90**	**2.62**	3.84	0.96
MetaPhys (No pretrain)	(AFRL & MMSE) / UBFC	3.80	5.32	0.13	0.84
Supervised Pretrain + FT on Test Support Set	(AFRL & MMSE) / UBFC	6.26	7.37	-0.23	0.72
(Unsupervised) Meta-rPPG [64]	Self-Collected / UBFC	5.97	7.42	-	0.53
Supervised Pretrain [70]	(AFRL & MMSE) / UBFC	4.42	6.13	1.87	0.79
(Unsupervised) POS [148]	None / UBFC	6.44	9.48	0.55	0.66
(Unsupervised) CHROM [33]	None / UBFC	7.31	9.85	0.93	0.57
(Unsupervised) ICA [111]	None / UBFC	10.2	14.4	-0.19	0.50

MAE = Mean Absolute Error, RMSE = Root Mean Squared Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

4.2.3 Experiments

Datasets

Implementation Details

MetaPhys was implemented in PyTorch [106], and all the experiments were conducted on a Nvidia 2080Ti GPU. We first implemented the backbone network (TS-CAN) and modified it to use a window size of 20 frames (rather than 10) because we empirically observed a larger window size led to better overall performance. Then, we implemented MetaPhys based

on a gradient computation framework called higher [45]. Compared with most previous meta-learning studies that were trained and evaluated on a single dataset (e.g., miniimagenet [147]), we used three datasets to perform pretraining and cross-dataset training and evaluation. Our backbone was pretrained on the AFRL dataset, and the training (described in Algorithm 1) and evaluation of our meta-learner were performed with the UBFC and MMSE datasets. We picked the size of the support set (K) for personalization to be 540 video frames for each individual. For a 30 fps video recording this equates to an 18-second recording which is a reasonably short calibration period. During the meta training and adaptation, we used an Adam optimizer [58] with an outer learning rate (β) of 0.001 and a stocastic gradient descent (SGD) optimizer with an inner learning rate (θ) of 0.005. We trained the meta-learner for 10 epochs, and performed one step adaptation (i.e., gradient descent).

As baselines, we implemented traditional supervised training (TS-CAN) on AFRL and evaluated on MMSE and UBFC. Fine tuning with the support set and testing on the query set was implemented as our adaptation baseline. To assure a fair comparison across all experiments, we forced the test data (test query set) to remain the same within each task. We also implemented three established unsupervised algorithms (CHROM, POS, ICA) using iPhys-Toolbox [85]. We applied post-processing to the outputs of all the methods in the same way. We first divided the remainder of the recordings for each participant into 360-frame windows (approximately 12 seconds), with no overlap, and applied a 2nd-order butterworth filter with a cutoff frequency of 0.75 and 2.5 Hz (these represent a realistic range of heart rates we would expect for adults). We then computed four metrics for each window: mean absolute error (MAE), root mean squared error (RMSE), signal-to-noise ratio (SNR) and correlation (ρ) in heart-rate estimations. Unlike most prior work which evaluated performance on whole videos (often 30 or 60 seconds worth of data), we perform evaluation on 12 second sequences which is considerably more challenging as the model has much less information for inference.

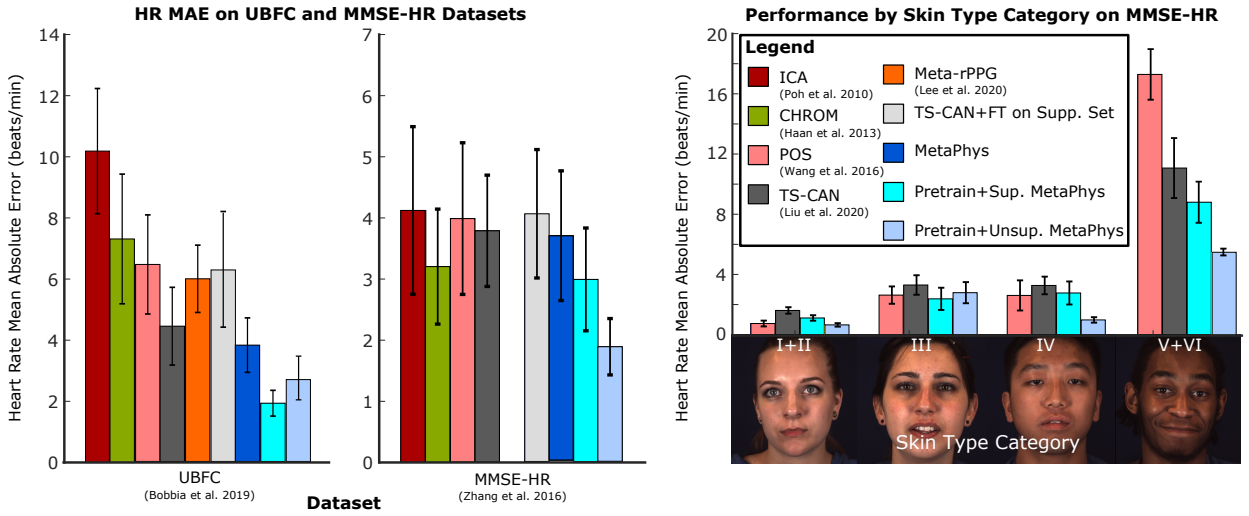

Figure 4.2: Left) MAE in HR estimates (12-second windows) for the UBFC and MMSE-HR datasets. Right) MAE in HR estimates by skin type on the MMSE-HR dataset. Standard error bars shown.

4.2.4 Results and Discussion

Comparison with the State-of-the-Art:

In this section, we compare the performance of MetaPhys with other state-of-the-art approaches using Mean Absolute Error(MAE) of heart rate. Details of the evaluation metrics are in the Appendix section. For the MMSE dataset, our proposed supervised and unsupervised MetaPhys with pretraining outperformed the state-of-the-art results by 7% and 42% in MAE, respectively (see Table 4.1). On the UBFC dataset, supervised and unsupervised MetaPhys with pretraining showed even greater benefits reducing error by 57% and 44% compared to the previous state-of-the-art, respectively. Meta-learning alone is not as effective as meta-learning using weights initialized in a pretraining stage (19% and 50% improvements in MMSE and UBFC). We also compared our methods against the only other meta-learning based method (Meta-rPPG) where we reduced the MAE by 68%. Furthermore, we compared MetaPhys

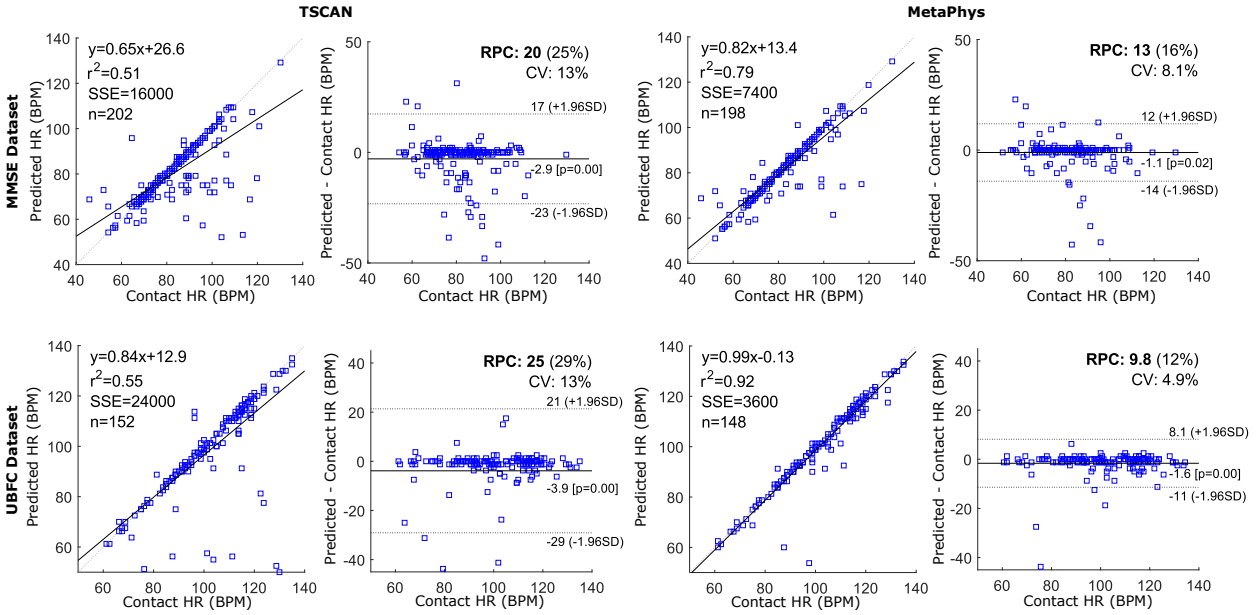

Figure 4.3: Left) Estimated HR and gold-standard HR reference measurements in MMSE and UBFC datasets and the corresponding Bland-Altman plots from TS-CAN [70]. Right) Estimated HR and gold-standard HR reference measurements in the MMSE and UBFC datasets and the corresponding Bland-Altman plots from MetaPhys.

against a traditional personalization method (fine-tuning), and our approach gained 54% and a 61% improvements in terms of MAE on MMSE and UBFC, respectively. We also evaluated performance using different time windows in the support set (6s, 12s and 18s), and the results showed training with 18s (RMSE: 3.12) outperformed 6s (RMSE: 5.43) and 12s (RMSE: 5.53) on the MMSE dataset. A similar trend was also observed on the UBFC dataset (RMSE of 18s: 3.12, RMSE of 12s: 4.48, RMSE of 6s: 3.46). Fig. 4.3 compares the heart rate estimates from MetaPhys and the-state-of-art algorithm (TS-CAN [70]) (y-axes) to the gold-standard measurements (x-axes). The strong correlation coefficients and bias/limits in the Bland-Altman plots presented support that MetaPhys can produce more accurate and reliable estimates. Our results from unsupervised MetaPhys indicate that pesudo labels

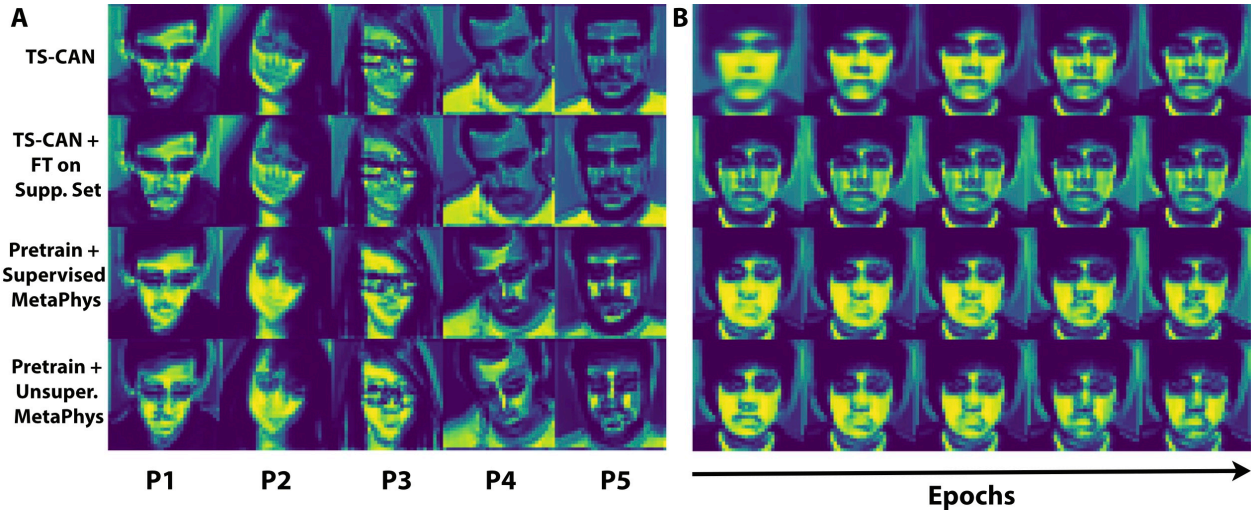

Figure 4.4: (A) An illustration comparing the attention masks of five subjects. The masks were generated in using four training schemes: 1) traditional supervised training (TS-CAN), 2) TS-CAN with fine tuning, 3) supervised MetaPhys and 4) unsupervised MetaPhys. (B) An illustration comparing the attention masks in the learning progress from four training schemes.

provided by a relatively simple de-mixing approach (POS) can be used in a meta-learning context to obtain strong results. The training with these pseudo labels, sometimes with noisy labels, well outperforms the de-mixing approach itself.

Unsupervised vs. Supervised Adaptation

Next, we examine the difference between a supervised and unsupervised training regime in MetaPhys. For UBFC, the *supervised* model (MAE=1.90 BPM), outperformed the *unsupervised* model (MAE=2.46 BPM). Whereas, for the MMSE dataset the *unsupervised* model (MAE=1.87 BMP) outperformed the *supervised* model (MAE=2.98 BMP). The fact that the unsupervised model achieves broadly comparable results to the supervised model is surprising and encouraging because there are many applications where unsupervised adaptation would be

more convenient and efficient (e.g., calibrating a heart rate measurement app on a smartphone without needing a reference device). We also observe that the unsupervised model, even though it used the POS signal as training input, significantly outperforms POS on both datasets, suggesting MetaPhys is able to form a better representation.

Visualizing Adaption

To help us understand why MetaPhys outperforms the state-of-the-art models we visualized the attention masks for different subjects. In Fig. 4.4-A, we compare the attention masks from the appearance branch of TS-CAN based on four training schemes which are: 1) supervised training with TS-CAN, 2) pretraining TS-CAN on AFRL and then fine tuning TS-CAN on the support set used for the meta-learning experiments, 3) pretraining on AFRL and supervised MetaPhys training, 4) pretraining on AFRL and unsupervised MetaPhys training. The differences are subtle, but on inspection we can notice that MetaPhys leads to masks that put higher weight on regions with greater pulsatile signal (e.g., forehead and cheeks) and less weight on less important regions (e.g., clothing - see P5 as an example). In Fig. 4.4-B, we visualize the progression of learning for the four different methods. Again the changes during learning are subtle, but the traditional supervised methods seem more likely to overfit even over a relatively small number of epochs meaning that the attention to important regions of the face is not as high as with the meta-learning approach, presumably because the traditional supervised learning has to capture a more generic model which is not well adapted to any one specific individual.

Freezing Appearance vs. Motion

We questioned whether the adaptation of the appearance mask was the main, or sole reason for the improvements provided by MetaPhys. To test this, we froze the weights in the motion branch of the TS-CAN during the meta-training stage and only updated weights in the appearance branch. From the results of these experiments, we observe that there is a 20% increase in MAE, indicating that MetaPhys not only noticeably improves the quality of the

attention mask, but also learns additional temporal dynamics specific to an individual's pulse waveform.

Robustness to Skin Type

Our motivation for adopting a meta-learning approach is to improve generalization. One challenge with camera PPG methods, and PPG methods in general, is their sensitivity to skin type. A larger melanin concentration in people with darker skin leads to higher light absorption compared to lighter skin types [99], thus reducing the reflectance signal-to-noise ratio. Fig. 4.2 shows a bar plot of the MAE in heart rate estimates by skin type (we group types I+II and V+VI as there were relatively few subjects in these categories). Both the AFRL and UBFC datasets are heavily skewed towards lighter Caucasian skin type categories. Therefore supervised methods trained on these datasets (e.g., TS-CAN) tend to overfit and not perform well on other skin types. Entirely unsupervised baselines do not perform any better, possibly because they were mostly designed and validated with lighter skin type data as well. While the highest errors for unsupervised MetaPhys still come in the darkest skin type categories, the reduction in error for types V+VI is considerable (68% compared to POS, 50% compared to TS-CAN). We are encouraged that these results are a step towards more consistent performance across people of different appearances.

Future Work

Federated Meta-Learning:

Federated learning (FL) refers to a training paradigm in which a global model is trained across multiple users by exchanging weights/gradients via a central server. In federated learning, a user's data will be stored locally reducing access to private information. For these reasons FL is attractive in the health domain [16, 40]. MetaPhys could deployed in a federated learning system as personalized weights and gradients of each individual could easily be uploaded to the central server without sharing images or videos of the subject. By

doing so, a more robust central model could be used as the backbone to further improve the generalizability and performance. In the future, we will explore how to extend MetaPhys in federated settings.

Explainable MetaPhys:

Although Figure 4.4 provides certain degree of insight as to why MetaPhys outperforms traditional fine tuning, it focuses on interpreting the spacial dimension and does not shed light on how well MetaPhys captures temporal information. PPG is a time-varying signal, therefore extracting unique temporal information in each individual is key. Towards more explainable remote physiological measurement, we plan to investigate why and how MetaPhys adapts to different faces and environments.

Limitations

There is a trend towards inferior performance when the skin type of the subject is darker. We acknowledge this limitation and plan to use resampling to help address this bias in future. Synthetic data may also be an answer to this problem and has shown promising early results [90]. Both the MMSE and UBFC datasets have somewhat limited head motion and future work will also investigate whether meta-learning can help with generalization to other motion conditions. In this paper, our proposed method is based on MAML [41], however, it is also worth to explore other other meta-learning algorithms. Moreover, we acknowledge that the datasets we evaluated on are non-clinical datasets. We plan to validate the robustness, feasibility and safety of our system in clinical adoptions.

4.3 Personalized Mobile System with Dual-Camera

4.3.1 Introduction

In traditional clinical settings, physicians often use high-end medical devices to help calibrate customer-level medical sensors for each patient. The procedure of calibration helps combat individual differences in sensor performance and strengthens the validity of the output. Therefore, training a personalized model for each individual in different environments is ideal. However, getting high-quality synchronized video and ground-truth physiological signals for training a personalized model is difficult. This is especially complicated if patients want to calibrate with their smartphones' cameras because external medical sensors are barely compatible with smartphones. A mobile system that performs self-calibration in camera contactless physiological sensing is attractive.

As deep learning methods struggle to generalize to unseen tasks and data, developing a personalized physiological sensing model using only a few unlabeled samples is promising. Encouraged by success on other tasks, we leverage meta-learning as the way of adapting our camera contactless PPG sensing algorithms. This work builds upon two specific examples of meta-learning applied to PPG measurement. Meta-rPPG [64] first introduced meta-learning for heart rate estimation. It achieves self-supervised weight adjustment by generating synthetic gradient and minimizing prototypical distance. MetaPhys [72] was then proposed is based on Model-Agnostic Meta-Learning (MAML) [41]. It took advantage of the advanced on-device network architecture [70] and probed into both supervised and unsupervised training regimes, both of which yielded satisfactory results. For supervised learning, ground-truth signal comes from medical-grade contact sensors, while in the unsupervised version, pseudo labels are used instead in the meta-learner training process. Though effective, these prior works rely much on synchronized video and ground truth obtained from medical-grade devices. However, it is difficult and laborious to collect a large-scale physiological dataset. In this work, we propose a mobile sensing system that leverages both front and rear cameras to generate contact PPG labels and personalize a camera contactless physiological system to address this

issue. We summarize the difference between popular recently published neural methods in camera contactless physiological sensing in Table 4.2.

In this work, we use contactless PPG measurement as an example to demonstrate how novel sensing systems can provide reliable pseudo physiological labels to meta learning algorithms for training few-shots adaption models. We propose a self-calibrating meta-learning system called MobilePhys, which leverages both front and rear cameras available on smartphones and the ability for us to measure physiological signals from multiple parts of the body. Specifically, we design a system that simultaneously measures the PPG signal from the finger-tip and the face of the subject during a calibration period to personalize a contactless measurement model that only relies on analyzing the face. The pseudo PPG labels generated from MobilePhys using the rear camera and the index finger can provide similar waveform quality to ground-truth PPG signals from medical-grade pulse oximeters. We demonstrate that this is also reliable in challenging real-world conditions (e.g., motion, lighting and darker skin types). Models customized or personalized using MobilePhys could then be deployed on the phone or shared with other smart devices (such as a laptop or smart mirror [110]) to enable convenient contactless measurement.

In summary, we propose a novel smartphone-based personalized physiological sensing system called MobilePhys, that leverages the back and front RGB camera to perform self-adaptation. More specifically, our contributions include:

- Proposing a novel mobile dual camera contactless physiological sensing system that generates high-quality pseudo PPG labels for few-shot personalization and adaptation.

- Demonstrating that we can leverage contact finger-tip PPG signal derived from the smartphone's rear camera to train a personalized camera contactless physiological neural network.

- Exploring and evaluating the performance of MobilePhys under different conditions such as different mobile devices, lighting conditions, motions, activities, skin types, and

camera settings through comprehensive user studies.

- Studying and investigating mobile camera settings, which we believe will be valuable for guiding future research in mobile physiological sensing.

- Finally, we collected and will release the first-ever multi-modality mobile camera contactless physiological dataset with different mobile devices, lighting conditions, motions, activities, and skin types. The documented dataset has gold standard oximeter recordings and synchronized finger PPG signals from the rear camera, and face videos along with other sensor signals (e.g., IMU, ambient light, etc.) from the smartphone. The dataset comprises close to six hours of video and sensor data from 39 participants.

Table 4.2: Comparison of State-of-the-Art Methods in camera Contactless Physiological Sensing

Method	**On-Device**	**Adaptation**	**Reliable Pseudo Label**
TS-CAN [70]	✓	✗	✗
Meta-rPPG [64]	✗	✓	✗
MetaPhys [72]	✓	✓	✗
MobilePhys (Ours)	✓	✓	✓

4.3.2 Method

In the following sections, we first describe the design of our sensing system for personalized camera contactless physiological sensing called MobilePhys. We then explore how to combine meta learning with MobilePhys to perform few-shot personalization without the need for clinical-grade sensors.

Mobile camera Contactless Physiological Sensing System

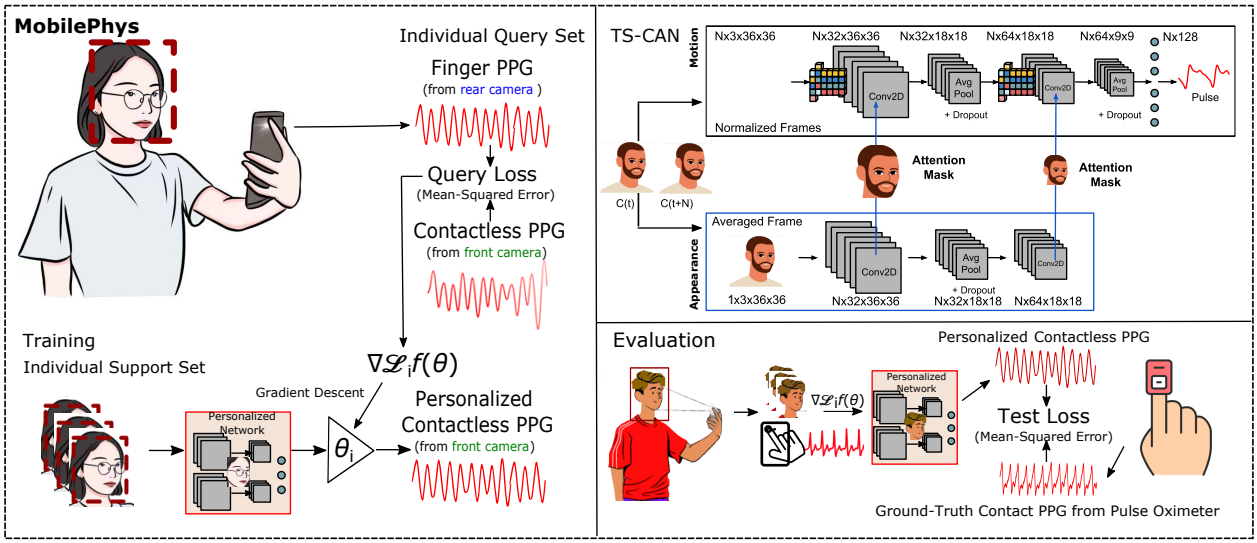

Figure 4.5: We present MobilePhys, a novel mobile camera contactless physiological sensing system that leverages rear camera to generate self-supervised "ground-truth" PPG label to help train a contactless and personalized physiological model

We propose a novel mobile dual-camera sensing system to perform personalized physiological measurement called MobilePhys (Figure 4.5). We leverage the fact that modern smartphones are equipped with at least two RGB cameras. The one on the front of the device is typically used for “FaceTiming” and selfies, and another one on the back of the device is used for photography and video recording. Thanks to significant investment in camera technology, these mobile cameras are typically high-quality optical sensors (e.g., low signal-to-noise ratio and high spatial resolution). Previous work has shown that the PPG signal and heart rate can be measured accurately from the fingertip and face using these cameras [30]. However, there are no examples that combine these sensors together to create a self-supervised system.

The basic principle behind both contact and non-contact PPG is that the volume of blood flow influences light absorption in the skin. The optimal frequency is approximately

540nm (where the absorption of light by hemoglobin is highest) [10]. Thus customized PPG sensors are usually designed with green light sensors (e.g., Apple Watch). Unfortunately, most smartphones are not equipped with green light sensors specifically designed for PPG measurement. However, the camera can be used as a proxy.

Our smartphone-based sensing system leverages the rear camera to provide a reference "ground-truth" PPG signal for personalizing a non-contact algorithm that measures the PPG directly from a video of the face. More specially, for a short calibration recording, we spatially average the red channel camera frames from the backward-facing (rear) camera while the participant has their finger over it. We turn on the built-in flashlight to increase the illumination. Simultaneously we capture frames of the participant's face using the front RGB camera. As 4.5 demonstrates, a person holds the smartphone while pressing their index finger on the rear camera. By using this dual-camera sensing system, it gives us perfectly synchronized contact and non-contact PPG data. Through using a few-shot meta learning algorithm (described in the next section), we then train a personalized non-contact PPG measurement algorithm (bootstrapping from a pretrained network). Each user only needs to hold the smartphone in this way for 18 seconds to create their own personalized physiological sensing model and, in the future can leverage that model without needing to place their finger on the rear camera.

Design and Implementation

While it may seem a simple concept, implementing MobilePhys was not trivial. We used a Xiaomi MI 8 for this project because it has a well-supported API for dual-camera development. To validate our method, we created a system to synchronize gold standard PPG measurements from a clinical-grade pulse oximeter, fingertip video recordings from the rear camera, and facial video from the front camera. Something that we observed was that many smartphone cameras struggled to maintain a consistent sampling rate while recording from both cameras and connecting a medical-grade pulse oximeter. This may in part be due to power consumption. To solve this limitation, we developed a customized router-based flask system in which we can connect the medical-grade pulse oximeter to a laptop while

synchronizing the external pulse oximeter and the data generated from our mobile app. More specifically, our smartphone app system provides a trigger signal to the laptop that is connected with the gold standard pulse oximeter when the mobile app starts recording. The laptop and the mobile phone were connected to a local network to minimize delay in communication. We viewed precise synchronization as an important part of our system so that the data can ultimately be used for training the camera contactless measurement algorithm that predicts the pulse waveform value for every video frame.

In building our system, we identified several important parameters that impacted the data quality. Smartphone video recording software has many automatic controls to make video recordings more visually pleasing. These include but are not limited to automatic exposure, white balance and sensitivity controls. We found that these controls on smartphones were more aggressive than those on other video capturing devices (e.g., DSLRs). This is perhaps because these devices are expected to operate with little user control in many contexts and lighting conditions. Therefore, as part of our analysis, we explored how these mobile camera controls affected the accuracy of camera contactless PPG measurement. Since camera contactless physiological measurement aims to capture very small changes of color changes from the skin, subtle noise can easily corrupt or interfere with the signal we aim to extract. Surprisingly, this type of systematic analysis is not often performed and we believe that characterizing the impact of these features will help other researchers who are building camera contactless physiological sensing systems.

To recover the PPG from the rear camera video, we used the shift method to decode the Android color integer format pixel data into four ARGB components [2], where A represents the Alpha (transparent) component, and R, G, B represents the Red, Green, and Blue components respectively. We also save the corresponding timestamp of the frame for subsequent data synchronization processing. During the measurement, the participants placed their index finger on the rear camera as Figure 4.7 illustrates. The finger PPG signal is the spatial

[2]https://developer.android.com/reference/android/graphics/Color#decoding

average of R-channel values with Max-min normalization from the frames collected by the rear camera. We believe that the contact finger-tip PPG signal is generally very accurate and close to those from the gold standard pulse oximeter.

Personalized Algorithm

camera contactless physiological sensing is sensitive to many types of noise, including motion noise, lighting noise (e.g., different ambient lighting), and appearance noise (e.g., different skin types). Due to the complexity of collecting physiological data and potential risks of leaking sensitive information, it is challenging to collect a large-scale camera contactless physiological dataset to perform traditional supervised training and train a generalizable model. Moreover, the issues of overfitting in neural network learning based methods have also been raised [27]. Past research [72] called MetaPhys has demonstrated that meta learning has a great potential in camera contactless physiological sensing. MetaPhys has shown that that combing meta learning with a 18s video from the target person can help generate a personalized model for a specific appearance (e.g., skin type). However, this paper did not demonstrate the use of personalization in different lighting conditions and motion tasks and in more ubiquitous and mobile settings.

In this paper, we adopt and create a variant of MetaPhys to enable few-shot personalization, described in Algorithm 1. Similar to MetaPhys, we also used Model-Agnostic Meta-Learning (MAML) [41] to update model parameters and create customized models with few-shot learning. The goal of MAML is to produce a robust initialization that enables faster and efficient (i.e., few shots or less data) learning on an unseen target task. Since our algorithm is based on MetaPhys, we apply a similar scheme to train our algorithm. Our algorithm starts with a pre-trained TS-CAN model [70] to initialize basic representation to model the spatial and temporal relationship between skin pixels and physiological signals. TS-CAN is a two-branch (appearance and motion) on-device neural network for contactless physiological sensing. The input to TS-CAN is a sequence of video frames (e.g., face) and the output of it is a sequence of physiological signals (first-derivative of the pulse signal). The appearance branch takes raw video frames and generates attention masks for the motion branch and helps

the motion branch to focus on the regions of interest containing physiological signals instead of unessential parts (e.g., hair, cloth). The motion branch leverages tensor shift module [69] to efficiently perform temporal and spatial modeling simultaneously and extract temporal relationships beyond consecutive frames.

Algorithm 2 MobilePhys Training: Meta-learning Algorithm for Mobile Physiological Signal Personalization

Require: S: Subject-wise video data

Require: A batch of personalized tasks τ where each task τ_i contains N video frames from the front camera and subject S_i

Require: A label generator G using rear PPG

1: $\theta \leftarrow$ **Pre-training** TS-CAN on AFRL dataset

2: **for** $\tau_i \in \tau$ **do**

3: $K \leftarrow$ Sample K support frames from videos of τ_i with rear-camera generated contact PPG labels

4: $K' \leftarrow$ Sample K' query frames from videos of τ_i with rear-camera generated contact PPG labels

5: $\theta_{\tau_i} \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i} f(K, \theta)$, Update the personalized params. based on indiv. support loss

6: **end for**

7: $\hat{\theta} \leftarrow \theta - \beta \nabla_\theta \sum_{\tau_i} \mathcal{L}_{\tau_i} f(K'_{\tau_i}, \theta_{\tau_i})$, Update the global params. based on individuals' query loss

Upon the TS-CAN backbone, we treat each individual as a task and split each individual's data (i.e., facial video data) into a support set K and a query set K'. After training with the support set, a personalized model is produced after parameter updates. However, most meta-learning applications assume the labels are available during training, which is not always the case in many machine learning applications. Our novel system, MobilePhys, can generate self-supervised high-quality "ground truth" PPG signal during the training of meta learning

algorithm and produce a personalized model for each unseen individual (task). Therefore, both K and K' come with self-supervised "ground truth" PPG signal labels from finger-tip contact PPG using the rear camera. Since the ultimate goal of meta learning is to generate a robust and efficient initialization, we then use the query set and the self-supervised labels to evaluate the performance of the personalized model θ_i and further optimize the global initialization accordingly. The details of the algorithm are described in Algorithm 1.

4.3.3 Data Collection

In this section, we describe our data collection study, including the apparatus, participant demographic, the user study design and procedure.

Figure 4.6 shows the apparatus used for data collection. To obtain the gold standard PPG signal, we used a finger pulse oximeter[3]. The pulse oximeter was connected to a desktop via USB. The raw PPG data were streamed to a desktop running custom python software via the UART protocol. The desktop hosted a back-end server for logging the raw video data from a Xiaomi 8 and an iPhone 11 via a WiFi router. By clicking a start button on a customized data collection application, the mobile app was triggered to simultaneously start the data collection. We recorded raw video from the front RGB camera, the true depth camera (iPhone 11), the ambient light sensor, the microphone, 9-axis inertial measurement unit (IMU), and the rear camera with flash on for camera contact gold standard finger PPG. In this work, we only use the RGB videos and pulse oximeter data; however, we anticipate that the other sensor data will be useful in future research and were recorded in our dataset.

We recruited a total of 39 participants (14 females and 25 males) in age of 20-35 (avg. = 27.1, s.d. = 4.1). Data were collected for 24 subjects using a Xiaomi 8 and for 15 subjects with an iPhone 11. Table **??** illustrates the distribution of gender and Fitzpatrick skin type [42] in our cross-device dataset, as well as the number of subjects who wore glasses, and/or makeup or had facial hair in the video recordings. All the participants are healthy adults and were

[3]HKG-07C+ infrared pulse sensor. http://hfhuake.com/

Figure 4.6: The hardware setup includes (A) an oximeter for gold standard contact PPG measurement, (B) a WiFi router for wirelessly streaming the smartphone's data to the desktop, (C) a desktop with the back-end server for collecting and synchronizing data, and (D) a Xiaomi 8 / iPhone 11 smartphone providing signals from multiple built-in sensors for physiological sensing.

recruited from a local university. It is worth noting that the second natural light condition is worse in the iPhone 11 data due to lower natural light intensity during the winter months, while the Xiaomi 8 data were collected in the spring/summer.

Table 4.3: The distribution of gender and Fitzpatrick skin type in our cross-device dataset.

Device	I+II	III+IV	V+VI	Female	Male	Glass	Makeup	Facial Hair	Total
Xiaomi Mi 8	8	14	3	7	17	7	5	11	24
iPhone 11	3	6	5	7	8	4	4	6	15
All	11	20	8	14	25	11	9	17	39

4.3.4 Experimental Design

In this study, we not only explore a personalization approach for adapting contactless PPG measurement models using a dual smartphone camera system MobilePhys, but also systematically investigate the effect of motion tasks and lighting conditions. Each participant

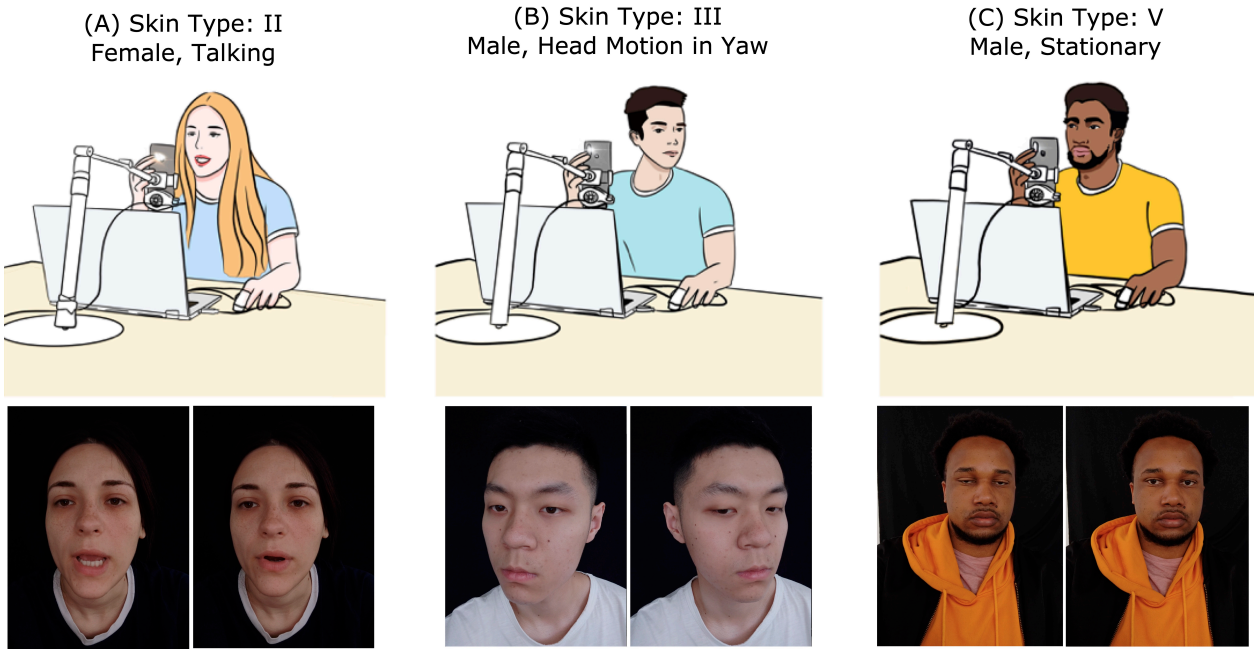

Figure 4.7: An illustration of some of the tasks in our data collection. We recruited subjects with different skin types and recorded the data under different motion tasks and lighting conditions. The head-shot images show video frames recorded by Xiaomi 8/iPhone 11's front RGB camera.

was also recorded before and after exercises. The details of our experiment design are summarized in Table 4.4 and main variables are discussed in the following:

- **Lighting Condition:** Natural sunlight, LED light, and incandescent light. These three common lights have significant spectrum differences, as Figure 4 shows.

- **Lighting Intensity:** To better investigate how light intensity impacts the performance of camera contactless physiological sensing, we also recorded data in three LED lighting intensities (bright(220 Lux)/moderate(110 Lux)/dim(55 Lux)) in the second batch experiment using an iPhone 11. By controlling the luminance value under LED lighting

conditions, we were able to mimic the light intensity in different scenarios (220 Lux - office lighting; 110 Lux - family living room lighting; 55 Lux - smartphone screen lighting).

- **Head Motion:** stationary, talking, head rotation in yaw direction, and random head motions as Figure 4.7 illustrates.

- **Exercise:** participants were instructed to raise their heart rate by conducting 30 seconds of exercise such as running.

- **Skin Type:** We recruited participants from different backgrounds and have different skin types. The participants are splitted into into three groups based on Fitzpatrick skin type: 1) I+II, 2) III+IV, and 3) V+VI.

The experiments were conducted in a conference room with a large window to ensure the availability of natural ambient light. The natural light could be blocked by closing a thick curtain. A desk was placed in the center of the conference room. Participants were asked to sit on the side facing the window. Another black curtain was used as the background screen.

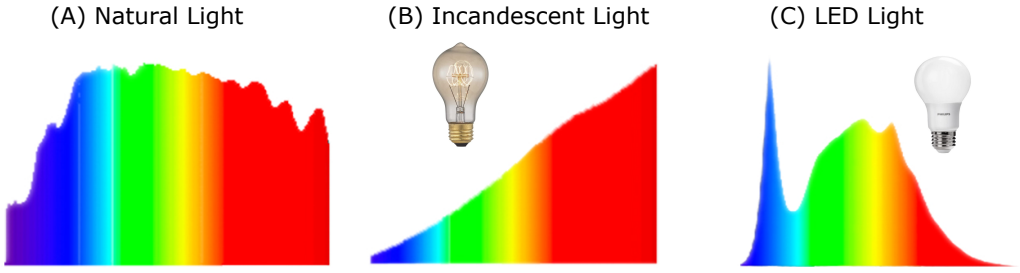

Figure 4.8: The light spectrum of the three lighting conditions. An incandescent light bulb and a LED lamp are illustrated in (B) and (C). Natural light is broader spectrum than both LED and incandescent illumination.

Every participant was welcomed into the experiment room and informed of the research goals and procedure. After reading and signing the consent form, each participant was instructed to place their left index finger into the pulse oximeter and their right index finger onto the smartphone's rear camera. The position of the smartphone was adjusted to ensure the front camera captured the participant's face. Then, to start each data recording period, the experimenter clicked a button on our customized Android mobile app. This started video recording from the front and rear cameras, the smartphone's sensors (e.g., IMU and microphone) and the contact pulse oximeter. Under the natural sunlight condition, the curtains were adjusted to ensure the facial video was not over dark or exposed. Under LED or Incandescent lighting conditions, curtains were drawn down to minimize the sunlight. A LED or incandescent lamp was used to simulate these two lighting scenarios. The distance between the participant and the lamp was carefully adjusted to alter the luminance value measured using a photometer at the participant's face. When participants performed head motions, they were asked to turn their heads at a uniform speed. To explore variable heart rate ranges, we also asked participants to conduct exercises to raise their heart rate on two trials. It took approximately 40 minutes to complete the recordings for each participant. Each participant received a 25 USD gift card.

Dataset Description In total, we collected 168 60-second front camera videos from Xiaomi Mi 8 and 200 front camera videos from iPhone 11. These videos were synchronized with the smartphone's rear-camera based contact PPG signal, true-depth camera signal (iPhone only), 9-axis motion signal (accelerometer, gyroscope, and magnetometer), front ambient light sensor data, and audio signal as well as the oximeter PPG signal. This cross-device dataset was collected to explore multiple methods to enable camera contactless physiological sensing using ubiquitous smartphone sensors. In this paper, we only use a subset of the dataset for contactless PPG measurement. Specifically, we utilized facial RGB videos, finger PPG signal and gold standard oximeter PPG signal to explore mobile personalization methods for contactless PPG measurement using the dual-camera setting of the commodity smartphone. However, because of the diversity and high quality of this dataset, researchers will have

Table 4.4: Details of experimental order under different conditions.

Trial No.	Exercise	Lighting	Motion	Duration (s)
1	NO	LED (220 Lux - iPhone only)	Stationary	60
2	NO	LED (110 Lux)	Stationary	60
3	NO	LED (55 Lux - iPhone only)	Stationary	60
4	NO	Incandescent	Stationary	60
5	NO	Natural Sunlight	Stationary	60
6	NO	Natural Sunlight	Random	60
7	NO	Natural Sunlight	Yaw Rotation	60
8	NO	Natural Sunlight	Talking	60
9	YES	Natural Sunlight	Stationary	60

opportunities to explore other interesting research questions. We plan to release this dataset with this paper.

4.3.5 Training and Evaluation

Implementation Details of MobilePhys

In order to train MobilePhys, we used two datasets: one for pre-training the backbone (TS-CAN), and the other for training the meta learner θ. We used a similar training regime as MetaPhys [72]. We first trained a TS-CAN backbone with the AFRL dataset [39]. Along with the AFRL dataset, we leveraged the UBFC [12] to train the initialization of our meta learner θ. It is worth noting that both AFRL and UBFC are recorded by a professional high-end camera, not a camera from a mobile device such as a smartphone. Neither of these datasets includes subjects with darker skin types (e.g., V, VI). All the data were recorded under one indoor lighting, and the subjects were stationary.

We implemented MobilePhys based on MetaPhys's open-sourced code [72]. MobilePhys

is also based on a gradient computation framework called higher [45]. We first trained our backbone network, TS-CAN, using the AFRL dataset. We then train the meta-learner using the UBFC dataset. In this meta-learner training stage, we considered each subject's dataset as a task and used 18 seconds of video data and PPG labels from the pulse oximeter (same as MetaPhys) as the support set K (see 4.3.2) for training a personalized model. The rest of the data in the task (i.e., subject) is considered as query data to further evaluate the personalized model and optimize the meta-learner through back-propagation. The output of this training process is a pre-trained meta-learner which only takes 18-second labeled data to generate a personalized model.

In the testing stage, as Algorithm 1 shows, we considered each experimental trial as a task (e.g., natural light + stationary + iPhone 11). The support set is the first 18-second frames recorded from the smartphone's front RGB camera (Xiaomi 8 or iPhone 11) and the ground-truth PPG labels τ_i were generated by its rear RGB camera. The output of this adaptation process for each task is a personalized model, which can be further used to evaluate the rest of the data within the task. An Adam optimizer [58] and a learning rate of 0.001 were used to optimize the outer loop of Algorithm 1. A stochastic gradient descent (SGD) optimizer with an inner learning rate (θ) of 0.003 was used to optimize the adaption stage. We trained 10 epochs for each experiment.

Baseline Implementation

Since prior work has shown that neural network based methods significantly outperform signal processing based demixing approaches. In this work, we only compare MobilePhys's performance against the state-of-the-art neural method - TS-CAN [70] and MetaPhys [72]. Our goal is to propose a mobile physiological sensing system; therefore, using models that run on-device is important. To the best of our knowledge, TS-CAN and MetaPhys are the best baseline papers that focus on on-device camera physiological sensing. Therefore, we chose TS-CAN and MetaPhys as our baselines.

For TS-CAN experiments, we trained TS-CAN with aggregated training datasets of AFRL and UBFC as the backbone network. In the testing stage of each task (same experiment trial

as the MobilePhys's testing stage), we then use the first 18 second's video data and label to fine tune the network and evaluate the rest of the data within the task.

For MetaPhys experiments, we used the same UBFC pretrained meta-learner described in the previous section. However, during the teasing stage, the label used for adaption in MetaPhys is pseudo labels generated by POS [148] according to MetaPhys. POS is an unsupervised signal processing method that computes a projection plane orthogonal to the skin type based on optical and physiological principles. The difference between MobilePhys and MetaPhys is the label used for few-shot adaption where our proposed MobilePhys is able to generate high-quality pseudo labels while MetaPhys relies on the unreliable pseudo label, which could be very noisy in challenging tasks (e.g., darker skin types, lighting, motion, etc.)

After getting the predicted PPG waveform from the network, we applied a band-pass filter to remove unessential noise. More superficially, we used a 2nd-order butterworth filter with a cutoff frequency of 0.75 and 2.5 Hz to keep the signal containing realistic adults' heart rate. During the evaluation stage, we compare the heart rate computed from filtered predicted PPG signal against the heart rate calculated from PPG labels recorded by a pulse oximeter (see Figure 4.5).

4.3.6 Results and Findings

Quantitative Results of MobilePhys

We compare the performance of MobilePhys with the state-of-the-art traditional supervised training algorithm (TS-CAN) and model-agnostic meta-learning algorithm (MetaPhys) using the data collected from Xiaomi 8 (Android) and iPhone 11 (iOS). The reported measurements include Mean-Square-Error (MAE), Signal-to-Noise Ratio (SNR) and Pearson Coefficient as metrics (ρ). In order to investigate how different motion tasks impact the performance of camera contactless physiological sensing systems, we first conducted experiments when the lighting condition was fixed. As Table 4.5 shows, under the natural light condition, we compared the performance of MobilePhys and baselines in three motion tasks: 1) head motion in yaw, 2) talking and 3) random head motion. On the head of motion in yaw task,

Table 4.5: Pulse Measurement (Heart Rate) on Different Motion Tasks.

	Experimental Conditions			**Xiaomi 8 / iPhone 11**		
Method	Exercise	Task	Lighting	MAE	SNR	ρ
MobilePhys	No	Head Motion in Yaw	Natural Light	**10.71**/**11.40**	**-8.73**/**-8.05**	**0.40**/**0.34**
TS-CAN	No	Head Motion in Yaw	Natural Light	12.04/12.89	-9.91/-9.46	0.06/0.33
MetaPhys	No	Head Motion in Yaw	Natural Light	13.35/13.93	-8.81/-12.77	0.37/0.21
MobilePhys	No	Talking	Natural Light	**3.38**/**6.33**	**-2.16**/**-7.61**	**0.70**/**0.59**
TS-CAN	No	Talking	Natural Light	11.61/9.38	-7.05/-10.53	0.04/0.33
MetaPhys	No	Talking	Natural Light	3.99/7.67	-3.24/-9.93	0.65/0.43
MobilePhys	No	Random Head Motion	Natural Light	**4.74**/**6.27**	**-6.47**/**-7.04**	**0.84**/**0.58**
TS-CAN	No	Random Head Motion	Natural Light	13.44/7.14	-11.9/-7.68	0.31/**0.58**
MetaPhys	No	Random Head Motion	Natural Light	8.92/6.98	-8.69/-8.25	0.33/0.56

MAE = Mean Absolute Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

Table 4.6: Pulse Measurement (Heart Rate) after Exercising.

	Experimental Conditions			**Xiaomi 8 / iPhone 11**		
Method	Exercise	Task	Lighting	MAE	SNR	ρ
MobilePhys	Yes	Stationary After Exercise	Natural Light	**3.49**/**7.68**	-2.67/**-7.43**	**0.85**/**0.79**
TS-CAN	Yes	Stationary After Exercise	Natural Light	8.67/10.83	-5.36/-9.75	0.44/0.53
MetaPhys	Yes	Stationary After Exercise	Natural Light	3.69/12.63	**-2.06**/-10.79	0.81/0.33

MAE = Mean Absolute Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

MobilePhys improved 11.3%/19.1% in MAE, 13.0%/18.0% in SNR, and 130.0%/22.2% in Pearson coefficient in average across all the subjects in two mobile devices when compared with TS-CAN and MetaPhys respectively. On the talking task, MobilePhys outperformed our baselines by 58.0%/16.5% in MAE, 49.3%/26.8% in SNR, and 334.0%/16.3% in Pearson coefficient, compared to our baseline methods. For the random head motion task, MobilePhys showed it could enhance the performance by 51.6%/34.8% in MAE, 34.9%/21.5% in SNR and 78.8%/76.8% in Pearson coefficient.

Furthermore, to help us understand whether the models work well for higher heart rates,

Table 4.7: Pulse Measurement (Heart Rate) on Different Lighting Conditions.

	Experimental Conditions			Xiaomi 8 / iPhone 11		
Method	Exercise	Task	Lighting	MAE	SNR	ρ
MobilePhys	No	Stationary	Natural Light	**0.97**/**4.27**	**1.54**/**-7.04**	**0.99**/**0.71**
TS-CAN	No	Stationary	Natural Light	4.32/7.14	-0.31/-7.68	0.49/0.58
MetaPhys	No	Stationary	Natural Light	1.49/6.98	0.24/-8.25	0.97/0.56
MobilePhys	No	Stationary	Incandescent	**0.73**/**5.50**	2.87/**-7.33**	**0.99**/**0.85**
TS-CAN	No	Stationary	Incandescent	1.05/10.72	1.63/-10.76	0.97/0.02
MetaPhys	No	Stationary	Incandescent	2.60/10.34	**3.06**/-9.91	0.73/0.43
MobilePhys	No	Stationary	LED	1.76/**3.05**	**2.85**/**-1.99**	0.96/**0.86**
TS-CAN	No	Stationary	LED	**1.63**/8.37	2.22/-5.07	**0.99**/0.09
MetaPhys	No	Stationary	LED	1.65/7.69	2.56/-7.58	0.96/0.54

MAE = Mean Absolute Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

Table 4.8: Pulse Measurement (Heart Rate) on Different Skin Types. All the participants are used to evaluate how skin types impact MobilePhys.

	Experimental Conditions			All Subjects		
Method	Skin	Task	Lighting	MAE	SNR	ρ
MobilePhys	I+II	Stationary	Natural Light	**1.51**	**0.16**	**0.98**
TS-CAN	I+II	Stationary	Natural Light	4.52	-1.92	0.60
MetaPhys	I+II	Stationary	Natural Light	2.19	-0.95	0.95
MobilePhys	III+IV	Stationary	Natural Light	**1.04**	**0.09**	**0.99**
TS-CAN	III+IV	Stationary	Natural Light	2.69	-1.18	0.91
MetaPhys	III+IV	Stationary	Natural Light	1.91	-1.04	0.88
MobilePhys	V+VI	Stationary	Natural Light	**2.31**	**-7.30**	**0.98**
TS-CAN	V+VI	Stationary	Natural Light	8.45	-8.63	0.68
MetaPhys	V+VI	Stationary	Natural Light	17.31	-8.65	0.42

MAE = Mean Absolute Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

we also evaluated MobilePhys on the recordings collected immediately after the participants conducted one minute exercise (e.g., running). As Table 4.6 shows, MobilePhys demonstrates

Table 4.9: Pulse Measurement (Heart Rate) on Different LED Light Intensities. Only participants with iPhone 11 were enrolled in light intensity studies.

	Experimental Conditions			**iPhone Subjects**		
Method	Light Intensity	Task	Lighting	MAE	SNR	ρ
MobilePhys	55 Lux	Stationary	LED	**4.58**	**-6.24**	**0.87**
TS-CAN	55 Lux	Stationary	LED	6.78	-7.91	0.14
MetaPhys	55 Lux	Stationary	LED	6.98	-9.40	0.80
MobilePhys	110 Lux	Stationary	LED	**3.27**	-3.03	**0.81**
TS-CAN	110 Lux	Stationary	LED	6.28	-4.79	0.36
MetaPhys	110 Lux	Stationary	LED	6.26	**-2.55**	0.12
MobilePhys	220 Lux	Stationary	LED	1.93	-1.59	0.97
TS-CAN	220 Lux	Stationary	LED	**1.64**	**-1.38**	**0.98**
MetaPhys	220 Lux	Stationary	LED	4.94	-5.59	0.65

MAE = Mean Absolute Error, ρ = Pearson Correlation, SNR = BVP Signal-to-Noise Ratio.

superior performance compared to TS-CAN and MetaPhys where it can reduce the MAE by 46.3%/28.4%. The SNR and Pearson Coefficient are also improved by 36.1%/20.0% and 74.2%/32.2%, respectively.

Next, we explored the effects of lighting conditions that potentially could have a large impact on the performance of mobile camera contactless physiological sensing systems. We conducted three sets of experiments to examine three different lighting conditions: 1) natural sunlight, 2) incandescent light, and 3) LED light. All the experiments were conducted on the video recordings collected while the participants were stationary to exclude confounds from motion. As Table 4.7 illustrates, in the natural light condition, MobilePhys improves 58.6%/37.8% in MAE, 44.0%/41.8% in SNR and 68.2%/8.6% in pearson coefficient. In the incandescent light condition, MobilePhys enhances 46.2%/54.0% in MAE, 66.4%/45.4% in SNR and 54.8%/52.3% in pearson coefficient. In the LED light condition, MobilePhys enhances 46.6%/43.2% in MAE, 269.3%/173.8% in SNR and 43.1%/15.4% in pearson coefficient. Besides of different types of lighting conditions, we also evaluated MobilePhys on different lighting

intensities on LED light (e.g., dimmer light). As Table 4.9 illustrates, in the 55 Lux condition, MobilePhys outperforms the baseline methods by 32.4%/31.5% in MAE, 21.5%/33.7% in SNR and 520%/9% in the pearson coefficient regarding the light intensity. In the 110 lux setting, we also observed a similar trend where MobilePhys outperforms both baseline methods in MAE and Pearson Coefficient. However, in the Lux 220 settings, MobilePhys achieves similar performance as TS-CAN while MetaPhys failed to generalize well in this setting.

To ensure our proposed system does not have a bias on a specific population or race, we evaluated MobilePhys's effectiveness across different skin types as Table 4.8 shows. In this set of experiments, we combined the subjects from Xiaomi 8 and iPhone 11 to get more balanced skin distribution across skin types. Results show that MobilePhys outperforms the baseline methods by 66.6%/31.1% in MAE, 108%/117% in SNR and 63.3%/3.1% in the pearson coefficient regarding the skin type I and II; 61.3%/45.5% in MAE, 108%/109% in SNR and 8.8%/12.5% in the pearson coefficient regarding the skin type III and IV; and 72.7%/86.7% in MAE, 15.4%/15.6% in SNR and 44.1%/133% in the pearson coefficient regarding the skin type V and VI.

Findings of Mobile Camera Settings

We observed that the Android phone camera settings greatly affected the quality of the video as well as the overall performance of camera physiological sensing. Anecdotally, we observed similar behavior in iPhone. Therefore, we conducted a systematic analysis on various camera settings on Xiaomi Mi 8 to explore how mobile camera settings affect the performance of contactless PPG measurement.

We choose the three most common camera settings that are: auto-white-balance (AWB), exposure time and sensitivity. Since camera physiological sensing systems aim to extract very subtle color changes on the skin, those settings play a significant role in the RGB values captured by the camera. More explicitly, these settings together determine the brightness and color of the video and the intake of the light from the camera lens. To further complicate matters, in video recordings, these parameters are typically changed dynamically and are not constant for a single video.

In the Android API, two parameters: color correction gains and color correction transform determine the AWB algorithm. Color correction gains are gains applying to Bayer raw color channels for white-balance, and color transform is a matrix used to transform the sensor RGB color space to the output linear sRGB color space. These two values are automatically generated when AWB is on, while in manual mode, users set these two parameters manually.

We conducted 13 experiments to examine the influence of auto-white-balance, camera sensitivity and exposure time on the performance of our camera physiological sensing system. The experiment procedure and the results are shown in Table 4.10. We repeated each experiment five times using the same camera setting, and the average heart rate MAEs are reported. All the experiments were completed in an hour to ensure that the lighting conditions were consistent across different experiments. In experiment #1, when auto exposure and auto white balance are turned on, a sensitivity of 175 exposure time of 1/30s was obtained as typical values and was used as the basis of the subsequent experiments. Typical color correction gains and color correction transform values were also obtained in this experiment and used for manual AWB mode. In the experiments #2-#10, auto exposure was turned off. Moreover, we also explored the sensitivity-exposure time space across nine different experiments. Experiment #11-#13 adopted the same exposure time and sensitivity as experiment #7, #5 and #8. The AWB parameters were set manually to see if AWB has a positive impact on the video quality,

In Table 4.10, the results show a noticeable decrease in averaged MAE when exposure time and sensitivity are set to a lower value. This indicates that these settings controlled the light admitted by the system and made the videos relatively darker. camera contactless physiological measurement aims to extract subtle changes (δ) of skin pixels; therefore, making the video too bright could corrupt these subtle signals. However, the averaged MAE in experiment #10 was increased sharply when we continued lowering the exposure time. This increment of MAE indicates that forcing the video to be too dark also makes extraction of physiological signals challenging as some of the individual's facial details were not even visible. Therefore, we set the exposure time to 1/50 s and sensitivity to 100 to balance the

brightness of videos. As for AWB, in all three experiments where AWB is off, the average MAE is slightly higher, which indicates that AWB helps capture higher quality of videos for the camera contactless physiological system. These findings in mobile camera settings help our system to get the best quality of videos recorded by smartphones' cameras.

Moreover, we also observed similar results on an OPPO Reno 5 smartphone, which suggests that this pattern is not device-dependent. This may guide future studies to set the proper mobile camera settings for camera contactless physiological sensing.

Table 4.10: Experiments on exploring the effect of smartphone front camera settings on the pulse measurement performance.

Trial No.	Duration (s)	Auto White Balance	Auto Exposure	Exposure time (s)	Sensitivity	MAE (beats/min)
1	30	On	On	Auto	Auto	5.6
2	30	On	Off	1/30	175	2.9
3	30	On	Off	1/30	100	1.2
4	30	On	Off	1/30	250	5.4
5	30	On	Off	1/50	175	2.5
6	30	On	Off	1/100	175	0.8
7	30	On	Off	1/50	250	2.5
8	30	On	Off	1/50	100	0.5
9	30	On	Off	1/100	250	0.6
10	30	On	Off	1/100	100	3.9
11	30	Off	Off	1/50	250	3.1
12	30	Off	Off	1/50	175	2.8
13	30	Off	Off	1/50	100	1.3

4.3.7 Discussion

In this paper, we demonstrated the feasibility of our proposed mobile personalizing camera contactless physiological sensing system using the power of meta learning, without the need for a clinical-grade sensor. This is achieved by using both the front and rear cameras on a smartphone to generate high-quality synchronized self-supervised labels. These labels are then used for training personalized contactless camera PPG models. These personalized

models could then be shared with other devices. We foresee the opportunity of future IoT applications (e.g., smart mirrors or other fitness applications) utilizing the smartphone's rear camera for a short calibration of a person's appearance and environment to personalize the physiological model. In the following sections, we discuss our major findings, limitation, and future work.

How Does MobilePhys Compare with the Baseline Methods?

Based on the results shown in Table 4.5-4.9, MobilePhys consistently outperforms the baseline methods across the tasks, with very few exceptions. Specifically, it helps significantly on the motion tasks (e.g., talking and random head motion) and in measurement across skin types based on Table 4.5 and Table 4.8. We find that MobilePhys achieves similar performance to MetaPhys in the talking and stationary after exercising tasks, but MetaPhys performs significantly worse than MobilePhys and TS-CAN in the random head motion, incandescent light and skin type (V+VI) tasks. We believe this is because MetaPhys is highly reliant on the POS method [148] to generate high-quality pseudo labels, but POS, a signal separation method fails to yield accurate PPG waveform in some of these more challenging tasks.

How Does MobilePhys Perform in iOS and Android? Based on Table 4.5-4.9, MobilePhys sees similar performance improvements on both Xiaomi 8 (Android) and iPhone 11 (iOS). These results indicate that MobilePhys has the potential to generalize to different mobile devices. Overall, we see no reason that the method would not provide benefits on other mobile devices, given that imaging devices on almost all mid-range and above smart phones have good specifications. However, while there are improvements over the baselines, we do observe that errors for subjects using the iPhone 11 are higher than those for subjects using Xiaomi 8. After analyzing the results, we found that there were two reasons causing this difference: 1) the data from iPhone 11 were collected over the winter quarter; therefore, there was a lower intensity of natural light in the recording environments than for the Xiaomi 8 data, which was collected over the spring/summer. It is harder to capture subtle skin pixel changes in low lighting settings. 2) The iPhone 11 data include 5 subjects with skin

types V+VI (1 V + 4 VI). Those subjects have higher melanin content in their skin, which impacts the intensity of light reflected from the body and the amount of light captured by a camera. Therefore, extracting subtle pixel changes from darker skin type subjects is more challenging [99]. In general, the iPhone 11 batch is simply a more challenging dataset than the Xiaomi 8 batch. We do not think that these differences are as much due to the hardware as these other factors.

How Do Motion Tasks Impact camera Contactless Physiological Systems?

As Table 4.5 shows, MobilePhys leads to considerable performance gains on tasks with larger motions (taking and random head motion). However, despite MobilePhys achieving more than 10% improvement in MAE of the task of head motion in yaw, it is failed to remove the large noise introduced by the larger motions as the MAE for this task even after personalization is still quite large. The subjects had larger motion in this task compared to the talking and random head motion tasks. These results indicate that it is easier for the model to pick up the personalized features in relatively stationary video frames. Large motions bring extra noise and MobilePhys may get "confused" about what to learn during the personalization phase. Moreover, UBFC only contains videos of stationary subjects; therefore, our meta-learner was only trained to learn how to adapt to relatively stationary tasks. The magnitude of motion was smaller in the tasks of talking and random head motion, than in yaw, and MobilePhys is able to yield greater improvements on these tasks.

How Do Lighting Conditions Impact camera Contactless Physiological Systems?

Based on the results in Table 4.7, MobilePhys yields superior performance on the conditions of natural light and incandescent light. MobilePhys reduces errors in the videos with incandescent light by approximately 40% on average across all the subjects. It is not surprising that MobilePhys achieves such results because none of the videos in AFRL and UBFC are recorded under incandescent lighting and incandescent has a different spectral composition compared to other lighting (e.g., sunlight), as illustrated in Figure 4.8. MetaPhys also does not work well in such complex lighting conditions, as POS was not designed to

handle videos with complex lighting environments. Furthermore, we observe that MobilePhys provides even more benefit for the videos under natural light. Natural sunlight has a broad spectrum (see Figure 4.8-A), and we hypothesize that our training data failed to represent a complete spectrum of natural sunlight. Therefore, performing a few-shot personalization substantially reduces the error by helping the model adapt to the different spectral profiles.

On the other hand, MobilePhys, MetaPhys and TS-CAN achieved similar performance under LED lighting. Since our training data (AFRL and UBFC) have a similar lighting spectral profile as the videos we collected under LED lighting, TS-CAN already performs strongly, showing that there is no need to personalize the model if the training data includes similar lighting conditions. However, in practice it is unlikely that the exact lighting conditions will be known at training time.

How Does Exercise Impact camera Contactless Physiological Systems?

It is clear that MobilePhys achieves better performance than TS-CAN after subjects raise their heart rate by exercising. Heart rate can be dramatically elevated after exercise. However, most of our training data have a regular range of heart rate (60 BMP to 80 BMP). MetaPhys also achieves similar performance as MobilePhys because the task was relatively stationary and under good lighting conditions so that POS was able to generate good pseudo PPG labels. Our results suggest that personalization is important when camera contactless physiological sensing technologies are deployed in fitness settings.

How Does Skin-type Impact camera Contactless Physiological Systems?

As Table 4.8 shows, MobilePhys achieves superior performance, especially in the subjects who have darker skin types such as VI. It is not surprising that MetaPhys also helps to provide an improvement on subjects of I, II, III and IV because of the personalization process. However, when it comes to the subjects in categories V+VI, MetaPhys failed to generalize to these subjects because POS cannot yield reliable pseudo PPG waveforms for the meta-learner. Previous work has already highlighted this issue with POS [99]. MetaPhys was even beaten by TS-CAN, and we believe it is because the meta-learner used an inaccurate waveform to generate the personalization model. On the other hand, MobilePhys was still able to

provide nearly 75% improvement compared to pre-trained TS-CAN. Through these results, we believe MobilePhys will be specifically useful in improving the equability of camera contactless physiological sensing.

4.3.8 Limitations

Although MobilePhys demonstrates it is able to achieve superior results in complex conditions, there remain limitations: (1) During data collection the smartphone was fixed on a stand. This helped us reduce some subtle motion artifacts. It might be reasonable to ask users to place/hold the phone for 18 seconds during the deployment; however, it would not be trivial to guarantee that they do so. (2) Although MobilePhys can help reduce the performance gaps between people with different skin types, there still remains some disparities. As Table 4.5-4.9 show, MobilePhys achieved higher errors on the iPhone 11 dataset than Xiaomi 8 dataset. We hypothesize this is because there are more VI subjects in the iPhone dataset. Our proposed method cannot completely close the gap, although we believe that our approach is a step in the right direction; (3) We are aware that we have a non-uniform distribution of skin types in this dataset, and the same is true for many other PPG datasets [99]. Specifically, we only have eight participants with the skin type of VI and V. Recent efforts have been made to address these imbalances, but these data are not publicly available at this time [90, 21]. We plan to expand our dataset with better coverage of skin types. (4) Finally, our system was only evaluated on limited daily motion tasks such as talking, and it is worth collecting more data with more routine activities such as typing, walking, etc. However, recruitment was challenging during the COVID-19 pandemic and we tried to capture a range of environments, activities and demographics; (5) The current system requires running a few-shot personalization process on every single task, which means users need to calibrate the system when they change their environment or activity.

4.3.9 Future Work

We are pushing toward more robust and generalizable mobile physiological sensing by demonstrating MobilePhys's performance on videos with two mobile devices, large head motions, different ambient lighting conditions and mobile camera settings. However, we conducted our experiments in a lab environment, and simulating different lighting conditions does not reflect the full diversity of conditions observed in everyday life. To further enable practical mobile physiological sensing, we would expect future work to study other mobile settings, such as deploying camera contactless physiological sensing in outdoor environments or in a gym. Moreover, modern smartphones are equipped with advanced cameras such as true-depth, IR cameras. It is also worth exploring how to leverage these advanced sensors to perform physiological sensing.

We noticed a 1-Hz noise signal that was constantly an issue when we collected the facial videos using the front RGB camera. We have noticed that this issue is not isolated only to the Xiaomi Mi 8 smartphone model. We observed the same issues on three other brands of Android smartphones and on iPhone with different camera specifications. Although we were not fully aware of the reason behind the 1-Hz noise issue, we observed that it was more likely to occur when the smartphone started to heat up and when using a large ISO setting and a longer exposure. Therefore, to ensure the quality of the facial videos, it is better to set a smaller ISO, use a faster shutter. We adopted a cooler and switched smartphones between recordings to ensure the phone did not overheat. Finally, we double-checked all recordings at the end of each experiment. We re-collected the data if we observed a significant 1-Hz noise issue in specific recordings. This issue seems to be partially related to the hardware and not entirely to our processing of the video.

We collected a large multi-modality mobile physiological sensing PPG dataset, which will be released with this paper. We would expect future work to explore novel contactless or contact physiological sensing methods and applications using our dataset. We foresee the opportunities of a multi-modality sensing approach (e.g., IMU, Audio, Ambient light and

RGB videos, etc.), contactless PPG measurement using the front IR cameras, computing other physiological signals (e.g., respiratory rate or heart rate variability), and other physiological computing applications.

4.4 Collaborative Learning with Imperfect Data

4.4.1 Introduction

Federated learning (FL) enables distributed devices (e.g., cellphones) to collaboratively learn models without data leaving each device [93, 59]. While creating traditional machine learning systems involves uploading raw data and labels to a centralized location for training, FL can avoid this. A core premise is that a model trained from aggregated decentralized data can be more effective than training with the data that any one device has access to on its own. More specifically, federated learning leverages locally-computed updates (weights) from a large number of single devices to create a robust aggregated model that can then be shared. To summarize, federated learning has several useful properties, the ability to: 1) preserve privacy more easily by only sharing model weights instead of raw data and labels, 2) increase the diversity and generalizability of a model by aggregating a diverse population's data, 3) reduce the bandwidth and storage resources required when uploading raw data to a centralized server.

The benefits of FL are particularly attractive in applications in which models rely on sensitive data that are also personally identifiable. This is very true in contexts that involve biometric, physiological and health data. Video recordings that contain the necessary fidelity to capture physiological changes contain *both* private health data *and* personally identifiable information. The physiological signals themselves have personally identifiable features [49] and the video frames may also contain visually recognizable body parts (e.g., the face). Furthermore, to effectively measure the very subtle changes in the body associated with these physiological processes, the videos should not be compressed too heavily as motion-compression algorithms typically remove the signals of interest [86]. As such, the recordings

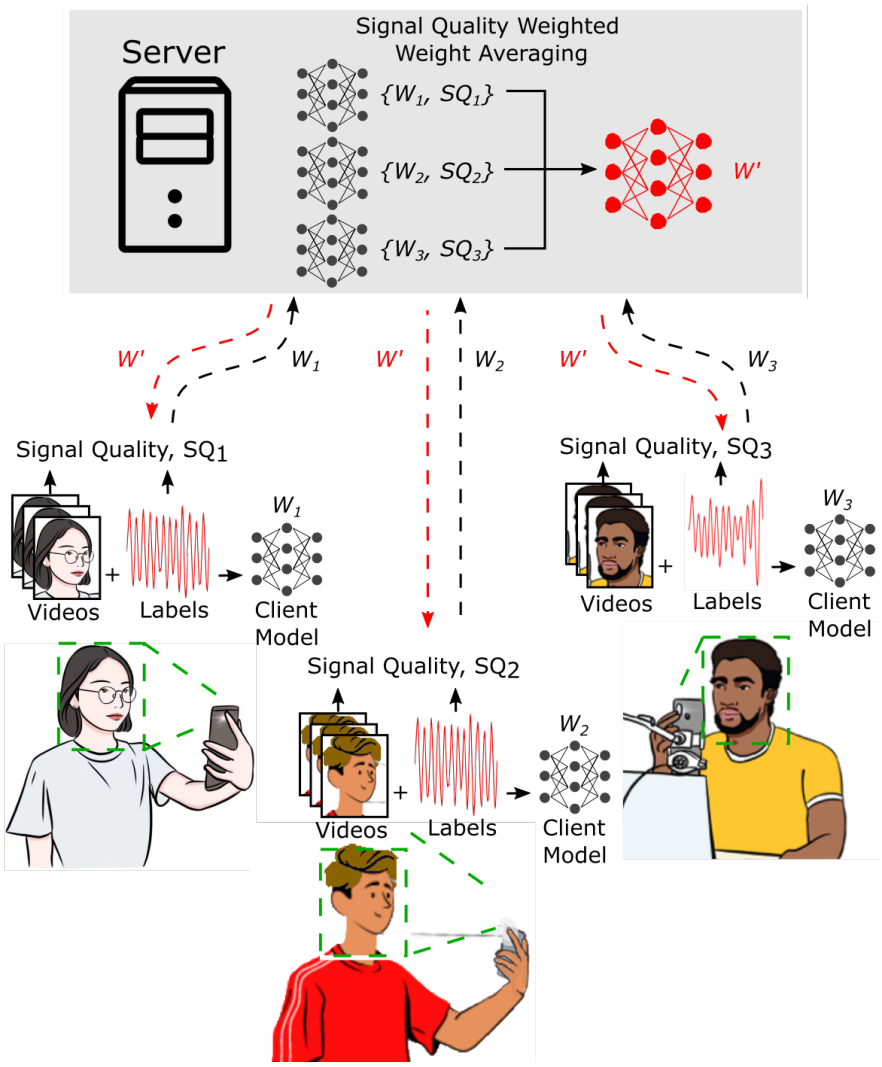

Figure 4.9: We present a privacy preserving federated system for on-device, camera physiological sensing. We propose a novel weight averaging approach that significantly improves on model robustness in the presence of noisy videos and labels. W_N represents the weights from each client, SQ_N represents the signal quality score either for the video, labels or both, and W' represents the server weights after weight averaging.

contain sensitive data and are often large; therefore, they ideally would not be transferred or stored in great volumes in the cloud.

When building models for measuring physiological vital signs, it is critical that the learned representations are not corrupted because of "bad" data (either features or labels) from a few devices. However in the context of FL where the server does not have access to the data itself, how do we ensure that this does not happen? Ideally, during weight aggregation it would be possible to adapt to, or *exploit*, client weights that were derived from cleaner rather than noisier data. At the same time, we do not want to completely ignore weights from a given client as every client will have access to data from a subject that was not "seen" by other clients and generally we would want a model to *explore* and maximize the diversity of our observations.

As shown in Fig. 4.9, in our scenario we have individuals collecting video on their own mobile devices alongside reference sensor measurements for training (as in [72]). In this case, there could be different levels of video noise resulting from camera sensor quality and automatic gain calibration. There could also be noise in the reference label, for example if a person was moving during the calibration period or did not attach the reference sensor correctly. Fortunately, both video and the physiological signals of interest (i.e., the PPG signal) have been studied extensively. We have strong statistical priors about the nature of these signals. In this work, to demonstrate our approach clearly we perform experiments assuming knowledge about the signal-to-noise ratios in the videos and labels. However, we could equally leverage domain knowledge to automatically calculate weight contributions from different devices. Our method does not discard the weights from clients with noisy data, but rather includes all weights while accounting for signal quality.

The contributions of this work are: 1) to introduce the first federated camera remote physiological measurement system, 2) to show that this system can match the performance of a traditional supervised learning approach, 3) to introduce a critical averaging approach that accounts for the signal quality and diversity of samples. 4) to provide an on-device mobile training and inference implementation. Our code, models, and video figures are provided in the supplementary materials.

4.4.2 Method

Traditional supervised learning approaches to camera physiological sensing have been trained on large-scale centralized video datasets and physiological labels [27, 159, 72]. There are several drawbacks to this. First, the data are highly identifiable containing appearance (e.g., faces) and physiological information. Second, these data consume considerable data storage resources (data for each subject often excess 1GB). For these reasons it would be desirable to have a solution that only involves analyzing videos on the client (so that videos need not be shared) and ideally in distributed manner. In this paper, we explore the use of federated learning in camera video-based physiological measurement. We leverage domain knowledge about the expected noise profile within our data to intelligently dynamically adjust how the model weights are averaged on the server. Our results empirically show that approach creates a more accurate physiological estimation model.

Federated Learning based Video-based Physiological Measurement. FL is a decentralized training schema where clients (i.e., smartphones) perform local training and upload trained model weights to a centralized server (e.g., the cloud). This training mechanism minimizes the risks associated with leaking identifiable or sensitive data. In the health and physiological sensing domain, federated learning has significant potential. Specifically in our scenario, FL means that facial video data and physiological gold-standard signals can remain on the mobile device and/or be processed in real-time and not transferred to any cloud storage. By only updating model parameters to the centralized server, we can learn a shared model through aggregating a large diverse population without collecting their own data.

As a baseline, we use FedAvg [93], the most commonly used federated learning algorithm. As Fig. 4.9 illustrates, each client uses video recordings and reference PPG signals captured by the owner of the device. These are used to train models local to each client. The model weights are then uploaded to a centralized server to execute model aggregation. FedAvg [93] uses an iterative model averaging approach to updating the model server's model's weights.

Algorithm 3 FedWeight: Federated Remote Physiological Measurement with Signal Quality Weighting

Require: S: Subject-wise video data

Server Update: with an initialization W_0
for each round $t = 1, 2, 3...$ **do**
    $S_t \leftarrow$ random select a set of clients
    for each client k in S_t **do**
        $\omega_t^k, b_t^k, \sigma_k = ClientUpdate(k, W_t)$
    end for
    $W_t = \frac{\sigma_k}{\sum \sigma_k} \cdot (\omega_t^k + b_t^k)$
end for
Client Update: (k, θ)
for each batch B in **do**
    $\omega_t^k, b_t^k \leftarrow \theta - \beta \nabla_\theta \mathcal{L}(f(\theta))$
    $\sigma_k \leftarrow$ assessing signal quality of client k based on noisy levels
end for

This approach has been shown to be effective on image classification tasks so we start with this technique as a baseline for creating camera physiological measurement models in a federate manner.

Noise Weighted Federated Learning. When training video-based physiological measurement algorithms, the goal is to recover physiological changes from very subtle (often sub-pixel) variations in image intensity. As we shall see training with FedAvg is effective if the training data from every client is "clean" (i.e., not corrupted). However, in reality it is much more likely to be the case that the quality of the training data on some individual devices will be better than others. This could be due to camera noise (e.g., quantization error) which can be most severe in poor lighting conditions when the gain is increased or user error in collecting and synchronizing the videos and reference physiological signals. Treating the weights from

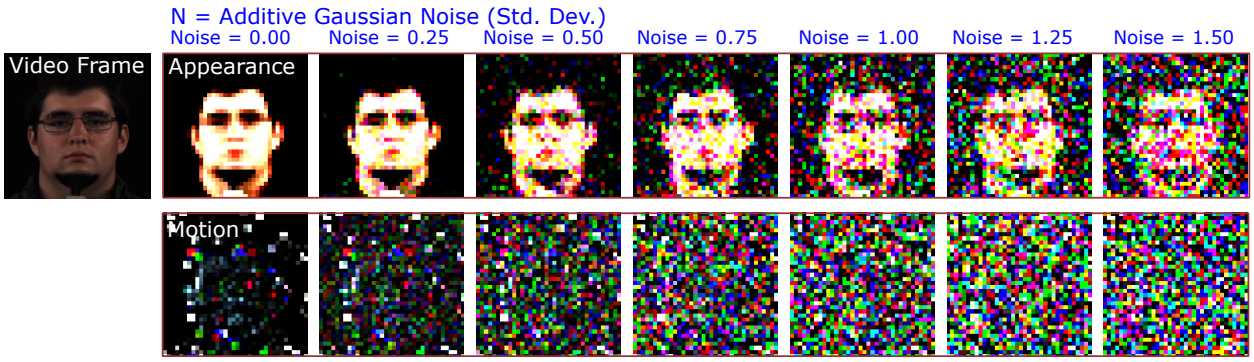

Figure 4.10: In our experiments we simulate camera sensor noise by adding Gaussian noise to the images. Here we illustrate the impact on the appearance and motion inputs to the two branch convolutional attention network.

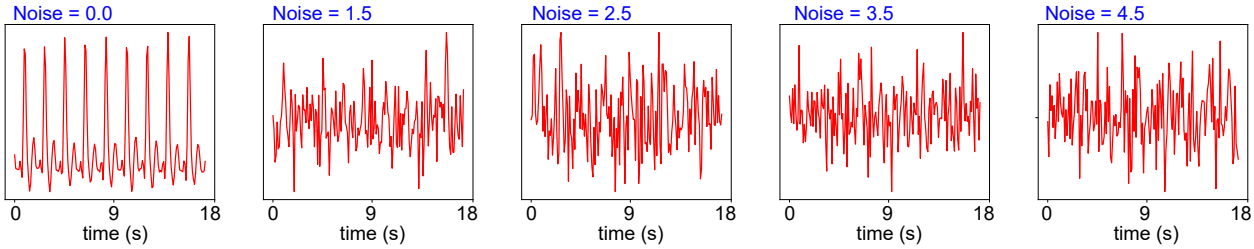

Figure 4.11: In our experiments we simulate contact reference PPG sensor noise by adding Gaussian noise to gold-standard contact sensor measurements. Here we illustrate the impact on training labels.

every client equally is naive and does not appear to be the best way to solve optimization if the quality of the data from some devices is worse than that from others. We would prefer to have a method that promotes weights from clients with less noisy data (exploitation) while still considering weights from all clients to promote diversity (exploration). In this paper, we propose a simple but effective version of federated averaging, called FedWeight, by leveraging

knowledge about the signal quality from each client. The centralized server model weight is calculated as in Equation 4.1 where k is the index of a layer, σ_i is the signal quality of client i, ω_i^k is the client i's model weights in the layer k , b_i^k is the bias in the client model weights in the layer k.

$$W_{server}^k = \frac{\sigma_i}{\sum \sigma_i} \cdot (\omega_i^k + b_i^k) \tag{4.1}$$

Our proposed signal-based aggregation is outlined in Algorithm 1. We first have an initialized centralized model weight W_0. Within each round of federated training, we randomly select a subset of clients for training. For each selected client, we then run a one-step optimization. After finishing local training for all the selected clients, we then perform signal-quality based aggregation as Equation 4.1 does. The output of each round in federated training is an aggregated model based on signal quality of selected clients' weights. Unlike FedAvg, which treats weights from all clients equally during model aggregation, our proposed leverages the fact that signal quality has a big impact on model performance to perform a more adaptive form of aggregation.

4.4.3 Implementation Details

We implemented our system in PyTorch [106], and all the experiments were conducted on an Nvidia 2080Ti GPU. We chose TS-CAN [70] as our backbone network to evaluate how FL works in remote physiological measurement since TS-CAN is the state-of-the-art neural network and can process frames in real-time on mobile platforms. To briefly summarize, TS-CAN is a two-branch neural network for on-device camera physiological measurement. The network contains an appearance branch that takes a sequence of normalized frames as inputs and generates attention masks to guide TS-CAN's motion branch. The motion branch takes a sequence of normalized difference frames (difference between every two consecutive frames). TS-CAN also leverages tensor shift modules to efficiently model temporal relationships which helps extract the subtle physiological signals in the videos. More details can be found in [70].

We first implemented TS-CAN with a window size of 20 frames instead of 10 frames because prior work has empirically shown a larger window size leads to better overall performance [72]. In this work, we focus on cross-dataset evaluation since the performance on cross-dataset evaluation is substantially worse than within-dataset evaluation using current state-of-the-art methods [27, 70]. We conducted all the federated training on the AFRL dataset [39] and evaluated the aggregated model on UBFC [12] and MMSE [163] datasets. For the federated training, we chose the Adam optimizer [58] with an learning rate of 0.001 on the client updates. We trained all the federated experiments for seven rounds until convergence. We followed the same training schema to replicate the traditional supervised performance of TS-CAN [70, 72].

To simulate different levels of noise in our training data (AFRL), we first sampled a subject noise level, σ_s, for each of the 25 subjects in the dataset from a Gaussian distribution with a mean equal to the experiment noise level (e.g. 0.25) and standard deviation of 0.1. During the training, to add noise to the videos we added Gaussian pixel noise from another distribution with mean of zero and standard deviation at the subject's noise level, σ_s. To add noise to the labels we added a vector of Gaussian noise from a distribution with mean of zero and standard deviation at the subject's noise level, σ_s. These noise samples were then were added to each video frames or ground-truth label vector, respectively, as the Fig. 4.10 and 4.11 illustrate. In the federated weighting process, the signal quality score was assigned to σ_s after normalizing across all subjects. As Fig. 4.10 and 4.11 show, we performed experiments adding six levels of noise to the videos [0.25, 0.50, 0.75, 1.00, 1.25, 1.50], and four levels of noise to the ground-truth labels [1.5, 2.5, 3.5, 4.5], respectively.

Since our network is trained on the derivative of the PPG signal [27], we applied standard post-processing steps to extract the heart rate estimate: 1) calculating cumulative sum and using a detrending function [136] (λ=10) to convert the signal to the PPG waveform; 2) dividing the estimated and ground-truth values for each participant into 360-frame non-overlapping moving windows (approximately 12 seconds); 3) applying a 2nd-order Butterworth filter with a cutoff frequency of 0.75 and 2.5 Hz which represents a realistic range of heart rates for adults. Following those steps, we then computed three metrics for each window including

Table 4.11: Comparison between traditional supervised training and FL with noise level of 0. Bold numbers reflect better performance.

	UBFC			MMSE		
Method	MAE↓	SNR↑	Pearson↑	MAE↓	SNR↑	Pearson↑
Supervised Training [70]	2.31	4.34	0.93	**2.99**	**2.42**	**0.79**
Federated Training	**2.00**	**4.38**	0.93	3.65	1.45	0.77

MAE = Mean Absolute Error in HR estimation, SNR = BVP Signal-to-Noise Ratio, ρ = Pearson Correlation in HR estimation.

the mean absolute error (MAE) in heart rate frequency between the predicted signal and the reference contact PPG, signal-to-noise ratio (SNR) [33] of the waveform and the Pearson correlation coefficient between the heart rate estimates and the those from the reference contact PPG. For heart rate estimation the frequency of the heart rate was determined by selecting the frequency with maximum power in the range [40Hz, 150Hz].

To explore the efficiency of end-to-end deployment in on-device training and inference, we also conducted experiments on a quad-core Cortex-A72 Raspberry Pi 4B to evaluate the model's performance on an edge device. We trained the model and performed inference 10 times to get a reliable averaged on-device training and inference time.

4.4.4 Results & Discussion

How does FL compare to regular supervised training? The results of regular supervised training and FedAvg FL are summarized in Table 4.11. For the UBFC dataset, FL outperforms regular supervised training. On the other hand, regular supervised training outperforms FL on the MMSE dataset. Through this comparison, we observe that the differences are small and that there is not a consistent accuracy difference between the two. However, FL has several additional benefits compared to regular training as have been discussed. Therefore, our results point to a promising future for FL in privacy preserving camera cardiac measurement.

How does video and label noise impact FL? Next, we examine how the performance

Table 4.12: Comparison between FedAvg and FedWeight with different levels of video noise.

		MAE (beats/min)↓		SNR (dB)↑		Pearson↑	
Dataset	**Noise**	**FedAvg**	**FedWeight**	**FedAvg**	**FedWeight**	**FedAvg**	**FedWeight**
	0	2.00	2.00	4.38	4.38	0.93	0.93
	0.25	3.06	**2.44**	2.33	**3.53**	0.83	**0.92**
	0.50	4.14	**2.90**	1.69	**2.01**	0.76	**0.89**
UBFC	0.75	4.59	**3.47**	0.02	**2.07**	0.76	**0.87**
	1.00	5.18	**4.16**	-1.18	**0.4**	0.75	**0.81**
	1.25	7.48	**7.02**	**-2.77**	-3.22	0.66	**0.79**
	1.50	7.44	**4.59**	-2.33	**-0.03**	0.66	**0.79**
	0	3.93	3.93	2.29	2.29	0.80	0.80
	0.25	4.58	**4.33**	0.67	**0.84**	0.65	**0.67**
	0.50	5.22	**4.44**	0.07	**0.41**	0.57	**0.68**
MMSE	0.75	6.46	**5.38**	-0.51	**0.07**	0.46	**0.54**
	1.00	6.58	**5.39**	-1.12	**0.01**	0.44	**0.56**
	1.25	6.61	**5.77**	-0.90	**-0.64**	0.44	**0.53**
	1.50	6.92	**6.17**	-2.29	**-1.79**	0.43	**0.55**

MAE = Mean Absolute Error in HR estimation, SNR = BVP Signal-to-Noise Ratio, ρ = Pearson Correlation in HR estimation.

Table 4.13: Comparison between FedAvg and FedWeight with different levels of label noise.

		MAE (beats/min)↓		SNR (dB)↑		Pearson↑	
Dataset	**Noise**	**FedAvg**	**FedWeight**	**FedAvg**	**FedWeight**	**FedAvg**	**FedWeight**
UBFC	0	2.00	2.00	4.38	4.38	0.93	0.93
	1.5	**1.79**	2.41	**4.72**	4.70	**0.96**	0.93
	2.5	2.05	**2.02**	**4.83**	4.69	0.94	**0.96**
	3.5	**1.88**	2.67	**4.28**	3.44	**0.96**	0.93
	4.5	2.73	**2.16**	3.94	**4.97**	**0.96**	0.94
MMSE	0	3.93	3.93	2.29	2.29	0.80	0.80
	1.5	**3.72**	4.07	**0.97**	-0.27	**0.78**	0.73
	2.5	4.60	**3.88**	-0.28	**0.36**	0.73	**0.79**
	3.5	4.44	**3.91**	-0.34	**0.38**	0.74	**0.79**
	4.5	4.94	**4.42**	-0.81	**-0.78**	0.72	**0.74**

MAE = Mean Absolute Error in HR estimation, SNR = BVP Signal-to-Noise Ratio, ρ = Pearson Correlation in HR estimation.

of FL is affected by noise in the videos and labels. Tables 4.12 and 4.13 and Fig. 4.12 show that the performance of the camera pulse measurement and heart rate estimation degrades significantly when using a naive weight averaging when some of the data is corrupted by noise. For example, in the noisy video experiments, we observed that the HR MAE increases by 19% and 20% when the noise level was increased from 0.25 to 0.5 and from 0.5 to 0.75 (UBFC dataset). However, a different pattern was found in the noisy label experiments described in Table 4.13. The MAE results remain similar across different noise levels, which indicates that noisy label does not significantly affect the performance of training and could be used as a regularization technique during training. Overall, the label noise had a much less severe impact on performance. In summary, simple federated averaging struggles with either noisy data or noisy labels in remote physiological measurement.

What is the impact of FedWeight? For the video noise level of 0.25, 0.5, 0.75, 1.0, 1.25 and 1.5, FedWeight improves 20%, 30%, 24%, 20%, 6% and 38% in MAE respectively,

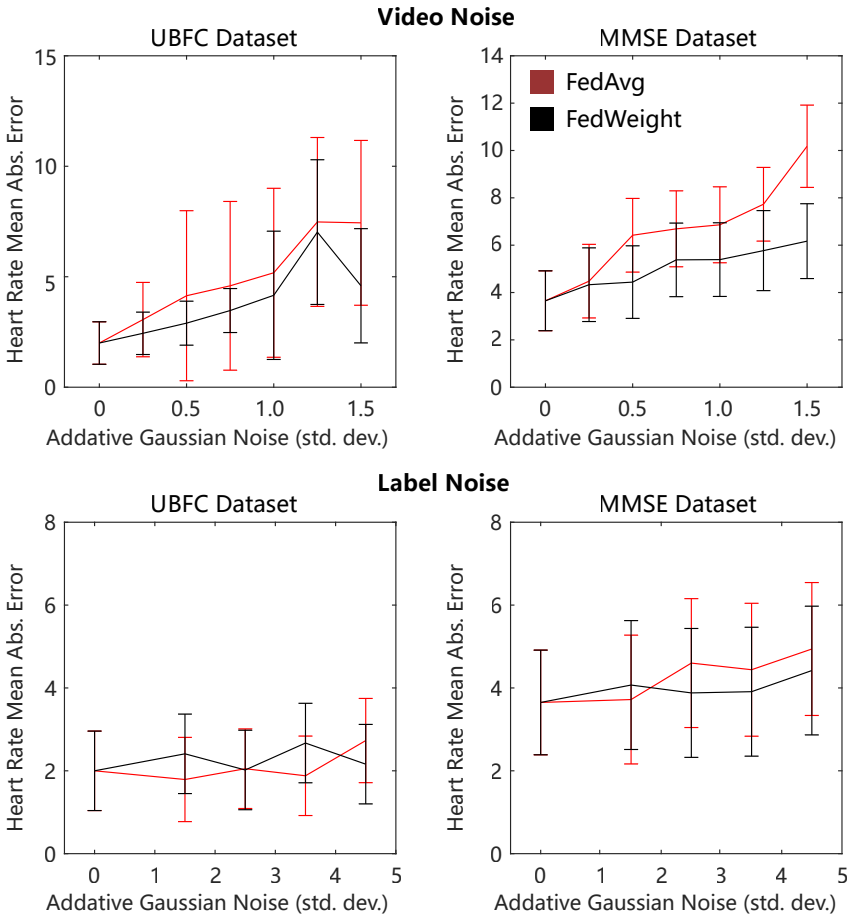

Figure 4.12: The heart rate mean absolute error for FedAvg and FedWeight at different video/label noise levels in the UBFC and MMSE datasets. Error bars reflect standard error where N is the number of videos.

when compared to FedAvg. A similar pattern was also observed in the MMSE dataset where FedWeight leads to a reduction of errors by 5%, 15%, 17%, 18%, 13% and 11% respectively. Moreover, our proposed FedWeight achieved comparable results as FedAvg in the case of noisy labels on the UBFC dataset. FedWeight helped achieve slightly better results in the MMSE dataset, but we still argue that noisy labels don't significant affect the performance of federated training or traditional supervised training. To summarize, intelligently combining weights using a signal quality weighted averaging method leads to a considerably more robust

model if the features (videos) are corrupted by noise. We believe that this result would likely by consistent for many other computer vision and machine learning tasks.

How to automate signal quality measurement? In this paper, we assume the noise level and signal quality are available to the centralized server. This could be the case if clients were able to provide a data quality report based on their knowledge of their individual sensor noise profiles. However, automating signal quality measurement would be preferred in many real-world scenarios. We are aware of this limitation and actively working on building an range of automatic signal quality metrics to test. Inspired by the metric in the task of super resolution, we argue that Peak Signal-to-Noise Ratio (PSNR) could be one way of measuring image noise level and quality. Moreover, we are also actively studying using the patterns of training loss and the quality of estimated PPG signal to assess the quality of videos.

Can we create an on-device FL prototype? We deployed our FL system on-device as part of our experimentation. The average on-device inference time was 24.5ms per frame while the on-device training time was 105ms per frame. Based on these results, the training time is almost five time the inference time. Deploying models like our on edge devices is non-trivial. Most deep learning frameworks [2, 105, 23] focus on training on server machines, leaving inference to edge devices [66, 55]. To enable efficient federated learning on edge devices, several challenges need to be solved: the underlying framework needs to allow efficient local training on the heterogeneous device; the runtime has to be small enough to fit on to a resource-constrained device; flexible communication patterns should be supported and simple to implement for different aggregation algorithms. We are actively exploring this direction based on deep learning compilation techniques [25, 121], including extending current deep learning compilers to training workload and optimize kernels for heterogeneous devices automatically.

4.4.5 Limitations

Although our proposed FedWeight improves on the performance of federated camera physiological measurement in the presence of noise, there are still a few limitations. First, we

picked six representative video noise levels and four label noise levels. However, these noise levels do not represent the entire spectrum of real-world noise. We plan to run greedy search experiments to explore more noise levels in the future. Second, we assume the "ground-truth" noise levels are available to the centralized server during model aggregation. In the future, we plan to develop a system to automatically measure noise levels and signal quality using domain knowledge (e.g., skewness of PPG signal and PSNR in the image) in imaging and physiology as discussed in section 4.4.4. Finally, we performed experiments on datasets that are not fully representative of all physical appearances. Before similar sensing algorithms are deployed they would require further validation and clinical evaluation.

4.4.6 Broader Impact

Ubiquitous computing offers a lot of potential for improving access to healthcare. For those that find it difficult to, or cannot, travel to a physician easily would benefit from technology that provides reliable measurement of physiological vital signs. If measurement can be performed from only a video, what happens if we detect a health condition in an individual when analyzing a video for other purposes. When and how should that information be disclosed? If the system fails in a context where a person is in a remote location, it may lead them to panic. For example, non-contact camera vital sensing can be used to measure a person's stress level without any notification. Especially during this pandemic, video conference meeting has become the major way to communicate between people. Non-contact physiological sensing could be easily plugged in softwares such as Zoom or Teams. Employer could easily sense their employees' health status during the meeting if we don't have the law enforcement for th is technology.

In the United States, a high standard was set by the Health Insurance Portability and Accountability Act (HIPAA) to protect sensitive patient data. We believe non-contat camera physiological measurement also should be under HIPPA compliance. Given the unique characteristic of camera physiological measurement, it even includes more sensitive information (e.g., long facial videos) than many other healthcare technology. We argue that

a special protection of data transferring should be enforced to minimizing the risk of data leaking. A better way to do this is to store and run inference on local mobile devices. However, how to collect large-scale physiological and video data to train a "super" model still remains challenge due to the concerns of data leaking and management. In this paper, we have successfully demonstrated how federated learning interplays with non-contact physiological sensing. Even without uploading a single raw video or physiological data to centralized server, it is still possible to attain a "super" aggregated model for everyone to use.

4.5 Improving the Dataset Landscape

4.5.1 Introduction

Camera physiological measurement is a rapidly growing field of computer vision and computational photography that leverages imaging devices, signal processing and machine learned models to perform non-contact recovery of vital processes inside the body [84]. Data plays an important role in both training and evaluating these models. However, generalization can be weak if the training data are not representative and systematic evaluation can be challenging if testing data do not contain the variations and diversity necessary. Public datasets (e.g., [163, 96, 12]) have contributed significantly to the understanding of algorithmic performance in this domain. These datasets are time consuming to collect, contain highly personally identifiable and sensitive biometrics (including facial videos and physiological waveforms). It is difficult to collect datasets that contain a well distributed set of examples across multiple cardiac and pulmonary parameters (e.g., heart and breathing rates and variabilities, pulse arrival times, waveform morphologies). Furthermore, almost all of these datasets are collected in a single location, with limited diversity in subject appearance, ambient illumination, context and behaviors. Table 4.14 summarizes some of the properties of these datasets, including whether they are freely (i.e., at no cost) available to researchers in both industry and academia. Finally, at the time of writing, neural architectures [27, 70, 160] provide the state-of-the-art performance for camera measurement of physiology. Neural

models are "data hungry" and often performance is primarily a function of the availability and quality of the training dataset.

Synthetics have proven valuable in several areas of computer vision, particularly face and body analyses. In training, synthetics have been used successfully to create models for landmark localization and face parsing [152], body pose estimation [124] and eye tracking [153]. Although not completely representative of real observations, synthetics are also valuable in testing (e.g., for face detection [87] or eye tracking [133]). Parameterized computer graphics simulators are one way of testing vision models [141, 142, 143, 140, 116]. Generally, it has been proposed that graphics models be used for performance evaluation [47, 116, 87]. However, increasingly synthetics are also being used to help address shortcomings in performance, such as biases. Kortylewaski et al. [61, 62] show that the damage of real-world dataset biases on facial recognition systems can be partially addressed by pre-training on synthetic data. To address the issue of the lack of representation of skin type in camera physiology datasets computational techniques have been employed to translate real videos from light-skin subjects to dark-skin subjects while being careful to preserve the cardiac signals [7]. A neural generator was used in that work to simulate changes in melanin, or skin tone. However, this approach does not simulate other changes in appearance that might also be correlated with skin type. Nowara et al. [100] used video magnification for augmenting the spatial appearance of videos and the temporal magnitude of changes in pixels. These augmentations help in the learning process, ultimately leading to the model learning better representations.

Wood et al. [152] recently presented a sophisticated facial synthetics pipeline that produced high-fidelity data. They were able to successfully train state-of-the-art landmark localization and face parsing models. However, creating high fidelity 3D assets for simulating many different facial appearances (e.g., bone structures, facial attributes, skin tones etc.) is time consuming and expensive. The data that these pipelines can create will then not necessarily be available broadly to researchers. Therefore, in this paper we present a new dataset (SCAMPS) of high fidelity synthetic human simulations that are made publicly available. These data are designed for the purposes of training and testing camera physiological measurement methods.

Table 4.14: Summary of Public Camera Physiological Measurement Datasets.

Dataset	Subjects	Videos	Gold Standard	Sub. Div.	Env. Div.	Free Access
MAHNOB [127]	27	527	ECG, EEG, BR	✗	✗	✓
BP4D [163]	140	1400	BP, AU	✓	✗	✗
VIPL-HR [96]	107	3130	PPG, HR, SpO_2	✗	✗	✓
COHFACE [50]	40	160	PPG	✗	✗	✓
UBFC-RPPG [12]	42	42	PPG, PR	✗	✗	✓
UBFC-PHYS [94]	56	168	PPG, EDA	✗	✗	✓
RICE CamHRV [103]	12	60	PPG	✗	✗	✓
MR-NIRP [101]	18	37	PPG	✗	✗	✓
PURE [132]	10	59	PPG, SpO_2	✗	✗	✓
rPPG [60]	8	52	PR, SpO_2	✗	✗	✓
OBF [67]	106	212	PPG, ECG, BR	✗	✗	✗
PFF [53]	13	85	PR	✗	✗	✓
VicarPPG [137]	20	10	PPG	✗	✗	✓
CMU [32]	140	140	PR	✓	✓	✓
SCAMPS*	2800	2800	PPG, PR, BR, AU	✓	✓	✓

ECG = Electrocardiogram waveform, EDA = Electrodermal activity, EEG. = Electroencephalogram waveforms, PPG = Photoplethysmogram waveform, BP = Blood pressure waveform, PR = Pulse rate, BR = Breathing rate, SpO_2 = Blood oxygenation, AU = Action Units.
* SCAMPS is the only synthetic dataset.

To summarize our contributions: 1) We present the first public synthetic dataset for camera physiological measurement. 2) These data include precisely synchronized multi-parameter physiological ground-truth waveforms (cardiac, breathing) alongside facial action and head pose. 3) Results illustrating baseline performance training on the SCAMPS dataset and testing on three public datasets (UBFC-rPPG [12], MMSE-HR [163] and PURE [132]). We hope that this dataset allows researchers to explore the potential for synthetics in the domain of camera physiological measurement, including but not limited to: addressing the simulation-to-real (sim2real) generalization gap, leveraging very precisely aligned segmentation maps and physiological waveforms for learning models, multimodal learning combining estimation of physiological (e.g., HR) and behavioral (e.g., AUs) signals, and using synthetic data to help address bias in camera physiological measurement models.

4.5.2 Waveform Synthesis

Our synthesis pipeline starts with a module for generating the underlying physiologic and behavioral signals. These signals are then used to drive those properties of the synthetic humans providing precisely synchronized ground-truth labels.[4] Examples of the generated waveforms can be found in Fig. 4.15. To create physiological waveforms with variability we sampled several waveform parameters, such as heart rate variability standard deviation of NN intervals (HRV SDNN), relative amplitude of the systolic and dicrotic waves and the delay between the systolic and dicrotic waves from a set of uniform distributions. The bounds used for each of these parameters are specified below.

Inter-beat Interval, PPG, ECG Waveforms. The PPG and ECG signals were created to have the same underlying beat sequence. We first sample the beat sequence based on a heart rate (HR) frequency sampled uniformly from 40 to 150 beats/min. Heart rate variability is simulated by adding random perturbations to the beat timings. The standard

[4]It is important to note that the purpose of our waveform synthesis approach was not to create signals derived from a true physical model of arterial hemodynamics and tissue perfusion, but instead to develop a simple and efficient way to generate physiologically plausible waveforms.

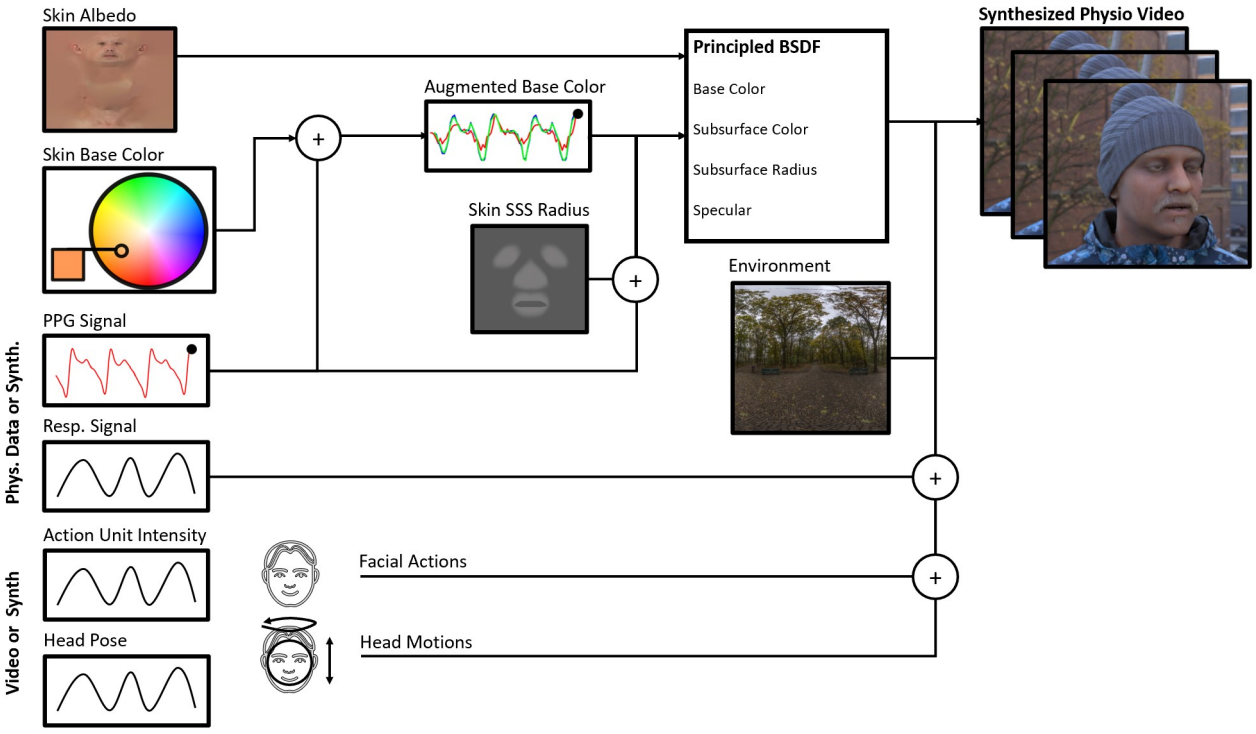

Figure 4.13: The synthetic videos were created using a graphics pipeline. We create a model of facial blood flow by adjusting properties of the physically-based shading material used for the skin and a model for breathing by controlling the motion of the head and torso. Facial actions and head motions are added to create realism and variability.

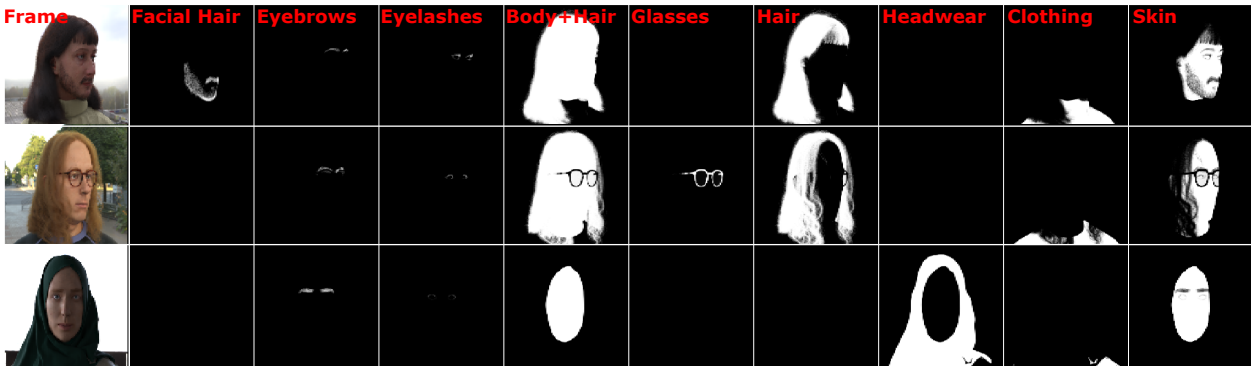

Figure 4.14: Each RGB frame is accompanied by segmentation masks for facial and body hair, eyelashes, eyebrows, glasses, skin, head wear and clothing.

deviation of these perturbations reflects the standard deviation of NN intervals (SDNN) and was sampled uniformly from 0.05 seconds to 8/HR seconds. We observed that it was important for the upper bound to be proportional to the heart rate (or mean NN interval) to create realistic variability.

For the purposes of this simulation, the morphology of the ECG wave is not relevant (e.g., we do not try to simulate a realistic QRS complex), only the timing. Thus, the ECG waveform is constructed as a time delayed series of impulses based on the NN intervals. We provide the interbeat intervals directly so that no peak detection is required for the ground-truth waveforms.

Given the beat timings and pulse arrival time (PAT) the PPG wave was then composed of a forward wave and dicrotic wave. The forward wave is created by convolving a Gaussian window with the beat impulse sequence. The leading slope of the dicrotic wave is created by convolving a Gaussian with a time lagged copy of the beat impulse sequence, the trailing slope is generated by performing the same convolution with a decaying exponential in place of the Gaussian window.

These waves are then summed together with a dicrotic amplitude factor. The forward

and dicrotic waves are then superimposed, with parameterized attenuation of the dicrotic wave relative to the forward wave, to create a physiologically plausible PPG waveform.

This signal was then low pass filtered to clean up the edges of the Gaussians, using a filter cut-off frequency of 8 Hz. Finally, the signal was normalized to give a signal of maximum amplitude of 1. This process creates PPG waveforms with the characteristic profile of systolic peaks and smaller diastolic peaks or inflections, but also with variability in the form. Finally, a small baseline drift at the breathing frequency is applied to the PPG signal to capture the subtle variations observed with breathing.

Breathing Waveforms. Each breathing waveform was created using sequence of breathing times based on a breathing frequency sampled from 8 to 24 breaths/min. A Gaussian window was convolved with the resulting impulse sequence. This signal was then low pass filtered to clean up the edges of the Gaussians, using a filter cut-off frequency of 8 Hz. Finally, the signal was normalized to give a signal of maximum amplitude of 1.

Facial Actions, Blinking and Head Pose. Unlike the physiologic waveforms, facial actions (with the exception of perhaps blinking) are rarely periodic. Therefore, we adopt an event based model [139]. For each facial action the event signal was created by a set of ramped step functions. The minimum and maximum event durations were 1 and 4 seconds, respectively. Blinking was treated separately from the other facial actions as the behavior is relatively more frequent and repetitive. For blinks the min and max event durations were 0.3 and 1 second respectively.

In each video we generate action unit "events". The start time and duration since previous event govern when the events onset and the gap between two events of the same action unit. These were are sampled from uniform distributions with bounds [0.3, 18] seconds and [1, 18] seconds, respectively. As such, in videos with action unit events there are examples of the onset and offset of most actions, some multiple times. Because facial actions are sparse but blinking occurs frequently, we generated all videos with blinking (eyes closed) events but only a subset of videos with facial actions, more details are provided below.

4.5.3 Video Synthesis

Identity. To create the avatars a generative 3D face model captures how face shape vary, and change during facial expressions. A blendshape-based rig is used with 7,667 vertices and 7,414 polygons and the identity basis is learned from a set of high-quality facial scans. In the creation of each avatar, we use a texture map transferred from one of the high-quality 3D facial scan as the albedo of the material for creating each face. These texture maps are sampled from a set of 511 facial scans of subjects including a range of skin types/tones, genders and ages. The distribution of gender, age and ethnicity of the subjects who provided the facial scans can be found in [152] (see Fig. 4). While these scans are not uniformly distributed across all demographic profiles, they do provide a wide range of appearances. Ongoing efforts are focused on creating more balanced facial scan dataset to help create even more diverse renderings. As only varying the blood flow signal in the skin is important for our use case the facial hair is removed from these textures by an artist. Then the skin properties can be easily manipulated. Hair (and clothing) are added back in later to create the final appearance.

We simulate blood flow by adjusting properties of the physically-based shading material[5]. We use a similar approach to that described by Wood et al. [152] and McDuff et al. [90]. We want the renders to display both diffuse and specular reflection effects, the diffuse reflection is handled as described below when we simulate blood flow and the specular reflection is controlled with an artist-created roughness map. Specular reflections make some parts of the face (e.g. the lips) shinier than others.

Hair, clothing and other apparel are added back in once the blood perfusion signal has been created. Hair is modeled as over 100,000 individual 3D strands to create a realstic effect. Hair as with clothing then occlude perfusion changes, as would be the case in real life.

Photoplethysmography. Changes in diffuse reflection due to blood flow are achieved by varying the surface color and subsurface scattering of the skin texture map. We simulate

[5]https://www.blender.org/

blood flow by adjusting properties of the physically-based shading material we use for the face. The synthesized PPG waveform is used to drive the temporal changes. We manipulate skin tone changes using the subsurface color parameters. The weights for this are derived from the absorption spectrum of hemoglobin and typical frequency bands from an exemplar digital camera[6] (Red: 550-700 nm, Green: 400-650 nm, Blue: 350-550 nm). We manipulate the subsurface radius for the channels to capture the changes in scattering as the blood volume varies within the skin. A subsurface scattering radius texture is used to spatially-weight these and simulate variations in the thickness of the skin across the face using an artist-created subsurface scattering radius texture. The same relative weighting of the RGB channels (0.36, 0.41, 0.23) is used for the BSDF subsurface radii. In absence of a more complex temporal-spatial model, we vary the parameters across the skin pixels in the same way across all frames. We recognize this is unlikely to be optimal, but does limit blood flow changes to skin pixels. We hope to be able to introduce a more realistic spatial variation in future. We used relative subsurface scattering coefficients of 0.36 (+/- 0.1), 0.41 (+/- 0.1) and 0.23 (+/- 0.1) for the red, green and blue channels respectively. Empirically we have found that this procedure works for creating data for training camera vital sign measurement. We found that varying the subsurface scattering alone, without changes in subsurface color, was too subtle and could not recreate the effects of BVP on reflected light observed in real videos.

Breathing. Inhaling and exhaling cause motions of the head and chest. To capture this in the avatars we use an approximation by controlling pitch of the chest and head using the synthesized breathing input signal. The amplitude of the head and chest motions were subtle and when combined with the head rotations and facial expressions are often difficult to see; however, prior validation has shown the models trained on similar synthesized data can generalise to real videos.

Facial Actions Facial expressions are controlled using blendshapes that map approximately to 10 facial action units [38]: outer brow raise (AU2), brow lowerer (AU4), eye lid

[6] https://www.bnl.gov/atf/docs/scout-g_users_manual.pdf

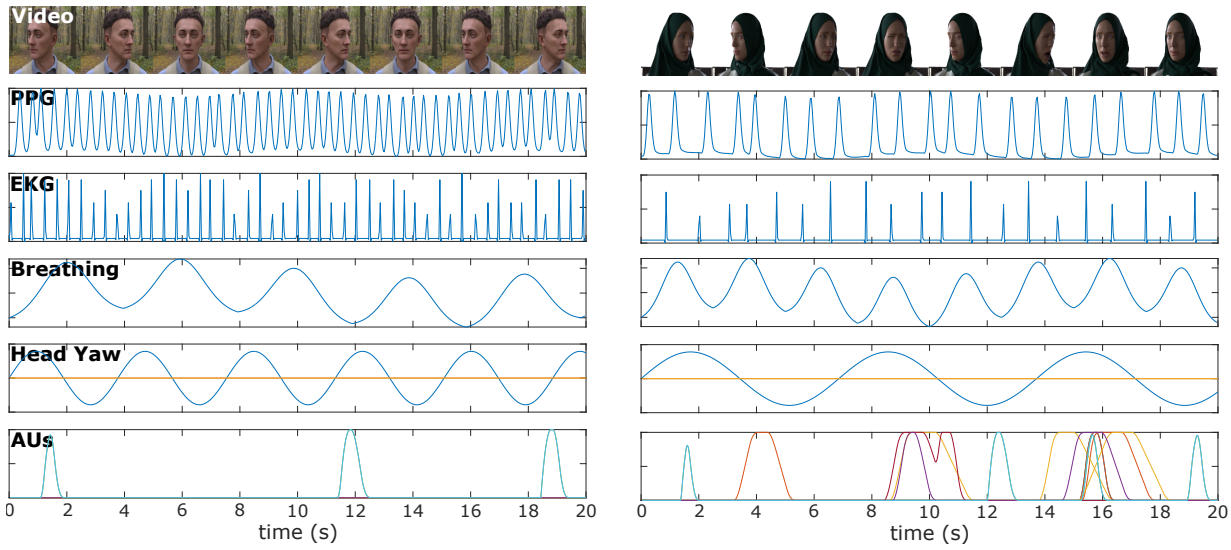

Figure 4.15: Our synthetic videos are accompanied by frame-level PPG, pseudo ECG/interbeat intervals, breathing, head pose and action unit labels. Here we show examples of two videos with a subset of video frames for reference.

tightener (AU7), lip corner puller (AU12), lip corner depressor (AU15), chin raiser (AU17), lip puckerer (AU18), jaw drop (AU26), mouth stretch (AU27) and eyes closed (AU43). The facial action coding system is a widely used and relatively objective method for quantifying facial movements [38]. The goal of controlling these actions is to create upper and lower facial motions. We recognize that the behaviors do not necessarily simulate realistic talking or expressions, as the dynamics of these are difficult to simulate.

4.5.4 SCAMPS Dataset

We created a dataset of 2,800 video sequences. The rendering required 24 machines each with an NVIDIA M40 GPU running for 720 hours each (a total of 17,280). This illustrates that creating synthetic data of this kind is not trivial and in part justifies the need for public datasets that can be shared amongst researchers. Each video has frame level ground-truth

Figure 4.16: Example frames from the SCAMPS dataset showing the diversity in avatar appearance, behavior and environment.

labels for PPG, inter-beat (RR) intervals, breathing waveform, breathing intervals and 10 facial actions. We also provide video level ground-truth labels for HRV SDNN, r-peak pulse arrival time (rPAT) and dicrotic wave amplitude. These parameters were used to generate a set of 20 second PPG waveforms at 300Hz. Finally, action unit intensities were generated. The ground-truth metrics are provided as both MAT and CSV files. Each video was then rendered using the corresponding waveforms and action unit intensities, and randomly sampled appearance properties, including skin texture, hair, clothing and environment.

Figure 4.17 shows the distribution of heart rates, HRV SDNNs, dicrotic wave amplitudes and breathing rates in the dataset. HR, rPAT and dicrotic wave amplitudes were sampled uniformly. HRV SDNN was not sampled uniformly, as qualitatively large HRV values, while interesting, could create quite extreme differences in interbeat intervals and we deemed it appropriate to create more examples with smaller variability.

To create a dataset that can be used for training and testing under a diverse range of conditions we synthesized videos while systematically changing different confounders: 1) head motions, 2) facial actions, and 3) dynamic illumination. A training, validation and test split of the data is provided on our project page as is a file indicating which confounders are

present in each video. As each video was sampled with a different combination of appearance parameters, they all contain avatars with different appearance. However, some avatars may look similar if they have the same skin texture and hair style. Figure **??** and 4.16 both show a collage of frames from different videos illustrating the diversity in appearance. The video frame (RGB) come with segmentation maps (see Fig. 4.14) that provide pixel level labels for beard, eyelashes, eyebrows, glasses, hair, skin and clothing. This is important as we know that the PPG signal will not be present in material that do not have blood flow (e.g., hair, clothing) and so we expect any supervised learning method to learn to segment skin as one of the operations. Therefore, we anticipate that segmentation maps will be useful to the community, both in training and in testing camera PPG methods.

Head Motions. Two thousand videos have rotation head motions and 800 have no head motion. Of the videos with head rotations, 1200 have smooth rotation (400 videos at 10, 20 and 30 degrees per second) and a further 800 have non-smooth head rotations in which the head was randomly positioned every second to a different angle. Ground-truth head angles are provided in the label files.

Facial Actions. Half of the videos (1,400) have facial actions generated with the event model described above, the other half have no facial actions. This enables training and/or testing systematically introducing the confounder of facial motions on the physiological measurement. The sequences and combinations of facial actions in each video were randomly sampled and therefore some of the facial expressions can look unnatural; however, this does provide a relatively dense set of examples of facial action onsets and offsets. We contrast this to many facial expression datasets in which facial actions are relatively sparse. We felt that more examples would generally be more useful for training models.

Background Motion and Dynamic Illumination. A set of 400 of the videos have dynamic illumination and background motion created by simulating the subject turning around in the environment. Half of these 400 videos have facial actions and half have head motions in addition to the background motion.

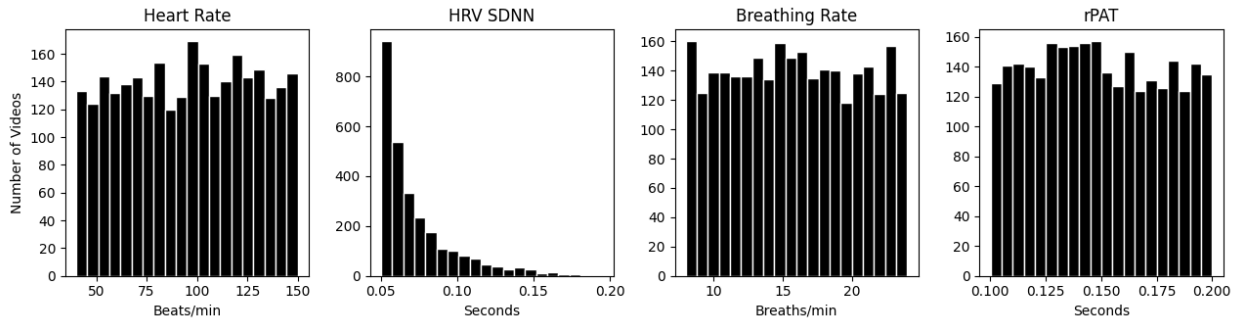

Figure 4.17: Examples of the distribution of heart rates, HRV SDNNs, breathing rates and dicrotic wave amplitudes in the SCAMPS dataset. An advantage of synthetic data pipelines is the ability to create a wide range of examples with specific distributions.

4.5.5 Baselines

One might ask the question "how well does a model trained on synthetic data generalize to real videos?" While there is some precedent for using synthetics for heart and breathing rate estimation [90, 91], those works did not use the SCAMPS dataset. To illustrate how this specific dataset can be used for video physiological measurement and provide initial baseline results, we performed experiments training with the SCAMPS dataset and testing on two public benchmark video datasets. To generate the results in this paper we used the opensource Deep Physiological Sensing Toolbox [76]. Links to the trained models can be found on our project page.

Model. Our goal here is not to provide an exhaustive list of results on different model architectures, but a representative baseline for researchers to compare to. We do not argue that this is the current state-of-the-art but rather is a reasonable starting point for future research with synthetic data in the field of camera physiological measurement. We implemented DeepPhys [27] as the baseline supervised model due to its relative simplicity. We trained on frames with resolution 72x72 pixels. First, we cropped the center 240x240 pixel region of each

320x240 pixel raw images. We then down sample these to 72x72 using a bilinear downsampling method. Difference frames were computed by performing a difference operation on successive frames. The resulting appearance and difference frames were normalized consistent with the method in Chen and McDuff [27]. These frames are then used for training the supervised model. We used a learning rate of 0.001 and the ADAM optimizer. We trained the model using videos from the SCAMPS training set for 10 epochs. We validated the SCAMPS validation set but used real-world videos as the testing sets. We used the Deep Physiological Sensing Toolbox [76] to complete all the training and testing procedures. The model from the epoch with lowest mean absolute error (MAE) heart rate estimation was selected and then we evaluated this model on the test sets. A Butterworth filter was applied to all model outputs (cut-off frequencies of 0.7 and 2.5 Hz) before computing the frequency spectra and heart rate.

Results. The results reported here are on the UBFC-rPPG [12], MMSE-HR [163] and PURE [132] datasets. Table 4.15 shows the mean absolute error (MAE), root mean squared error (RMSE) and correlation (ρ) in heart rate estimation compared to the gold-standard measures from each of the datasets. The results on both datasets show that the synthetic data are sufficient to train a reasonable supervised model. The trained model does not necessarily exceed the performance of the existing unsupervised methods and is in some cases a little worse. However, as first baselines these numbers do demonstrate that generalization from synthetic video to real ones is possible and also that there is room for improvement. By releasing the SCAMPS dataset we hope that researchers can design methods that bridge the sim-to-real gap that exists.

4.5.6 Access and Usage

The data may be used for research purposes and any images from the dataset can be used in academic publications. Researchers may redistribute the SCAMPS dataset, so long as they include all credit or attribution information and that the terms of redistribution require any recipient to do the same. The license agreement details the permissible use of the data and the

Table 4.15: Cross-dataset heart rate evaluation on UBFC, MMSE-HR and PURE (beats per minute).

	UBFC [12]			**MMSE-HR** [163]			**PURE** [132]		
Method	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑
DeepPhys[27] (trained on SCAMPS)	3.74	12.42	0.82	4.59	8.89	0.81	3.51	12.94	0.84
POS[150]	3.52	8.38	0.90	3.90	9.61	0.78	1.68	9.60	0.92
CHROM[33]	3.10	6.84	0.93	3.74	8.11	0.82	6.23	17.18	0.71
ICA[111]	4.39	11.60	0.82	5.44	12.0	0.66	5.70	18.10	0.70

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), ρ = Pearson Correlation in HR estimation.

appropriate citation, it can be found at: `https://github.com/danmcduff/scampsdataset`. Use of the dataset for commercial purposes is strictly prohibited, although research use at commercial companies is permissible. The authors commit to maintaining the dataset and ensuring access is available to the research community.

Some of our rendered faces may be close in appearance to the faces of real people. Any such similarity is naturally unintentional, as it would be in a dataset of real images, where people may appear similar to others unknown to them. As such there is no personally identifiable data or biometrics contained within the data, but the authors bear responsibility in case of any violation of rights that might occur.

4.5.7 Transparency and Broader Impacts

This dataset was created for research and experimentation on camera measurement of physiological signals. While the dataset is useful for testing models, was not designed as a test set for evaluating the clinical efficacy of a model, just because a model performs well on synthetic data does not mean it will generalize to videos of real people. The SCAMPS dataset was not designed for computer vision tasks such as face recognition, gender recognition, facial attribute recognition, or emotion recognition. We do not believe this dataset would be

suitable for these applications without further validation.

We have tried to make this dataset representative of a diverse population and the physiological waveforms are completely synthesized, so do not contain identifying information. However, our dataset still does not capture a uniform distribution of skin types and other appearance characteristics. We are working on addressing these limitations. When using this dataset, as with others, one should be careful to pay attention to biases that might exist. Please see the SCAMPS dataset datasheet [43] included in the supplementary material and linked from our project page for more details.

Non-contact camera vital sign monitoring has great potential as a tool for telehealth. Our proposed system can promote global health equity and make healthcare more accessible for those in rural areas or those who find it difficult to travel to clinics and hospitals in-person (perhaps because of age, mobility issues or care responsibilities). These needs are likely to be particularly acute in low-resource settings. An advantage of camera physiological measurement is that contact with the body is not required and that cameras are ubiquitous sensors. However, these advantages can cause problems. Unobtrusive measurement from small, ubiquitous sensors makes measurement without a subject's knowledge simpler. It is important that norms and regulations that govern on-body physiological measurement devices are extended to camera measurement systems. Consent should always be obtained from subjects before measuring physiologic data of this kind. It is always important to consider how such technology could be used by "bad actors". In the case of physiological measurement, it should be required to inform subjects when these methods are being used and for consent to be obtained before physiological data is measured or recorded. There should be no penalty for individuals who decline to be measured.

4.5.8 Future Directions

The SCAMPS datasets is a first of its kind. Therefore, we wanted to only include renderings for which we had a sufficiently robust synthetics pipeline. In the SCAMPS dataset we did not synthesize videos with very abnormal rhythms, or specific types of arrhythmia (e.g., Premature

Ventricular Contraction - PVC, Atrial Fibrillation - AFib., etc.) A distinct advantage of synthetic data generation is the ability to create examples of rare events "at will"; however, creating data that are faithful to real-world observations is non-trivial. Therefore, the first version of the SCAMPS dataset contains pulse signals with heart rate variability, but not specific arrhythmia. We hope that future research will address this gap.

To make the dataset more plausible, a simulation of ballistic forces (e.g., ballistocardiogram) would be helpful, as would a more sophisticated absorption model that reflects how absorption might change under different conditions. Simulating scar tissue, makeup and other skin markings (e.g., tattoos or piercings) would also help provide better representation of appearances to the dataset. Our current rendering engine is not capable of simulating scar tissue and the skin albedos we used did not have tattoos. These are examples of why it is important to pay attention to biases that might exist in models trained with SCAMPS and why it would not be appropriate to deploy a model trained on SCAMPS without further work.

4.6 Conclusion

In this chapter, a series of works is presented to address the gaps in equitability in camera health sensing. The proposed work include a few-shot unsupervised learning algorithm, a personalized mobile systems, a federated learning algorithm, and a synthesized dataset.

The first work introduces a novel few-shot adaptation framework called MetaPhys, which utilizes an optically-grounded unsupervised learning technique and a modern meta-learning framework to jointly train a remote physiological network. This is the first work to use pseudo labels in training a physiological sensing model and the first unsupervised deep learning method in remote physiological measurement. The proposed method substantially improves on the state-of-the-art and the performance on various skin types, and also reveals how the method achieved such improvement.

The second work presents a mobile camera contactless physiological sensing system called MobilePhys. This system leverages front and back cameras to provide self-supervised

"ground truth" labels to the few-shot meta learning algorithm to perform personalization and environmental adaptation. Additionally, the first-ever multi-modality mobile remote physiological dataset with different mobile devices, lighting conditions, motions, activities, and skin types is released to validate the robustness of the system. The proposed system substantially improves over the state-of-art system under different contexts and provides useful guidance for researchers who are building smartphone-based contactless physiological sensing systems.

The third work presents a federated learning system called FedWeight, which accounts for training imperfect data such as noisy data or noisy labels in the task of camera remote physiological measurement. The proposed method is more robust to corruption, particularly video noise, and has the potential for various applications in mobile health, especially in remote physiological measurement.

Lastly, the SCAMPS dataset is presented, which contains high-fidelity simulations designed for training and testing camera physiological sensing algorithms. The dataset captures a diverse range of appearances, environments, and lighting conditions and includes synchronized ground-truth signals such as interbeat and breath intervals and PPG, ECG, and breathing waveforms. The dataset also provides facial actions, blinking, and head pose labels. The benchmark experiments show that it is possible to train models only with synthetic data that generalize to real videos. The dataset aims to support research towards more robust and fair vision-based physiological sensing models.

Chapter 5

PUSHING THE LIMIT OF CAMERA HEALTH SENSING: A TRANSNATIONAL CLINICAL STUDY

5.1 Introduction

To date, the large majority of camera physiological measurement research has involved the development and evaluation of algorithms using data of healthy subjects in research labs as Figure 5.1 shows. There are some notable exceptions [1], but these are in the minority. While the research as a whole has contributed much to the understanding of the fundamental challenges associated with recovering subtle physiological signals from videos, it is also fundamentally limited. The existing datasets, especially those that are publicly available, by and large do not include irregular or problematic vitals such as atrial fibrillation and other forms of arrhythmia, low oxygen saturation levels (i.e., below 85%) or high blood pressure. It is important that validation of the performance of non-contact camera system against gold-standard measurements considers the full spectrum of physiological states that could be expected, even if only rarely.

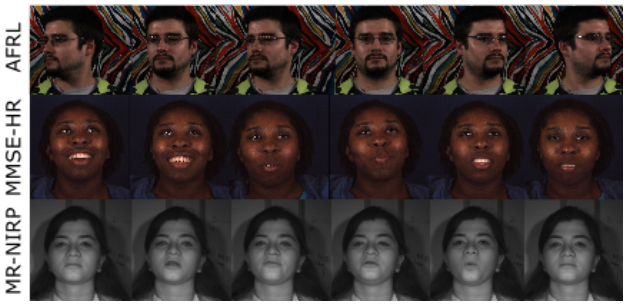

Figure 5.1: Public Datasets in Remote Photoplethysmography

To do this, deploying the system in clinical settings is necessary. The most successful examples of such deployments have been in neonatal intensive care units [1]. Another promising direction in clinical validation is deploying camera systems in tele-cardiology contexts. Validating non-contact solutions in clinical settings could open the door for future applications, such as large-scale screening for arrhythmia, but in order to do so there needs to be high confidence in the precision of these approaches [1]. How much positive impact and improvement in telehealth visits that non-contact camera system can generate is an interesting question for the entire community. Validating non-contact solutions in clinical settings will open the doors for numerous future clinical applications.

In this thesis, we conduct a transnational clinical study where we deploy a non-contact camera physiological sensing system in a patient room to monitor patients vitals while collecting ground-truth vital signals from the clinical-grade contact sensors. Validating non-contact solutions in clinical settings will open the doors for numerous future clinical applications.

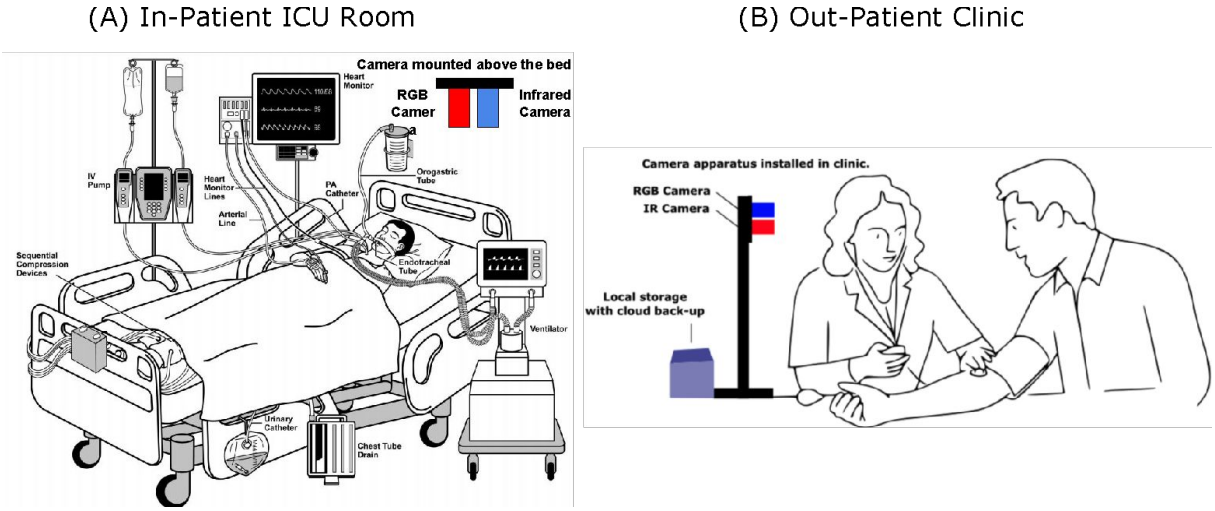

Figure 5.2: camera Health Sensing in Clinical Setting

5.2 Beyond Heart Rate: Estimating Blood Pressure from Videos

Remote photoplethysmography (rPPG) provides a non-invasive approach to capturing peripheral blood flow using a camera. While its potential to accurately predict blood pressure (BP) has been demonstrated in healthy, normotensive subjects, the validity of rPPG in predicting BP for patients with cardiovascular disease (CVD) remains unexplored. This study serves as the first investigation into the utilization of rPPG for BP prediction within a CVD patient cohort.

Between March and December 2022, a total of 132 patient visits were completed in a cardiology clinic, as illustrated in Figure 5.3-B. The study protocol involved patients being comfortably seated while a digital camera recorded two separate 2.5-minute videos focusing on their face and palm. Blood pressure measurements were taken from the left arm at the beginning, midpoint, and end of each session using an oscillometric cuff, thus providing a reliable reference for the rPPG-derived BP estimates.

Moreover, continuous heart rate and rhythm were recorded through a single-lead electrocardiogram (ECG) facilitated by electrodes positioned on the right and left wrists. In addition, a contact sensor placed on the left index finger provided continuous photoplethysmographic (PPG) measurements. This comprehensive data acquisition approach allowed for a multi-faceted examination of the physiological parameters relevant to BP prediction, thus providing a robust foundation for exploring the potential of rPPG in predicting BP among patients with CVD.

The video data was processed through a series of steps to extract the relevant physiological signals needed for blood pressure prediction. Initially, videos were cropped and subjected to spatial averaging to extract the green channel, which serves as a surrogate for the remote photoplethysmographic (rPPG) signal. The green channel is often used in rPPG signal extraction due to its relative insensitivity to changes in skin color and ambient lighting conditions.

Further, a pulse segmentation and selection process was implemented to isolate five

high-quality consecutive heartbeats from the rPPG signal. These beats were chosen based on their quality, which is of paramount importance for the accuracy of subsequent analysis and prediction.

Finally, the raw rPPG signal, along with its first and second derivatives, were fed into the neural network. The derivatives of the rPPG signal provide additional information about the rate of change of the signal, potentially revealing more subtle features and patterns in the data that the raw signal might not capture. This approach of incorporating raw signals along with their derivatives aims to leverage the comprehensive information content inherent in the rPPG signal for more accurate blood pressure prediction.

The structure of the network is constructed around five convolutional blocks, each followed by a pooling layer designed to reduce the spatial dimension of the network's output, thereby controlling overfitting. To incorporate patient-specific data and add a layer of context to the system's prediction capabilities, age and Body Mass Index (BMI) information are concatenated into the output of an intermediate convolutional block, acting as complementary sources of information.

The architecture of the network culminates in a fully connected layer that serves as the final predictive layer, which generates predictions for blood pressure. This design is based on the hypothesis that the combination of visual features, derived from convolutional blocks, and patient-specific data, such as age and BMI, can provide a comprehensive feature set for accurate blood pressure prediction. This network structure leverages the strengths of convolutional neural networks for feature extraction, and the use of personalized data, to develop a model capable of providing reliable blood pressure estimates.

The network's performance is optimized through a training process aimed at minimizing the mean squared error (MSE) between the predicted blood pressure values derived from video-based contactless measurements and the traditional cuff-based blood pressure measurements. This objective function ensures that the network's predictions are as close as possible to the actual blood pressure values, thereby improving the reliability and accuracy of the system.

To further validate the model's performance and its ability to generalize across different

subjects, a 5-fold subject independent cross-validation was implemented. This technique partitions the dataset into five subsets, ensuring that each subset comprises distinct subjects. The model is then trained and validated iteratively on different combinations of these subsets, thus ensuring that the evaluation of the model's performance is unbiased and that the model is capable of handling variability across different individuals. This rigorous cross-validation process enhances the robustness and reliability of the proposed model.

5.3 Results and Conclusion

Our study begins by validating the performance of heart rate estimation using rPPG. Utilizing a straightforward model that relies on the green channel, we achieved a mean absolute error (MAE) of 3.14 beats per minute and a root mean square error (RMSE) of 5.70, indicating a robust capability for heart rate estimation. The R-square value of 0.86, as displayed in Figure 5.3, further substantiates the efficacy of our heart rate estimation model.

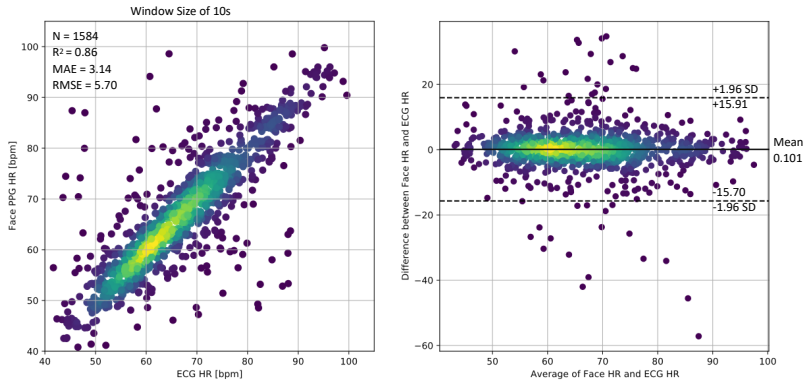

Figure 5.3: Heart Rate Estimation using Camera Health Sensing

Turning our attention to systolic blood pressure estimation (see Figure 5.4), our model demonstrated a bias of 0.75 mmHg, a standard deviation of 13.41 mmHg, and a Pearson correlation of 0.61. As for diastolic blood pressure estimation, our trained model achieved a

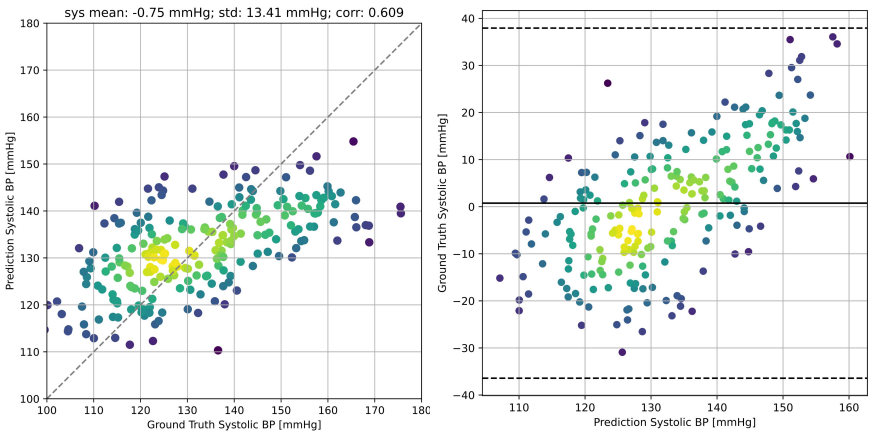

Figure 5.4: Systolic Blood Pressure using Camera Health Sensing

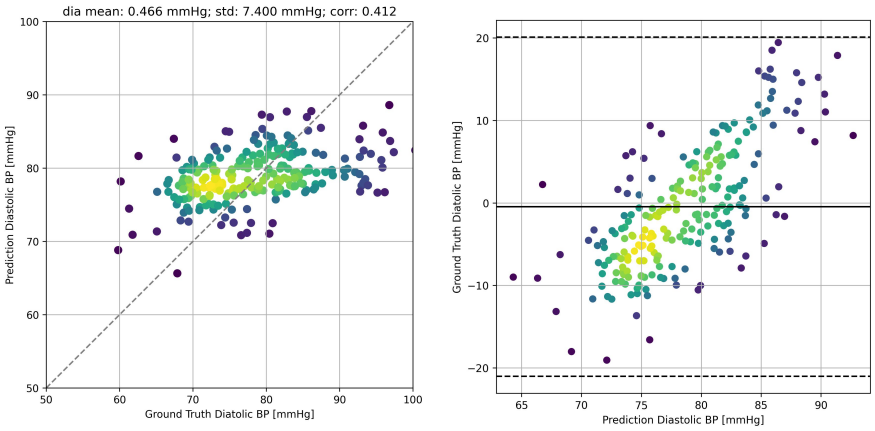

Figure 5.5: Diastolic Blood Pressure using Camera Health Sensing

bias of 0.47 mmHg, a standard deviation of 7.40, and a Pearson correlation of 0.41 (see Figure 5.5). Despite room for improvement in blood pressure estimation, these results underscore the potential of rPPG for extracting blood pressure measurements from CVD patients in a clinical setting.

This study paves the way for further research into harnessing rPPG for non-invasive, contactless blood pressure estimation, particularly among populations with cardiovascular disease. The robust performance of the proposed methods in a real-world clinical setting

supports the feasibility of this approach and its potential for broader application in personalized healthcare.

Chapter 6

OPEN-SOURCE RPPG-TOOLBOX: DEEP PHYSIOLOGICAL SENSING TOOLBOX

6.1 Introduction

There are hundreds of computational architectures that have been proposed for the field of camera health sensing [84]. However, standardization in the field of physiological measurement is still severely lacking. Based on our review of literature in the space, we identified four issues that hindered the interpretation of results in many papers. First, and perhaps most obvious, a lot of the published work is not accompanied by public code. While publishing code repositories with papers is now fairly common in the machine learning and computer vision research communities, it is far less common in the field of camera physiological sensing. While there are reasons that it might be difficult to release datasets (eg. medical data privacy), we cannot find good arguments for not releasing code.

Second, many papers do not compare to previously published methods in an "apples-to-apples" fashion. This point is a little more subtle, but rather than performing systematic side-by-side comparisons between methods, the papers pull numbers from previous work. Unfortunately, this often makes it unclear if performance differences are due to pre-processing steps, model design, post-processing, training scheme and hardware specifications, or a combination of the aforementioned. Continuing this thread, the third flaw is that papers use pre- and post-processing steps that are often lacking in detailed description. Finally, different researchers compute the "labels" (e.g., HR frequency) using their own methods from the contact PPG or ECG timeseries. Differences in these methods lead to different labels and a fundamental issue when it comes to benchmarking performance. When combined, the aforementioned four issues make it very difficult to draw conclusions from the literature about

Table 6.1: Comparison of rPPG-Toolbox with existing toolbox in camera physiological measurement.

Toolbox	Multiple Dataset Support	Uns. Eval	DNN Training	DNN Eval
iPhys-Toolbox [89]	✗	✓	✗	✗
Boccignone et al. [13]	✗	✓	✗	✗
PPG-I Toolbox [108]	✗	✓	✗	✗
pyVHR [14]	✓	✓	✗	✓
rPPG-Toolbox (our)	✓	✓	✓	✓

Uns. = Unsupervised learning methods, DNN = Deep neural network methods.

the optimal choices for the design of rPPG systems.

Open source code allow researchers to compare novel approaches to consistent baselines without ambiguity regarding the implementation or parameters used. This transparency is important as subsequent research invariably builds on prior state-of-the-art. Implementing a prior method from a paper, even if clearly written, can be difficult. Furthermore, it is an inefficient use of time for many researcher to re-implement all baseline methods. To address this, several open source toolboxes have been released for camera physiological sensing. These toolboxes have been a significant contribution to the community and provide implementations of methods and models [85, 13, 108]. However, these toolboxes are incomplete. McDuff and Blackford [85][1] implemented a set of source separation methods (Green, ICA, CHROM, POS) and Pilz [108] published the PPGI-Toolbox[2] containing implementations of Green, SSR, POS, Local Group Invariance (LGI), Diffusion Process (DP) and Riemannian-PPGI (SPH) models.

These toolboxes are implemented in MATLAB (e.g., [85]); however, Python is now the language of choice for a large majority of computer vision and deep learning research. There are several implementations of popular signal processing methods in Python: Bob.rrpg.base[3]

[1] https://github.com/danmcduff/iphys-toolbox

[2] https://github.com/partofthestars/PPGI-Toolbox

[3] https://pypi.org/project/bob.rppg.base/

includes implementations of CHROM, SSR and Boccignone et al. [13] released code for Green, CHROM, ICA, LGI, PBV, PCA, and POS. Several published papers have included links to code; however, often this is only inference code and not training code for neural models. Without providing training code for neural networks, it is challenging for researchers to conduct end-to-end reproducible experiments and build ideas on top of it.

In this paper, we present an end-to-end toolbox called rPPG-Toolbox[4] for camera physiological measurement. This toolbox includes: 1) support for multiple public datasets, 2) pre-processing code to format the datasets for training neural models, 3) implementations of neural model architectures and unsupervised learning methods, and 4) evaluation and inference pipelines for supervised and unsupervised learning methods for reproducibility. We hope that this toolbox helps the research community establish clearer benchmarks in order to compare methods in a more fair way.

6.2 The rPPG-Toolbox

To address the gaps in the current set of tools and to promote reproducibility and clearer benchmarking within the camera physiological measurement community, we present a toolbox with fully open-source code. This toolbox is designed to support multiple public datasets, preprcessing steps, model training, and evaluation.

6.2.1 Dataset and pre-processing Support

The toolbox includes pre-processing code that converts multiple public datasets into a form amenable for training with neural models. The standard form for the videos we select includes raw frames and difference frames (the difference between each pair of consecutive frames) stored as numpy arrays in a [N, W, H, C] format. Where N is the length of the sequence, W is the width of the frames, H is the height of the frames and C is the number of channels. There are six channels in this case, as both the raw frames and difference frames account

[4] https://github.com/ubicomplab/rPPG-Toolbox

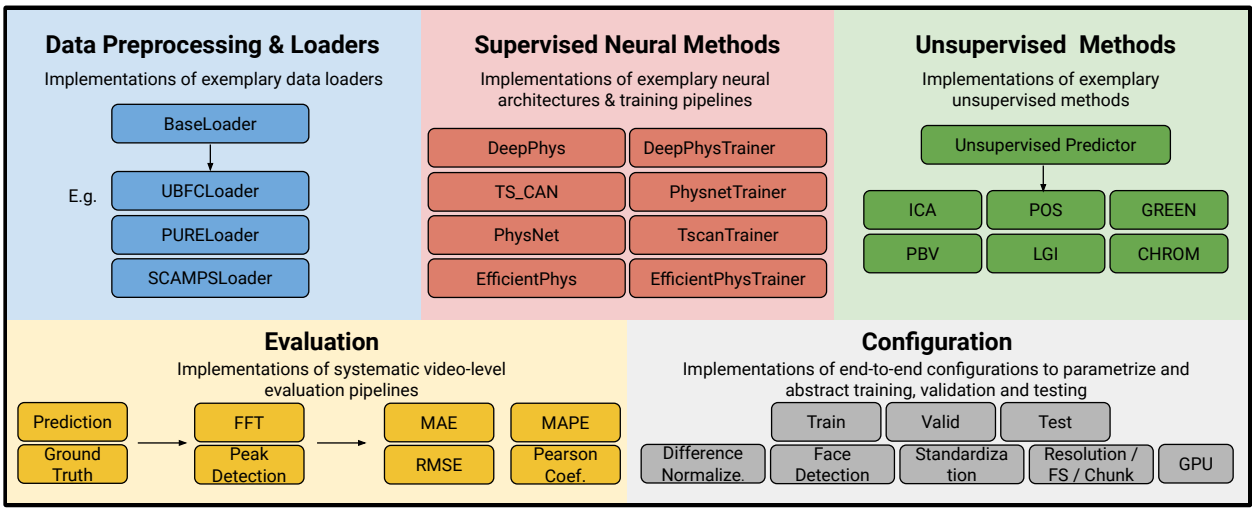

Figure 6.1: High-level schematic of the rPPG-Toolbox codebase.

for three color channels each. For faster data loading, all videos in the datasets are typically broken up into several "chunks" of non-overlapping N (e.g., 180) frame sequences. All of these parameters (N, W, H, C) are easy to change and customize. The PPG waveforms (labels) are stored as numpy arrays in a [N, 1] format. The entire pre-processing procedure is supported with multi-thread processing to accelerate the data processing time as possible.

In the first version of the rPPG-Toolbox we have provided pre-processing code for three commonly used, public datasets: UBFC [12], PURE [132] and SCAMPS [92].

6.2.2 Unsupervised Methods

The following methods all use linear algebra to recover the estimated PPG signal.

Green [144]: is an unsupervised learning method that uses the green channel information as the proxy for the PPG after spatial averaging of RGB video.

ICA [111]: is an unsupervised learning method that, using Independent Component Analysis (ICA), is applied to normalized, spatially averaged color signals to recover demixing

Table 6.2: Baseline results on the UBFC-rPPG [12] and PURE [132] datasets generated using the rPPG toolbox. For the supervised methods we show cross-dataset training results using and on the UBFC-rPPG and PURE datasets.

Training Set	**PURE** [132]				**UBFC** [12]			
Testing Set	**UBFC** [12]				**PURE** [132]			
	MAE↓	RMSE↓	MAPE↓	ρ ↑	MAE↓	RMSE↓	MAPE ↓	ρ ↑
Supervised								
TS-CAN [70]	1.29	2.87	1.50	0.99	3.69	13.84	3.38	0.82
PhysNet (Normalized) [159]	1.63	3.78	1.68	0.98	9.36	20.63	17.84	0.62
DeepPhys [27]	1.21	2.90	1.42	0.99	5.54	18.51	5.32	0.66
EfficientPhys-C [71]	2.07	6.32	2.10	0.94	5.47	17.04	5.39	0.71
Unsupervised								
POS [148]	4.00	7.58	3.86	0.92	3.67	11.82	7.25	0.88
PBV [34]	15.90	26.39	15.17	0.47	3.91	12.99	4.82	0.84
LGI [109]	15.80	28.54	14.70	0.36	4.61	15.38	4.96	0.77
CHROM [33]	3.98	8.72	3.78	0.89	5.77	14.92	11.52	0.81
ICA [111]	14.70	23.71	14.34	0.53	4.77	16.70	4.47	0.72
Green [144]	19.81	31.49	18.78	0.36	10.09	23.85	10.28	0.34

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), ρ = Pearson Correlation in HR estimation.

matrices.

CHROM [33]: is an unsupervised learning method that uses a linear combination of the chrominance signals obtained from the RGB video.

POS [148]: short for plane-orthogonal-to-the-skin (POS), is a method that calculates a projection plane orthogonal to the skin-tone based on physiological and optical principles. A fixed matrix projection is applied to the spatially normalized, averaged pixel values, which are used to recover the PPG waveform.

Table 6.3: For the supervised methods we show results training with the SCAMPS [92] dataset.

Training Set	**SCAMPS** [92]							
Testing Set	**UBFC** [12]				**PURE** [132]			
	MAE↓	RMSE↓	MAPE↓	ρ ↑	MAE↓	RMSE↓	MAPE↓	ρ ↑
Supervised								
TS-CAN [70]	3.62	6.91	3.53	0.93	4.66	13.69	5.83	0.82
PhysNet(Normalized) [159]	4.39	9.31	4.39	0.85	20.08	27.56	31.27	0.09
DeepPhys [27]	3.10	9.81	3.08	0.87	3.95	13.44	4.25	0.83
EfficientPhys-C [71]	12.64	24.00	11.26	0.34	10.24	21.65	11.70	0.46

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), ρ = Pearson Correlation in HR estimation.

PBV [34]: uses a signature, that is determined by a given light spectrum and changes of the blood volume pulse, in order to derive the PPG waveform while offsetting motion and other noise in RGB videos.

LGI [109]: is a feature representation method that is invariant to motion through differentiable local transformations.

6.2.3 Supervised Methods

The following implementations of supervised learning algorithms are included in the toolbox. All implementations were done using PyTorch [106]. Common optimization algorithms, such as Adam and AdamW, and criterion, such as mean squared error (MSE) loss, are utilized for training. The learning rate scheduler follows the 1cycle policy, which anneals the learning rate from an initial learning rate to some maximum learning rate and then, from that maximum learning rate, to some learning rate much lower than the initial learning rate. The total steps in this policy are determined by the number of epochs specified multiplied by the number of

training batches in an epoch. The 1cycle policy allows for convergence due to the learning rate being adjusted well below the initial and maximum learning rates, after numerous epochs in which the learning rate is much higher than the final learning rate. We found this learning rate scheduler to provide stable results with convergence using a maximum learning rate of 0.009 and 30 epochs.

DeepPhys [27]: is a two-branch 2D convolutional attention network architecture. The two representations (appearance and difference frames) are processed by parallel branches with the appearance branch guiding the motion branch via a gated attention mechanism. The target signal is the first differential of the PPG waveform.

PhysNet [159]: is a 3D convolutional network architecture. Yu et al. compared this 3D-CNN architecture with a 2D-CNN + RNN architecture, finding that a 3D-CNN version was able to achieve superior pulse rate prediction errors. Therefore, we included the 3D-CNN in this case. It is worth noting that we used difference-normalized frames as input to PhysNet as the original paper does not specify a concrete input data format.

TS-CAN [70]: is a two-branch 2D convolutional attention network architecture that leverages temporal shift operation information across the time axis to perform efficient temporal and spatial modeling. This network is an on-device, real-time algorithm. The target signal is the first differential of the PPG waveform.

EfficientPhys-C [71]: is a single-branch 2D convolutional neural network that aims to provide an end-to-end, super lightweight network for real-time on-device computation. The architecture has a normalization module that calculates frame differences and learnable normalization as well as a self-attention module to help the network to focus on skin pixels associated with PPG signal.

6.2.4 Pre-processing, Training, post-processing and evaluation

In the rPPG-Toolbox, we offer a configuration file system that enables users to modify all parameters used in pre-processing, training, post-processing, and evaluation. A YAML file is provided for every experiment and includes blocks for pre/post-processing, training, validation,

testing, model hyperparameters, and computational resources. The pre/post-processing for neural and unsupervised methods share similar settings, such as the same input resolution and face cropping.

In terms of pre-processing, we provide three input data types: 1) "DiffNormalized", which calculates the difference of every two consecutive frames and labels, and normalizes them by their standard deviation; 2) "Standardized", which standardizes the raw frames and labels using z-score; 3) "Raw", which uses the original frames and labels without modification. Additionally, we also provide parameters for face cropping, which is a vital aspect of our task. In the config file, users can use dynamic detection to perform face cropping every N frames and scale the face bounding box by a coefficient to maintain consistency of face cropping in motion videos.

With regard to training of neural network, our toolbox provides flexibility to parameterize which portion of the data used for training / vaidation / testing. For instance, we can use first 80% of UBFCC for training, the last 20% of UBFC for validation and then use the entire PURE dataset for testing. Moreover, the sceptical parameters of each neural network can be also define at the config file such as dropout rate etc.

For post-processing and evaualtion, there are several standard post-processing steps that are typically employed to improve model predictions. A 2nd-order Butterworth filter (cut-off frequencies of 0.75 and 2.5 Hz) is applied to filter the predicted PPG waveform. The choice of filtering parameters can have a significant impact on downstream results such as heart rate errors. A Fast Fourier Transform or a peak detection algorithm is then applied to the filtered signal to calculate the heart rate. In this toolbox, we support four metrics for video-level heart rate estimations: mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE) and Person Correlation (ρ). For better reproducibility, we also provide pre-trained models in our Github repository to allow researchers to perform model inference. The detailed definition of each config parameter is also provided at the Github repository.

6.2.5 Baselining and Results

To show that the implementations of the baseline methods are functioning as expected and provide benchmark results for consumers of the toolbox to reference and reproduce, we performed a set of baseline experiments using three commonly used video rPPG datasets: SCAMPS [92], UBFC [12] and PURE [132]. For neural models, a training batch size of 4, 30 epochs, and an inference batch size of 4 was utilized for all experiments. The models were trained using one Nvidia 2080 Ti GPU. Table 6.2 and Table 6.3 show MAE, RMSE, MAPE and Person Correlation (ρ) computed between the video-level heart rate estimations and gold standard measurements.

6.3 Conclusion

Research relies on the sharing of ideas. This not only allows methods to be verified, saves time and resources, but also allows researchers to more effectively build upon existing work. Without these resources and open-sourced code bases, fair evaluation and comparison of methods is difficult, creates needless repetitions, and wastes resources. We present an end-to-end and comprehensive toolbox, called rPPG-Toolbox, containing code for pre-processing multiple public datasets, implementations of supervised machine learning and unsupervised methods (including training pipeline), and post-processing and evaluation tools.

Chapter 7

CONCLUSION AND FUTURE DIRECTIONS

7.1 Can the lessons learned from camera-health sensing be applied other health sensing domains?

Throughout the process of developing camera health-sensing systems, several valuable insights have been gained that can be readily applied to other health sensing domains, such as audio health sensing or wearable health sensing.

A pivotal lesson derived is that accuracy and efficiency are both integral to the effectiveness of health sensing systems, primarily due to the necessity for privacy preserving on-device deployment. The system should not only perform accurately but also be computationally efficient, facilitating real-time or near-real-time operations on a range of devices. Beyond the system's performance, it is crucial to consider the ease of deployment and reproducibility of the system, which can promote the widespread adoption of the technology. Fostering an open-source community in the health sensing domain would be advantageous for collectively enhancing and expanding the impacts of the work.

The incorporation of self-explainable modules in the health sensing methods proposed in this thesis has proven to be beneficial. One salient example is the spatial attention module, which not only enhanced the system's performance but also facilitated debugging and visualization of the system's operation. Techniques such as T-SNE for visualizing the embedding space have provided valuable insights into how the learned features correspond to real-world data, offering a meaningful representation of different skin tones, individuals, and so forth.

Personalization is a critical aspect in health sensing systems. Given that biomarkers and sources can significantly vary across individuals, tailoring these systems to each user's unique

physiological characteristics and needs is essential. In this thesis, we explored the potential of few-shot personalization via meta-learning as a strategy for personalization. However, other strategies could also be leveraged. For instance, transfer learning can be employed to adapt a general model to an individual's specific profile based on a small amount of personal data.

In validating health sensing systems, it is vital not to confine the process to lab settings but to extend it into real-world settings. This could involve conducting large-scale longitudinal studies in naturalistic environments or engaging in rigorous clinical trials. Such an approach ensures the system's utility and robustness outside of controlled environments and contributes to more meaningful and applicable findings.

In summary, these lessons underpin the importance of accuracy, efficiency, ease of deployment, self-explanation, personalization, and real-world validation in designing and implementing health sensing systems. They offer a substantial foundation for future research and development in the broader field of health sensing. Applying these insights to other domains such as audio health sensing and wearable health sensing could facilitate the creation of more effective, user-friendly, and clinically relevant health technologies.

7.2 Future Directions

Building upon the lessons learned from this thesis, a compelling direction for future work is the development of a unified multimodal foundation model for mobile health as Figure 7.1 illustrates. This envisioned the foundation model would combine multiple modalities of physiological sensing, including vision-based, wearable, and audio sensors, into a comprehensive health assessment tool. Such an integration could leverage the complementary strengths of each modality, providing a more holistic and accurate health assessment.

An important aspect of this multimodal foundation model would be its ability to perform early diagnosis with few-shot tuning. With the burgeoning of personalized healthcare, the ability to quickly adapt to new patients and scenarios using minimal data is becoming increasingly crucial. The proposed foundation model will harness the power of few-shot learning, allowing it to learn new tasks or adapt to new patients using only a small number

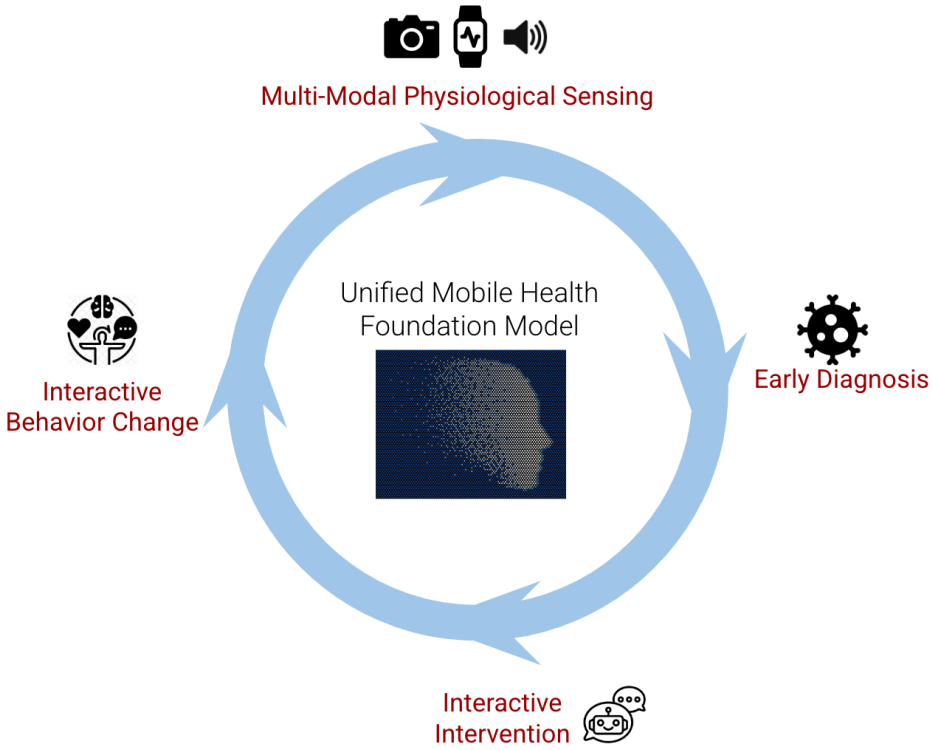

Figure 7.1: Unified Multimodal Mobile Health Foundation Model.

of examples.

Additionally, with the help of advancement of large language models, the foundation model would incorporate language capabilities, enabling it to interact directly with users. This interaction could take the form of automated patient interviews, reminders for medication adherence, or even interactive interventions for behavior change. The model's ability to 'communicate' can potentially enhance patient engagement, facilitate health education, and enable real-time intervention, leading to improved health outcomes.

Moreover, an integral part of the foundation model would be its ability to visualize its knowledge and insights. By offering users an understandable and interactive visualization of their health data and the model's outputs, we could empower users to make informed health decisions and encourage behavior change. A user-friendly interface would allow users

to access and interpret their health data, fostering an increased understanding and active participation in their health management.

In conclusion, the future direction proposed in this thesis emphasizes the importance of multimodal physiological sensing, few-shot learning for disease diagnosis, user interaction and data visualization with advanced language capabilities. The development of such a unified multimodal foundation model for mobile health has the potential to revolutionize personalized healthcare, promoting early diagnosis, patient engagement, and behavior change, ultimately leading to improved health outcomes.

7.3 In Summary

In summary, this thesis has significantly advanced the field of camera health sensing by addressing four critical research questions. Firstly, the design and deployment of efficient on-device neural networks has enabled accessible, real-time camera health sensing on mobile devices. This was achieved while ensuring state-of-the-art accuracy and ease of deployment, thereby overcoming substantial technical challenges. Secondly, a concerted effort was made to promote equity and generalizability in camera health sensing. Strategies such as the MetaPhys and MobilePhys algorithms, weighed federated learning, and the creation of a synthetic dataset have contributed to the development of systems that can effectively function across diverse skin tones, contexts, and motion levels. Furthermore, the boundaries of camera health sensing have been pushed beyond cardiac measurements, with successful blood pressure estimation from facial videos in a clinical setting. Lastly, the open-source toolbox introduced in this thesis will standardize the processes of preprocessing, training, and evaluation, encouraging wider participation and collaboration in the camera health sensing community. These contributions encapsulate a substantial stride towards accessible, equitable, generalizable, and useful camera health sensing.

BIBLIOGRAPHY

[1] Lonneke AM Aarts, Vincent Jeanne, John P Cleary, C Lieber, J Stuart Nelson, Sidarto Bambang Oetomo, and Wim Verkruysse. Non-contact heart rate monitoring utilizing camera photoplethysmography in the neonatal intensive care unit—a pilot study. *Early human development*, 89(12):943–948, 2013.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[4] Mohammed Hazim Alkawaz, Ahmad Hoirul Basori, and Siti Zaiton Mohd Hashim. Oxygenation absorption and light scattering driven facial animation of natural virtual human. *Multimedia Tools and Applications*, 76(7):9587–9623, 2017.

[5] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer. *arXiv:2103.15691 [cs]*, March 2021. arXiv: 2103.15691.

[6] Consumer Technology Association. Physical activity monitoring for heart rate, ansi/cta-2065, 2018.

[7] Yunhao Ba, Zhen Wang, Kerim Doruk Karinca, Oyku Deniz Bozkurt, and Achuta Kadambi. Overcoming difficulty in obtaining dark-skinned subjects for remote-ppg by synthetic augmentation. *arXiv preprint arXiv:2106.06007*, 2021.

[8] Nannapas Banluesombatkul, Pichayoot Ouppaphan, Pitshaporn Leelaarporn, Payongkit Lakhan, Busarakum Chaitusaney, Nattapong Jaimchariyatam, Ekapol Chuangsuwanich, Wei Chen, Huy Phan, Nat Dilokthanakul, et al. Metasleeplearner: A pilot study on fast adaptation of bio-signals-based sleep stage classifier to new individual subject using meta-learning.

[9] Gary G Berntson, John T Cacioppo, and Karen S Quigley. Respiratory sinus arrhythmia: autonomic origins, physiological mechanisms, and psychophysiological implications. *Psychophysiology*, 30(2):183–196, 1993.

[10] Ethan B Blackford, Justin R Estepp, and Daniel McDuff. Remote spectral measurements of the blood volume pulse with applications for imaging photoplethysmography. In *Optical Diagnostics and Sensing XVIII: Toward Point-of-Care Diagnostics*, volume 10501, page 105010Z. International Society for Optics and Photonics, 2018.

[11] Vladimir Blazek, Ting Wu, and Dominik Hoelscher. Near-infrared ccd imaging: Possibilities for noninvasive and contactless 2d mapping of dermal venous hemodynamics. In *Optical Diagnostics of Biological Fluids V*, volume 3923, pages 2–9. International Society for Optics and Photonics, 2000.

[12] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82–90, 2019.

[13] Giuseppe Boccignone, Donatello Conte, Vittorio Cuculo, Alessandro d'Amelio, Giuliano Grossi, and Raffaella Lanzarotti. An open framework for remote-ppg methods and their assessment. *IEEE Access*, 8:216083–216103, 2020.

[14] Giuseppe Boccignone, Donatello Conte, Vittorio Cuculo, Alessandro D'Amelio, Giuliano Grossi, Raffaella Lanzarotti, and Edoardo Mortara. pyvhr: a python framework for remote photoplethysmography. *PeerJ Computer Science*, 8:e929, 2022.

[15] Elizabeth Bondi, Debadeepta Dey, Ashish Kapoor, Jim Piavis, Shital Shah, Fei Fang, Bistra Dilkina, Robert Hannaford, Arvind Iyer, Lucas Joppa, et al. Airsim-w: A simulation environment for wildlife conservation with uavs. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–12, 2018.

[16] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

[17] Eoin Brophy, Maarten De Vos, Geraldine Boylan, and Tomas Ward. Estimation of continuous blood pressure from ppg via a federated learning approach. *arXiv preprint arXiv:2102.12245*, 2021.

[18] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[19] Bryson Carrier, Brenna Barrios, Brayden D Jolley, and James W Navalta. Validity and reliability of physiological data in applied settings measured by wearable technology: A rapid systematic review. *Technologies*, 8(4):70, 2020.

[20] Pak-Hei Chan, Chun-Ka Wong, Yukkee C Poh, Louise Pun, Wangie Wan-Chiu Leung, Yu-Fai Wong, Michelle Man-Ying Wong, Ming-Zher Poh, Daniel Wai-Sing Chu, and Chung-Wah Siu. Diagnostic performance of a smartphone-based photoplethysmographic application for atrial fibrillation screening in a primary care setting. *Journal of the American Heart Association*, 5(7):e003428, 2016.

[21] Pradyumna Chari, Krish Kabra, Doruk Karinca, Soumyarup Lahiri, Diplav Srivastava, Kimaya Kulkarni, Tianyuan Chen, Maxime Cannesson, Laleh Jalilian, and Achuta Kadambi. Diverse r-ppg: Camera-based heart rate estimation for diverse subject skin-tones and scenes. *arXiv preprint arXiv:2010.12769*, 2020.

[22] Peter H. Charlton, Timothy Bonnici, Lionel Tarassenko, David A. Clifton, Richard Beale, and Peter J. Watkinson. An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram. *Physiological Measurement*, 2016.

[23] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, 2015.

[24] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated end-to-end optimizing compiler for deep learning. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 578–594, 2018.

[25] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated end-to-end optimizing compiler for deep learning. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 578–594, 2018.

[26] Weixuan Chen and Daniel McDuff. Deepmag: Source specific motion magnification using gradient ascent. *arXiv preprint arXiv:1808.03338*, 2018.

[27] Weixuan Chen and Daniel McDuff. Deepphys: Video-based physiological measurement using convolutional attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–365, 2018.

[28] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

[29] Janghoon Choi, Junseok Kwon, and Kyoung Mu Lee. Deep meta learning for real-time target-aware visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 911–920, 2019.

[30] Thomas Coppetti, Andreas Brauchlin, Simon Müggler, Adrian Attinger-Toller, Christian Templin, Felix Schönrath, Jens Hellermann, Thomas F Lüscher, Patric Biaggi, and Christophe A Wyss. Accuracy of smartphone apps for heart rate measurement. *European journal of preventive cardiology*, 24(12):1287–1293, 2017.

[31] Theodore Curran, Xin Liu, Daniel McDuff, Shwetak Patel, and Eugene Yang. Camera-based remote photoplethysmography to measure heart rate and blood pressure in ambulatory patients with cardiovascular disease: Preliminary analysis. *Journal of the American College of Cardiology*, 81(8_Supplement):2301–2301, 2023.

[32] Ananyananda Dasari, Sakthi Kumar Arul Prakash, László A Jeni, and Conrad S Tucker. Evaluation of biases in remote photoplethysmography methods. *NPJ digital medicine*, 4(1):1–13, 2021.

[33] Gerard De Haan and Vincent Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, 2013.

[34] Gerard De Haan and Arno Van Leest. Improved motion robustness of remote-ppg by using the blood volume pulse signature. *Physiological measurement*, 35(9):1913, 2014.

[35] Eugene d'Eon, David Luebke, and Eric Enderton. Efficient rendering of human skin. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 147–157. Eurographics Association, 2007.

[36] Felipe I. Donoso, Rosa L. Figueroa, Eduardo A. Lecannelier, Esteban J. Pino, and Alejandro J. Rojas. Atrial activity selection for atrial fibrillation ECG recordings. *Computers in Biology and Medicine*, 43(10):1628–1636, oct 2013.

[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[38] Paul Ekman, Wallace V Friesen, and John Hager. *Facial action coding system: A technique for the measurement of facial movement*. Research Nexus, Salt Lake City, UT, 2002.

[39] Justin R Estepp, Ethan B Blackford, and Christopher M Meier. Recovering pulse rate during motion artifact with a multi-imager array for non-contact imaging photoplethysmography. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1462–1469. IEEE, 2014.

[40] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[42] Thomas B Fitzpatrick. The validity and practicality of sun-reactive skin types i through vi. *Archives of dermatology*, 124(6):869–871, 1988.

[43] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[44] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. Metasense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 110–123, 2019.

[45] Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

[46] Alessandro R Guazzi, Mauricio Villarroel, Joao Jorge, Jonathan Daly, Matthew C Frise, Peter A Robbins, and Lionel Tarassenko. Non-contact measurement of oxygen saturation with an rgb camera. *Biomedical optics express*, 6(9):3320–3338, 2015.

[47] Robert M Haralick. Performance characterization in computer vision. In *BMVC92*, pages 1–8. Springer, 1992.

[48] Junfeng He, Khoi Pham, Nachiappan Valliappan, Pingmei Xu, Chase Roberts, Dmitry Lagun, and Vidhya Navalpakkam. On-device few-shot personalization for real-time gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[49] Javier Hernandez, Daniel J McDuff, and Rosalind W Picard. Bioinsights: Extracting personal data from "still" wearable motion sensors. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6. IEEE, 2015.

[50] Guillaume Heusch, André Anjos, and Sébastien Marcel. A reproducible study on remote heart rate measurement. *arXiv preprint arXiv:1709.00962*, 2017.

[51] Manish Hosanee, Gabriel Chan, Kaylie Welykholowa, Rachel Cooper, Panayiotis A Kyriacou, Dingchang Zheng, John Allen, Derek Abbott, Carlo Menon, Nigel H Lovell, et al. Cuffless single-site photoplethysmography for blood pressure monitoring. *Journal of Clinical Medicine*, 9(3):723, 2020.

[52] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[53] Gee-Sern Hsu, ArulMurugan Ambikapathi, and Ming-Shiang Chen. Deep learning with time-frequency representation for pulse estimation from facial videos. In *2017 IEEE international joint conference on biometrics (IJCB)*, pages 383–389. IEEE, 2017.

[54] Sinh Huynh, Rajesh Krishna Balan, JeongGil Ko, and Youngki Lee. Vitamon: measuring heart rate variability using smartphone front camera. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 1–14, 2019.

[55] Ziheng Jiang, Tianqi Chen, and Mu Li. Efficient deep learning inference on edge devices. *ACM SysML*, 2018.

[56] Jorge Jimenez, Timothy Scully, Nuno Barbosa, Craig Donner, Xenxo Alvarez, Teresa Vieira, Paul Matts, Verónica Orvalho, Diego Gutierrez, and Tim Weyrich. A practical appearance model for dynamic facial color. *ACM Transactions on Graphics (TOG)*, 29(6):141, 2010.

[57] Jorge Jimenez, David Whelan, Veronica Sundstedt, and Diego Gutierrez. Real-time realistic skin translucency. *IEEE Computer Graphics and Applications*, 30(4):32–41, 2010.

[58] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[59] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[60] Mikhail Kopeliovich and Mikhail Petrushan. Color signal processing methods for webcam-based heart rate evaluation. In *Proceedings of SAI Intelligent Systems Conference*, pages 703–723. Springer, 2019.

[61] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2093–2102, 2018.

[62] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[63] Antony Lam and Yoshinori Kuno. Robust heart rate measurement from video using select random patches. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3640–3648, 2015.

[64] Eugene Lee, Evan Chen, and Chen-Yi Lee. Meta-rppg: Remote heart rate estimation using a transductive meta-learner. In *European Conference on Computer Vision*, pages 392–409. Springer, 2020.

[65] Jessica Lee, Deva Ramanan, and Rohit Girdhar. Metapix: Few-shot video retargeting. *arXiv preprint arXiv:1910.04742*, 2019.

[66] Juhyun Lee, Nikolay Chirkov, Ekaterina Ignasheva, Yury Pisarchyk, Mogan Shieh, Fabio Riccardi, Raman Sarokin, Andrei Kulik, and Matthias Grundmann. On-device neural net inference with mobile gpus, 2019.

[67] Xiaobai Li, Iman Alikhani, Jingang Shi, Tapio Seppanen, Juhani Junttila, Kirsi Majamaa-Voltti, Mikko Tulppo, and Guoying Zhao. The obf database: A large face video database for remote physiological signal measurement and atrial fibrillation detection. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 242–249. IEEE, 2018.

[68] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[69] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019.

[70] Xin Liu, Josh Fromm, Shwetak Patel, and Daniel McDuff. Multi-task temporal shift attention networks for on-device contactless vitals measurement. *arXiv preprint arXiv:2006.03790*, 2020.

[71] Xin Liu, Brian Hill, Ziheng Jiang, Shwetak Patel, and Daniel McDuff. Efficientphys: Enabling simple, fast and accurate camera-based cardiac measurement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5008–5017, 2023.

[72] Xin Liu, Ziheng Jiang, Josh Fromm, Xuhai Xu, Shwetak Patel, and Daniel McDuff. Metaphys: few-shot adaptation for non-contact physiological measurement. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 154–163, 2021.

[73] Xin Liu, Ziheng Jiang, Josh Fromm, Xuhai Xu, Shwetak Patel, and Daniel McDuff. MetaPhys: Few-Shot Adaptation for Non-Contact Physiological Measurement. *arXiv:2010.01773 [cs]*, March 2021. arXiv: 2010.01773.

[74] Xin Liu, Yuntao Wang, Sinan Xie, Xiaoyu Zhang, Zixian Ma, Daniel McDuff, and Shwetak Patel. Mobilephys: Personalized mobile camera-based contactless physiological sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(1), mar 2022.

[75] Xin Liu, Mingchuan Zhang, Ziheng Jiang, Shwetak Patel, and Daniel McDuff. Federated remote physiological measurement with imperfect data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2155–2164, 2022.

[76] Xin Liu, Xiaoyu Zhang, Girish Narayanswamy, Yuzhe Zhang, Yuntao Wang, Shwetak Patel, and Daniel McDuff. Deep physiological sensing toolbox. *arXiv preprint arXiv:2210.00716*, 2022.

[77] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[78] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv:2103.14030 [cs]*, March 2021. arXiv: 2103.14030.

[79] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[80] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[81] Hao Lu, Hu Han, and S Kevin Zhou. Dual-GAN: Joint BVP and Noise Modeling for Remote Physiological Measurement. page 10.

[82] Hao Lu, Hu Han, and S Kevin Zhou. Dual-gan: Joint bvp and noise modeling for remote physiological measurement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12404–12413, 2021.

[83] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, 2019.

[84] Daniel McDuff. Camera measurement of physiological vital signs. *ACM Computing Surveys (CSUR)*, 2021.

[85] Daniel McDuff and Ethan Blackford. iphys: An open non-contact imaging-based physiological measurement toolbox. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6521–6524. IEEE, 2019.

[86] Daniel McDuff, Ethan B Blackford, and Justin R Estepp. The impact of video compression on remote cardiac pulse measurement using imaging photoplethysmography. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 63–70. IEEE, 2017.

[87] Daniel McDuff, Roger Cheng, and Ashish Kapoor. Identifying bias in ai using simulation. 2018.

[88] Daniel McDuff, Sarah Gontarek, and Rosalind W Picard. Remote detection of photoplethysmographic systolic and diastolic peaks using a digital camera. *IEEE Transactions on Biomedical Engineering*, 61(12):2948–2954, 2014.

[89] Daniel McDuff, Sarah Gontarek, and Rosalind W. Picard. Remote Detection of Photoplethysmographic Systolic and Diastolic Peaks Using a Digital Camera. *IEEE Transactions on Biomedical Engineering*, 61(12):2948–2954, December 2014.

[90] Daniel McDuff, Javier Hernandez, Erroll Wood, Xin Liu, and Tadas Baltrusaitis. Using high-fidelity avatars to advance camera-based cardiac pulse measurement. *Transactions on Biomedical Engineering*, 2020.

[91] Daniel McDuff, Xin Liu, Javier Hernandez, Erroll Wood, and Tadas Baltrusaitis. Synthetic data for multi-parameter camera-based physiological sensing. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2021.

[92] Daniel McDuff, Miah Wander, Xin Liu, Brian L Hill, Javier Hernandez, Jonathan Lester, and Tadas Baltrusaitis. Scamps: Synthetics for camera measurement of physiological signals. *arXiv preprint arXiv:2206.04197*, 2022.

[93] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[94] Rita Meziatisabour, Yannick Benezeth, Pierre De Oliveira, Julien Chappe, and Fan Yang. Ubfc-phys: A multimodal database for psychophysiological studies of social stress. *IEEE Transactions on Affective Computing*, 2021.

[95] Benjamin W Nelson and Nicholas B Allen. Accuracy of consumer wearable heart rate measurement during an ecologically valid 24-hour period: intraindividual validation study. *JMIR mHealth and uHealth*, 7(3):e10828, 2019.

[96] Xuesong Niu, Hu Han, Shiguang Shan, and Xilin Chen. Vipl-hr: A multi-modal database for pulse estimation from less-constrained face video. In *Asian Conference on Computer Vision*, pages 562–576. Springer, 2018.

[97] Xuesong Niu, Shiguang Shan, Hu Han, and Xilin Chen. Rhythmnet: End-to-end heart rate estimation from face via spatial-temporal representation. *IEEE Transactions on Image Processing*, 29:2409–2423, 2019.

[98] Xuesong Niu, Zitong Yu, Hu Han, Xiaobai Li, Shiguang Shan, and Guoying Zhao. Video-based remote physiological measurement via cross-verified feature disentangling. In *European Conference on Computer Vision*, pages 295–310. Springer, 2020.

[99] Ewa M Nowara, Daniel McDuff, and Ashok Veeraraghavan. A meta-analysis of the impact of skin tone and gender on non-contact photoplethysmography measurements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 284–285, 2020.

[100] Ewa M Nowara, Daniel McDuff, and Ashok Veeraraghavan. Combining magnification and measurement for non-contact cardiac monitoring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3810–3819, 2021.

[101] Ewa Magdalena Nowara, Tim K Marks, Hassan Mansour, and Ashok Veeraraghavan. Sparseppg: towards driver monitoring using camera-based vital signs estimation in near-infrared. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1353–135309. IEEE, 2018.

[102] Joonas Paalasmaa, Hannu Toivonen, and Markku Partinen. Adaptive Heartbeat Modeling for Beat-to-Beat Heart Rate Measurement in Ballistocardiograms. 2014.

[103] Amruta Pai, Ashok Veeraraghavan, and Ashutosh Sabharwal. Camerahrv: robust measurement of heart rate variability using a camera. In *Optical Diagnostics and Sensing XVIII: Toward Point-of-Care Diagnostics*, volume 10501, page 105010S. International Society for Optics and Photonics, 2018.

[104] Eunbyung Park and Alexander C Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 569–585, 2018.

[105] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[106] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[107] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1278–1286, 2015.

[108] Christian Pilz. On the vector space in photoplethysmography imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[109] Christian S Pilz, Sebastian Zaunseder, Jarek Krajewski, and Vladimir Blazek. Local group invariance for heart rate estimation from face videos in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1254–1262, 2018.

[110] Ming-Zher Poh, Daniel McDuff, and Rosalind Picard. A medical mirror for non-contact health monitoring. In *ACM SIGGRAPH 2011 Emerging Technologies*, pages 1–1. 2011.

[111] Ming-Zher Poh, Daniel McDuff, and Rosalind W Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE transactions on biomedical engineering*, 58(1):7–11, 2010.

[112] Ming-Zher Poh, Daniel McDuff, and Rosalind W Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Optics express*, 18(10):10762–10774, 2010.

[113] Kim Pollock, Michael Setzen, and Peter F Svider. Embracing telemedicine into your otolaryngology practice amid the covid-19 crisis: An invited commentary. *American Journal of Otolaryngology*, page 102490, 2020.

[114] Adnan Qayyum, Kashif Ahmad, Muhammad Ahtazaz Ahsan, Ala Al-Fuqaha, and Junaid Qadir. Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge. *arXiv preprint arXiv:2101.07511*, 2021.

[115] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4318–4327, 2020.

[116] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *European Conference on Computer Vision*, pages 909–916. Springer, 2016.

[117] Joel D Reece, Jennifer A Bunn, Minsoo Choi, and James W Navalta. Assessing heart rate using consumer technology association standards. *Technologies*, 9(3):46, 2021.

[118] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

[119] Jared Roesch, Steven Lyubomirsky, Logan Weber, Josh Pollock, Marisa Kirisame, Tianqi Chen, and Zachary Tatlock. Relay: A new ir for machine learning frameworks.

In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 58–68, 2018.

[120] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[121] Nadav Rotem, Jordan Fix, Saleem Abdulrasool, Garret Catron, Summer Deng, Roman Dzhabarov, Nick Gibson, James Hegeman, Meghan Lele, Roman Levenstein, Jack Montgomery, Bert Maher, Satish Nadathur, Jakob Olesen, Jongsoo Park, Artem Rakhov, Misha Smelyanskiy, and Man Wang. Glow: Graph lowering compiler techniques for neural networks, 2019.

[122] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635. Springer, 2018.

[123] Anna Shcherbina, C Mikael Mattsson, Daryl Waggott, Heidi Salisbury, Jeffrey W Christle, Trevor Hastie, Matthew T Wheeler, and Euan A Ashley. Accuracy in wrist-worn, sensor-based measurements of heart rate and energy expenditure in a diverse cohort. *Journal of personalized medicine*, 7(2):3, 2017.

[124] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011.

[125] Anthony C Smith, Emma Thomas, Centaine L Snoswell, Helen Haydon, Ateev Mehrotra, Jane Clemensen, and Liam J Caffery. Telehealth for global emergencies: Implications for coronavirus disease 2019 (COVID-19). *Journal of telemedicine and telecare*, page 1357633X20916567, 2020.

[126] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

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

[128] Rencheng Song, Huan Chen, Juan Cheng, Chang Li, Yu Liu, and Xun Chen. Pulsegan: Learning to generate realistic pulse waveforms in remote photoplethysmography. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1373–1384, 2021.

[129] Rencheng Song, Senle Zhang, Chang Li, Yunfei Zhang, Juan Cheng, and Xun Chen. Heart rate estimation from facial videos using a spatiotemporal representation with convolutional neural networks. *IEEE Transactions on Instrumentation and Measurement*, 2020.

[130] Xuan Song, Xinyan Liu, and Chunting Wang. The role of telemedicine during the covid-19 epidemic in china—experience from shandong province, 2020.

[131] Radim Špetlík, Vojtech Franc, and Jirí Matas. Visual heart rate estimation with convolutional neural network. In *Proceedings of the British Machine Vision Conference, Newcastle, UK*, pages 3–6, 2018.

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

[133] Lech Świrski and Neil Dodgson. Rendering synthetic ground truth images for eye tracker evaluation. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 219–222, 2014.

[134] Chihiro Takano and Yuji Ohta. Heart rate measurement based on a time-lapse image. *Medical engineering & physics*, 29(8):853–857, 2007.

[135] L Tarassenko, M Villarroel, A Guazzi, J Jorge, DA Clifton, and C Pugh. Non-contact video-based vital sign monitoring using ambient light and auto-regressive models. *Physiological measurement*, 35(5):807, 2014.

[136] Mika P Tarvainen, Perttu O Ranta-Aho, and Pasi A Karjalainen. An advanced detrending method with application to hrv analysis. *IEEE Transactions on Biomedical Engineering*, 49(2):172–175, 2002.

[137] H Emrah Tasli, Amogh Gudi, and Marten Den Uyl. Remote ppg based vital sign measurement using adaptive facial regions. In *2014 IEEE international conference on image processing (ICIP)*, pages 1410–1414. IEEE, 2014.

[138] Yun-Yun Tsou, Yi-An Lee, Chiou-Ting Hsu, and Shang-Hung Chang. Siamese-rppg network: Remote photoplethysmography signal estimation from face videos. In *Proceedings of the 35th annual ACM symposium on applied computing*, pages 2066–2073, 2020.

[139] Thomas Vandal, Daniel McDuff, and Rana El Kaliouby. Event detection: Ultra large-scale clustering of facial expressions. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–8. IEEE, 2015.

[140] David Vazquez, Antonio M Lopez, Javier Marin, Daniel Ponsa, and David Geronimo. Virtual and real world adaptation for pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):797–809, 2014.

[141] VSR Veeravasarapu, Rudra Narayan Hota, Constantin Rothkopf, and Ramesh Visvanathan. Model validation for vision systems via graphics simulation. *arXiv preprint arXiv:1512.01401*, 2015.

[142] VSR Veeravasarapu, Rudra Narayan Hota, Constantin Rothkopf, and Ramesh Visvanathan. Simulations for validation of vision systems. *arXiv preprint arXiv:1512.01030*, 2015.

[143] VSR Veeravasarapu, Constantin Rothkopf, and Visvanathan Ramesh. Model-driven simulations for deep convolutional neural networks. *arXiv preprint arXiv:1605.09582*, 2016.

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

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

[146] Mauricio Villarroel, Sitthichok Chaichulee, João Jorge, Sara Davis, Gabrielle Green, Carlos Arteta, Andrew Zisserman, Kenny McCormick, Peter Watkinson, and Lionel Tarassenko. Non-contact physiological monitoring of preterm infants in the neonatal intensive care unit. *npj Digital Medicine*, 2(1):1–18, 2019.

[147] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[148] Wenjin Wang, Albertus C den Brinker, Sander Stuijk, and Gerard de Haan. Algorithmic principles of remote ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2016.

[149] Wenjin Wang, Albertus C. den Brinker, Sander Stuijk, and Gerard de Haan. Algorithmic Principles of Remote PPG. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, July 2017. Conference Name: IEEE Transactions on Biomedical Engineering.

[150] Wenjin Wang, Albertus C den Brinker, Sander Stuijk, and Gerard de Haan. Algorithmic principles of remote ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2017.

[151] Wenjin Wang, Sander Stuijk, and Gerard De Haan. Exploiting spatial redundancy of image sensor for motion robust rppg. *IEEE transactions on Biomedical Engineering*, 62(2):415–425, 2014.

[152] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3681–3691, 2021.

[153] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.

[154] Tete Xiao, Quanfu Fan, Dan Gutfreund, Mathew Monfort, Aude Oliva, and Bolei Zhou. Reasoning about human-object interactions through dual attention networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3919–3928, 2019.

[155] Shuchang Xu, Lingyun Sun, and Gustavo Kunde Rohde. Robust efficient estimation of heart rate pulse from video. *Biomedical optics express*, 5(4):1124–1135, 2014.

[156] Zhe Xu, Lei Shi, Yijin Wang, Jiyuan Zhang, Lei Huang, Chao Zhang, Shuhong Liu, Peng Zhao, Hongxia Liu, Li Zhu, et al. Pathological findings of COVID-19 associated with acute respiratory distress syndrome. *The Lancet respiratory medicine*, 8(4):420–422, 2020.

[157] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[158] Zitong Yu, Xiaobai Li, Pichao Wang, and Guoying Zhao. Transrppg: Remote photoplethysmography transformer for 3d mask face presentation attack detection. *IEEE Signal Processing Letters*, 2021.

[159] Zitong Yu, Wei Peng, Xiaobai Li, Xiaopeng Hong, and Guoying Zhao. Remote heart rate measurement from highly compressed facial videos: an end-to-end deep learning solution with video enhancement. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 151–160, 2019.

[160] Zitong Yu, Yuming Shen, Jingang Shi, Hengshuang Zhao, Philip Torr, and Guoying Zhao. Physformer: Facial video-based physiological measurement with temporal difference transformer. *arXiv preprint arXiv:2111.12082*, 2021.

[161] Zitong Yu, Yuming Shen, Jingang Shi, Hengshuang Zhao, Philip HS Torr, and Guoying Zhao. Physformer: facial video-based physiological measurement with temporal difference transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4186–4196, 2022.

[162] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

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

[164] Ying-Ying Zheng, Yi-Tong Ma, Jin-Ying Zhang, and Xiang Xie. COVID-19 and the cardiovascular system. *Nature Reviews Cardiology*, 17(5):259–260, 2020.

[165] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.