FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Nhi Nguyen

HEART RATES ESTIMATION USING RPPG METHODS IN CHALLENGING IMAGING CONDITIONS

Master's Thesis
Degree Programme in Computer Science and Engineering
May 2023

Nguyen N. (2023) Heart Rates Estimation Using RPPG Methods in Challenging Imaging Conditions. University of Oulu, Degree Programme in Computer Science and Engineering, 61 p.

ABSTRACT

The cardiovascular system plays a crucial role in maintaining the body's equilibrium by regulating blood flow and oxygen supply to different organs and tissues. While contact-based techniques like electrocardiography and photoplethysmography are commonly used in healthcare and clinical monitoring, they are not practical for everyday use due to their skin contact requirements. Therefore, non-contact alternatives like remote photoplethysmography (rPPG) have gained significant attention in recent years. However, extracting accurate heart rate information from rPPG signals under challenging imaging conditions, such as image degradation and occlusion, remains a significant challenge. Therefore, this thesis aims to investigate the effectiveness of rPPG methods in extracting heart rate information from rPPG signals in these imaging conditions. It evaluates the effectiveness of both traditional rPPG approaches and rPPG pre-trained deep learning models in the presence of real-world image transformations, such as occlusion of the faces by sunglasses or facemasks, as well as image degradation caused by noise artifacts and motion blur. The study also explores various image restoration techniques to enhance the performance of the selected rPPG methods and experiments with various fine-tuning methods of the best-performing pre-trained model. The research was conducted on three databases, namely UBFC-rPPG, UCLA-rPPG, and UBFC-Phys, and includes comprehensive experiments. The results of this study offer valuable insights into the efficacy of rPPG in practical scenarios and its potential as a non-contact alternative to traditional cardiovascular monitoring techniques.

Keywords: rPPG, HR, POS, EfficientPhys, Challenging Imaging

TABLE OF CONTENTS

FOREWORD

Writing a master's thesis is a defining moment in any academic journey. It signifies the culmination of years of diligent work, unyielding dedication, and unwavering perseverance in pursuing a specialized field of study. The research, analysis, experiments, and findings presented in this thesis are a testament to my passion and commitment.

The research topic for this thesis was suggested by Dr. Le Nguyen, with the aim of studying the performance of rPPG methods in the presence of real-world image transformations and exploring various image restoration techniques to enhance the performance of the selected rPPG method.

I would like to express my gratitude to my thesis advisor, Associate Professor. Miguel Bordallo López, for his unwavering support and guidance throughout the research process. I am also grateful to the faculty and staff of the University of Oulu for providing an environment that fosters academic excellence and research. I would like to extend a special thanks to Dr. Le Nguyen for his detailed instructions on the experiments and for proposing several ideas that were instrumental in improving the experimental results.

Finally, I would like to acknowledge the support and encouragement of my family and friends, whose unwavering support has been an inspiration to me throughout this journey. I am deeply honored to present this thesis and hope that it will serve as a valuable contribution to the academic community.

Oulu, May 1st, 2023

Nhi Nguyen

LIST OF ABBREVIATIONS AND SYMBOLS

rPPG	remote Photoplethysmography
HR	Heart Rate
RR	Respiration Rate
DFM	Dichromatic Reflection Model
ROI	Region of Interest
RGB	Red-Green-Blue
BSS	Blind Source Separation
POS	Plane-Orthogonal-to-Skin
DL	Deep Learning
ST-rPPG	Spatial Temporal rPPG
FPS	Frame-Per-Second
MTCNN	Multi-Task Cascaded Convolutional Network
bpm	beat-per-minute
MAE	Mean Absolute Error
RMSE	Root Mean Absolute Error
MAPE	Mean Absolute Percentage Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
t	time
$\mathbf{C_k}(t)$	RGB channels corresponding to the skin-pixel at time t and index k
$\mathbf{I(t)}$	luminance intensity
$\boldsymbol{v}_s(t)$	specular reflection
$\boldsymbol{v}_d(t)$	diffuse reflection
$\boldsymbol{v}_n(t)$	optical sensor's quantization noise
μ	mean value
σ	standard deviation value
e	mathematical constant e
π	mathematical constant pi
ρ	Pearson Correlation Coefficient
$x(t)$	input signal
$S(t)$	rPPG signal
σ^2	variance
$\lfloor \rfloor$	integer part
$\vee$	logical OR operation

1. INTRODUCTION

1.1. Background

The heart, blood vessels, and blood are what construct the cardiovascular system, whose principal purpose is to deliver nutrients and rich oxygen blood to all regions of the body while returning deoxygenated blood to the lungs. The human cardiovascular system plays a vital role in maintaining homeostasis within the body by regulating blood flow and oxygenation to various organs and tissues. This system's activity is employed in healthcare and clinical monitoring to detect vital signs such as heart rate (HR), oxygen saturation, respiration rate (RR), and blood pressure. Additionally, it is also used in the diagnosis of peripheral vascular diseases. A healthy adult usually has a pulse rate ranging from 60 to 100 beats per minute, a RR ranging from 12 to 20 breath cycles (contains an inhale and exhale), 95% or higher of SpO2 (a ratio between oxygenated hemoglobin and deoxygenated hemoglobin) and pressure of blood against the artery walls during the heart beats less than 120/80 mmHg [1].

There are various methods for detecting cardiovascular activity including contact-based and contactless methods. Contact-based methods involve the direct application of sensors to the skin or other surfaces in contact with the body, such as electrocardiography (ECG), photoplethysmography (PPG), ballistocardiograph (BCG), and impedance cardiography (ICG). Of all the contact-based techniques mentioned above, the information wave of ECG and PPG is most related to HR monitoring. ECG is a well-established technique that measures cardiac activity. It works by detecting the electrical impulses generated by the heart's polarization and depolarization and translating these impulses into a waveform corresponding to the HR. Its machines directly use electrical signals brought out by heart activity, which is the reason why they have electrodes connected to the chest. ECG signal is seen as reference information for measuring HR, so its accuracy is certainly higher. While PPG is a popular non-invasive contact-based method used for measuring several human vital signs including pulse rate, oxygen saturation, and respiration [2]. Utilizing a light source and a sensor, it gauges fluctuations in the amount of blood in the skin which is more efficient in size and price than ECG. Despite the fact that ECG and PPG are based on quite distinct technologies, one electrical and the other optical, the procedure of diagnosing abnormal heart rhythms might be practically identical. For example, after obtaining an accurate stream of RR intervals, as shown in Fig. 1, they can be processed to determine statistical distributions associated with atrial fibrillation.

Although both methods are extensively utilized and dependable in the medical industry, their devices are not something that ordinary people or normal families own, especially multichannel ECG devices which require adhesive electrodes to be attached to the chest. This can be considered a disadvantage because there are many situations where skin contact has to be prevented such as newborns, burn patients who have very sensitive skin, or during pandemics occurs, which makes person-to-person interaction poses a significant risk of transmission of a virus. Therefore, replacing contact-based technologies with non-contact alternatives would be advantageous.

Contactless methods for measuring HR, such as radar, thermal imaging, and remote photoplethysmography (rPPG), have gained significant attention in recent years as they do not require direct contact with the body. Among these methods, rPPG stands out as

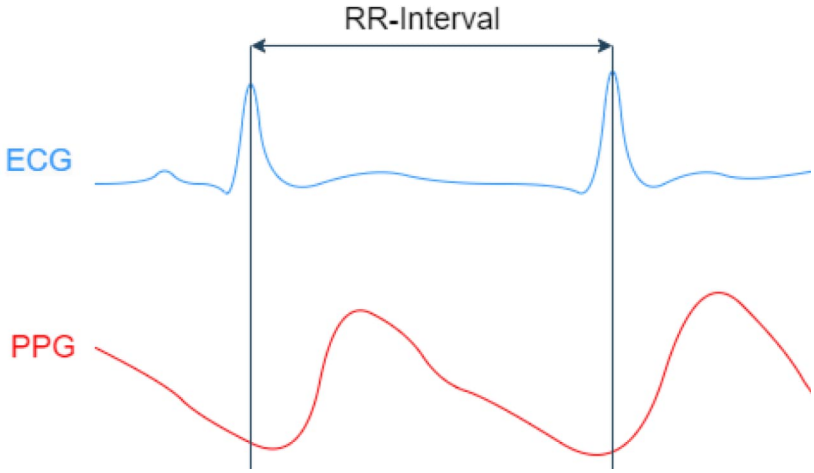


Figure 1. Examples of an ECG wave and PPG wave.

the most practical and accessible option, as it only requires a camera, such as a webcam or a smartphone camera, and a light source, such as a LED, to record videos of the skin. This makes it a more cost-effective and less device-dependent option compared to radar and thermal imaging, which require specialized equipment. This accessibility is further reinforced by the widespread availability of smartphones, as a recent study [3] found that the ownership of smartphones in high-income countries in North America and Europe has surpassed 80%, while rates in low and middle-income countries are on the rise. As a result, rPPG is an attractive option for monitoring HR in a wide range of settings, including at home and in remote areas. Furthermore, rPPG can be combined with other physiological information, such as video, audio, and temperature, to provide a more comprehensive picture of an individual's health. These advantages have motivated significant research interest in rPPG in recent years, as demonstrated in various studies [4, 5, 6]. The focus on developing accurate, reliable, and practical rPPG methodologies and technologies has been a key area of interest, with a range of techniques being proposed and evaluated. Despite this focus, accurately extracting HR information from rPPG signals still be a significant challenge under challenging image conditions, including image degradation and occlusion. Unfortunately, limited research has been conducted in this area, making it difficult to develop effective solutions.

This research thesis seeks to explore the effectiveness of different rPPG methods in extracting HR information from rPPG signals under challenging image conditions. Specifically, the study aims to evaluate the effectiveness of both traditional rPPG methods and rPPG pre-trained deep learning (DL) models in the presence of common image transformations encountered in real-world environments, such as occlusion of the face by sunglasses or facemasks, as well as image degradation caused by noise artifacts and motion blur. The study also investigates various restoration techniques to improve the performance of the selected rPPG method.

1.2. Research Motivation

In recent years, the rPPG method has been proposed as a way to estimate HR in real-world environments, such as using video data captured by a smartphone camera. The field of rPPG has gained significant attention in recent years, with potential applications in healthcare, sports, and human-computer interaction. However, extracting accurate HR information from rPPG signals under challenging image conditions, such as image degradation and occlusion, remains a significant challenge. Traditional rPPG methods have shown limited success in such scenarios, and the performance of DL-based approaches in these conditions is not well understood. Therefore, this master's thesis aims to investigate the effectiveness of various rPPG methods in extracting HR information from rPPG signals obtained under challenging image conditions. Specifically, the study focuses on evaluating the effectiveness of traditional-based and DL-based methods in the presence of common image transformations encountered in real-world environments. Among DL methods, this thesis focuses on pre-trained rPPG models that have been shown to achieve the most significant improvements in accuracy and robustness in normal rPPG datasets. The study not only evaluates the performance of selected rPPG methods but also explores the efficacy of various image restoration techniques in improving their performance. Additionally, The thesis experiments with different fine-tuning methods for the best-performing pre-trained model. The findings of this study can have significant implications for the design of remote physiological monitoring systems, especially in scenarios where real-time monitoring is essential. By developing and evaluating effective rPPG methods under challenging image conditions, this study can help enhance the reliability and accuracy of non-invasive physiological monitoring, thus opening up new avenues for healthcare, sports, and human-computer interaction.

1.3. Contributions

Our thesis research is designed to provide valuable insights and guidance for researchers and practitioners in this field, by investigating the effectiveness of rPPG methods in extracting heart rate information from rPPG signals in challenging imaging conditions. It evaluates the effectiveness of both traditional rPPG approaches and rPPG pre-trained deep learning models in the presence of real-world image transformations, such as occlusion of the faces by sunglasses or facemasks, as well as image degradation caused by noise artifacts and motion blur. The study also explores various image restoration techniques to enhance the performance of the selected rPPG methods and experiments with various fine-tuning methods of the best-performing pre-trained model. The research was conducted on three databases, namely UBFC-rPPG, UCLA-rPPG, and UBFC-Phys, and includes comprehensive experiments.

This master's thesis contributes to the advancement of remote physiological monitoring, which can have significant implications for improving healthcare and enhancing human performance in various domains. The findings of this study can also inform the design of new and more effective remote physiological monitoring systems, thus promoting the widespread adoption of this technology. The main results

and scientific findings of the thesis were compiled in a manuscript [7] available at arXiv, and intended to be published:

– Nguyen, N., Nguyen, L., Alvarez Casado, C., Silven, O., & Bordallo Lopez, M. (2023) Non-Contact Heart Rate Measurement from Deteriorated Videos. arXiv preprint arXiv: 2304.14789

1.4. Thesis Structure

The thesis is structured as follows: Chapter 2 provides a review of rPPG methods, encompassing traditional and DL approaches. Chapter 3 details the implementation of these methods for evaluating video data and signals, and introduces a set of transformation methods that are commonly encountered in practice. Additionally, we experiment with various image restoration methods that can be used to improve the performance of the selected rPPG method in these types of videos. In Chapter 4, we present a comparison of experimental results on three rPPG databases, along with a discussion of the findings. Finally, Chapter 5 summarizes the thesis work and outlines potential avenues for future research.

2. RELATED WORK

This section presents an overview of the key studies that have been conducted previously on the topic of rPPG. The aim of this review is to provide a comprehensive background for the current thesis work by summarizing the most important findings and contributions of previous research, this section serves as a foundation for the thesis investigation and analysis. The literature review divides the previous rPPG methods into two main categories: traditional and DL methods based on the crucial information that these techniques rely on.

2.1. Traditional Methods

2.1.1. Basic Framework

Typically, the conventional approaches to measuring HR follow a similar structure, as shown in Fig. 2. The process begins with the acquisition of a video of a human's skin area, which is captured via an imaging sensor like a digital camera, smartphone, or webcam. The light source can be either a dedicated light or ambient light. The resulting video is characterized by a frame rate that can range from as low as 10 frames per second to as high as 60 frames per second. Subsequently, a face detection algorithm, such as MTCNN from Facenet [8], is employed to extract the bounding box coordinates of the subject's face. It is followed by the selection of areas of interest (ROIs) within the video frames, such as the face, forehead, chest, and palm either manually or automatically, with the goal of identifying a region that possesses a robust signal for signal extraction. After selecting the ROI, the raw signals are obtained

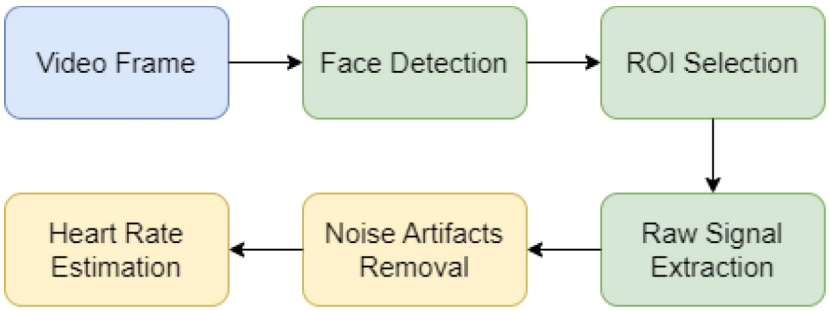


Figure 2. General framework of traditional methods for remote HR estimation Adapted and modified from [9].

through the calculation of the spatial average of the pixel values for each frame, as outlined in Equation 1. This intensity-based approach makes use of spatial averaging to improve the signal-to-noise ratio by averaging out the camera noise present in each pixel. The symbol $I(x, y, t)$ represents the intensity of the image at pixel (x, y) at time t. The average intensities of the red, green, and blue channels, represented by $i_R(t), i_G(t)$, and $i_B(t)$, respectively, are calculated by summing the intensities of all

pixels in the ROI and dividing by the number of pixels in the ROI, represented by $|ROI|$.

$$i_R(t), i_G(t), i_B(t) = \frac{\sum_{x,y \in ROI} I(x, y, t)}{|ROI|} \tag{1}$$

The raw signal from a source might contain undesirable noise from different sources like changes in illumination, skin tone, camera movement, and subject movement. Various signal processing approaches, such as low pass filtering, adaptive bandpass filtering, blind source separation, model-based methods, and signal decomposition can be employed to eliminate this noise. Vital signals such as heart rate or breathing rate can be obtained by performing frequency analysis or detecting peaks in the signal. Converting the signal into the frequency domain using a discrete Fourier transform is known as frequency analysis, which typically involves using the fast Fourier transform to calculate the frequency (F_s). In addition to the existing method, there are alternative techniques that can be utilized for signal conversion, such as the discrete cosine transform or the short-time Fourier transform. Additionally, Welch's method can be used for calculating the power spectral density. Peak detection involves calculating the number of peaks (N_s) within a defined processing period of time $T(s)$, from which the HR can be derived in minutes using the Equation 2 below:

$$\mathrm{HR} = 60 \times \mathrm{F_s} \text{ or } \mathrm{HR} = 60 \times \frac{N_s}{T} \tag{2}$$

The signal extraction step of traditional rPPG methods utilizes a dichromatic reflection model (DFM) shown in Fig. 3. According to this model, the signals obtained using photographic equipment are a blend of specular reflections, which come from surfaces, and diffuse reflections, which come from the body. However, the specular reflections that occur when light meets the skin surface do not provide any useful biological information. Therefore, rPPG techniques apply signal processing techniques to differentiate these reflections from the desired diffuse reflections that contain significant signals.

Based on the dichromatic model, the reflection of each skin pixel in a recorded image sequence can be represented as a time-varying function in the RGB channels [11]:

$$\mathbf{C_k}(t) = I(t) \cdot (\mathbf{v_s}(t) + \mathbf{v_d}(t)) + \mathbf{v_n}(t)) \tag{3}$$

Where t represents the t-th time and $\mathbf{C_k}(t)$ represents the RGB channels of the k-th skin-pixel, arranged in a column, $\mathbf{I(t)}$ represents the luminance intensity level which accounts for changes in intensity caused by both the light source and the distance between the light source, skin tissue, and camera. The luminance intensity $\mathbf{I(t)}$ is manipulated by the specular reflection $\mathbf{v_n}(t)$ and the diffuse reflection $\mathbf{v_d}(t)$ while $\mathbf{v_n}(t)$ represents the camera sensor's quantization noise.

The specular reflection refers to the reflection of light from the skin's surface that resembles a mirror and does not carry any pulsating information. The makeup of the light spectrum is similar to the light source. This reflection is influenced by the positioning between the skin surface, light source, and camera, which may change over time due to body movement. Therefore, $\mathbf{v_n}(t)$ can be represented as:

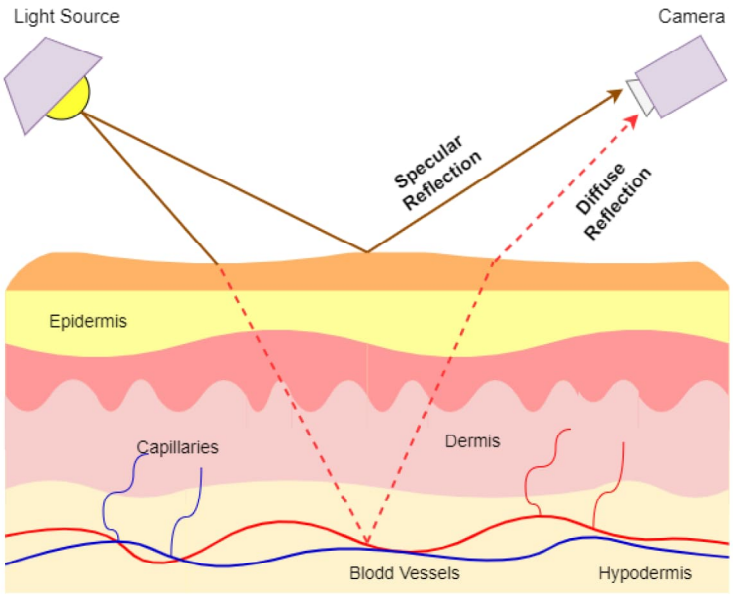


Figure 3. Dichromatic reflection model Redrawn from [10].

$$\mathbf{v_s}(t) = \mathbf{u_s} \cdot (s_0 + s(t)) \tag{4}$$

Where $\mathbf{u_s}$ refers to the color spectrum of the light source represented as a unit color vector. The terms s_0 and $s(t)$ stand for the stable and changing portions of the specular reflection, respectively. The changes in the reflection are a result of motion represented by $s(t)$.

The diffuse reflection arises due to the light being absorbed and scattered within the skin tissue. The hue of the diffuse reflection is influenced by the presence of hemoglobin and melanin in the skin tissue. It changes with variations in blood volume and is time-dependent. The expression for $\mathbf{v_d}(t)$ is represented below:

$$\mathbf{v_d}(t) = \mathbf{u_d} \cdot d_0 + \mathbf{u_p} \cdot p(t) \tag{5}$$

In the above equation, $\mathbf{u_d}$ represents the unit color vector of the skin tissue and the term d_0 represents the steady reflection strength. The relative amplitude of pulses in the RGB color space is represented by $\mathbf{u_p}$. The pulse signal is represented by $p(t)$. By inserting Equation 4 and Equation 5 into Equation 3, we obtain:

$$\mathbf{C_k}(t) = I_0 \cdot (1 + i(t)) \cdot (\mathbf{u_c} \cdot c_0 + \mathbf{u_s} \cdot s(t) + \mathbf{u_p} \cdot p(t)) + \mathbf{v_n}(t) \tag{6}$$

The variable $I(t)$ is obtained by adding a fixed component I_0 to a component that changes over time $I_0 \cdot (i(t))$. The camera observes a change in intensity corresponding to the brightness level, which is represented by $i(t)$. The signals $s(t)$ and $p(t)$ are zero-mean signals. The stable components of the specular and diffuse reflections are merged into one component that symbolizes the steady skin reflection. The unit color vector

of the skin reflection is represented by $\mathbf{u_c}$ and the reflection strength is represented by c_0. This is shown in Equation 7:

$$\mathbf{u}_\mathrm{c} \cdot c_0 = \mathbf{u}_\mathrm{s} \cdot s_0 + \mathbf{u}_\mathrm{d} \cdot d_0 \tag{7}$$

As defined in Equation 6, the purpose of a traditional rPPG method is to extract the pulse signal $p(t)$ from the combined signal $\mathbf{C_k}(t)$, and the approach to doing so can be divided into two categories based on the amount of prior information available: BSS-based and model-based methods. Model-based methods can use the color vectors of different components to guide the separation process. On the other hand, BSS-based methods can be used to separate $C(t)$ into sources to extract the pulse signal without prior information.

2.1.2. BSS-Based RPPG

BSS stands for Blind Source Separation was inspired by the early work in optical flow and computer vision, and they were designed to use the concept of background subtraction to distinguish a person's face from the background of a video and extract independent components from the color signals derived from multiple facial sub-regions. BSS methods can be divided into two categories: Conventional and Joint [12].

Conventional BSS: is a technique to extract the original, unobserved signals or sources from a set of observed mixtures without having any prior knowledge of the mixing process. In most cases, the observations are the outputs from sensors, where each output is a combination of various sources. Numerous research papers have been released utilizing BSS techniques, such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA). ICA is a commonly used method for conventional BSS, which has been shown to be effective in many applications. Poh et al. [13] proposed a JADE-based ICA algorithm to extract the HR component from RGB channel signals. The idea is that the signals for R, G, and B are formed by blending the pulse signal with other signals. The JADE algorithm eliminates connections and complex interdependencies among the RGB channels. The results showed a decrease in root mean square error (RMSE) from 19.36 to 4.63 bpm, indicating the effectiveness of ICA in HR evaluation during natural movement conditions. Balakrishnan et al. [14] presented a new way to measure HR without direct contact using a digital camera to record facial videos. They used the Viola-Jones face detector and Kanade-Lucas-Tomasi tracking algorithm to detect the face, find the region of interest, and track its feature points. The tracked points were filtered with a Butterworth filter and cleaned up with PCA to get the physiological signal. Ultimately, the rate of a pulse was determined by finding the peaks in the signal and applying a fast Fourier transform (FFT). Irani et al. [15] designed a method to extract HR using a webcam that is less susceptible to motion artifacts. This method was created to address the limitations of previous methods, which only worked with stationary subjects and did not take into account internal or external movements. The improved method considers different facial expressions and head positions and employs the discrete cosine transform (DCT) and a filter that averages over time. BSS-based methods

showed some level of tolerance towards movement but had limited improvement in handling severe movements, according to [16].

Joint BSS: Traditionally, conventional BSS techniques were developed to work with a single data set, for example separating the signals of several color channels originating from a specific facial area into individual components. In recent times, using signals from multiple subregions of the facial ROI has been adopted to improve the accuracy of HR measurement. As more data sets become accessible, various joint blind source separation (JBSS) techniques have been developed to reveal the fundamental sources in each data set while keeping the extracted sources consistently organized across all the datasets. Guo et al. [17] was the first to introduce the JBSS method in the field of rPPG, using independent vector analysis to analyze color signals from multiple facial subregions. Initial experimental findings suggested that using a discrete cosine transform (DCT)-based BSS technique yielded more precise HR measurements than using an independent component analysis (ICA)-based BSS method. Afterward, Qi et al. [18] introduced a new method for non-contact HR measurement by analyzing correlations among different subregions of the face using the JBSS technique. The results obtained by testing on a comprehensive public database demonstrated that the JBSS method proposed in this study surpassed earlier approaches that relied on ICA. A comprehensive overview of JBSS methods by Chen et al. [19] has highlighted its benefits from both multi-set and multi-modal perspectives as well as demonstrated the effectiveness and potential of JBSS as a tool for neurophysiological data analysis in various practical and multi-faceted applications. The use of JBSS methods for HR estimation is still in its early stages. In the near future, other types of multiple data sets, including those from multiple regions of the face and JBSS can be employed to achieve a more precise and dependable non-contact HR estimation, even with multiple types of data.

2.1.3. Model-Based RPPG

Model-based methods, on the other hand, were inspired by the observation that the pulsatile flow of blood in the face also causes changes in skin color. These methods use the color information in the video to estimate the pulse rate, and they are founded on the idea that changes in blood volume in the face cause variations in skin color, which can be measured from the video. Model-based methods have a common characteristic that can make use of the information contained in color vectors to regulate the separation of components, resulting in the elimination of dependence $C(t)$ on the average skin reflection color channels in the process of component derivation.

Begin with early work in the area was conducted by Verkruysse et al. [20] in the identification of the correct channel of ambient light for best results. The research reveals that while the green channel has the most valuable rPPG knowledge, the red and blue channels also possess critical rPPG information. To enhance the motion robustness of the rPPG model, a method called chrominance-based approach (CHROM) [21] was developed. This method takes into account both diffuse reflection and specular reflection that can cause changes in the color observed depending on the distance between the skin, camera, and light sources. Combining the individual red, green, and blue channels can help minimize the effects of motion disturbances.

The results from experiments showed that CHROM is more effective in dealing with exercise-related motions compared to previously used methods like ICA-based and PCA-based methods.

Following that, de Haan and van Leest [22] proposed a method to improve the motion robustness of rPPG by using a PBV-based approach. This method relies on the pattern of alteration in blood volume to distinguish between color changes associated with the pulse and those resulting from movement disruptions in the time-based RGB signals. The outcomes from experiments with subjects exercising on different fitness equipment showed that the PBV-based method performed better than the CHROM-based method. Afterward, Wang et al. [11] proposed the Plane-Orthogonal-to-Skin (POS) algorithm which involves defining a projection plane in the RGB color space that is orthogonal to skin tone, allowing for the extraction of the pulse signal. This algorithm aims to enhance the precision and robustness of HR measurement without contact by effectively separating the pulse-induced color changes from other sources of variation in the RGB data. Wang et al. also introduced an algorithm called "Spatial Subspace Rotation" (SSR) [23], which evaluates a group of skin pixels over time and determines their "rotation" to obtain the pulse. The SSR algorithm was evaluated in a laboratory environment on individuals with different skin tones and subjected to various lighting and physical activity conditions. It was found to be more effective than previous source separation techniques ICA and CHROM. In a recent study, Casado et al. [24] introduced a series of pipelines named Face2PPG for extracting rPPG from facial signals. In order to improve this pipeline, they have proposed three innovative approaches. The first involves utilizing rigid mesh normalization to stabilize the detected face. The second approach involves dynamically selecting facial regions that offer the most reliable raw signals while discarding regions that are susceptible to noise or artifacts. Lastly, the authors have introduced a new RGB to rPPG transformation technique called Orthogonal Matrix Image Transformation (OMIT) that uses QR decomposition to enhance the pipeline's robustness against compression artifacts. The introduction of these contactless measurements can greatly reduce the cost of monitoring and make it possible to be used in situations where traditional contact sensors are not suitable. Despite its potential in the future of digital healthcare, the rPPG technology still faces challenges as the signals obtained are weaker and require careful processing.

2.2. Deep Learning Methods

While the traditional techniques have shown improvement through each stage, limitations of these techniques still exist and more research is required for more accurate and robust non-contact HR measurement that is why DL methods have gained popularity in the rPPG research community due to their ability to map complex physiological processes, simplify the process, reduce the number of steps involved in processing, as well as make the process of selecting important features automatic for remote HR measurement. In the subsequent parts of this thesis, the DL techniques for remote HR measurement are classified into two groups: end-to-end and hybrid approaches as demonstrated in Fig. 4.

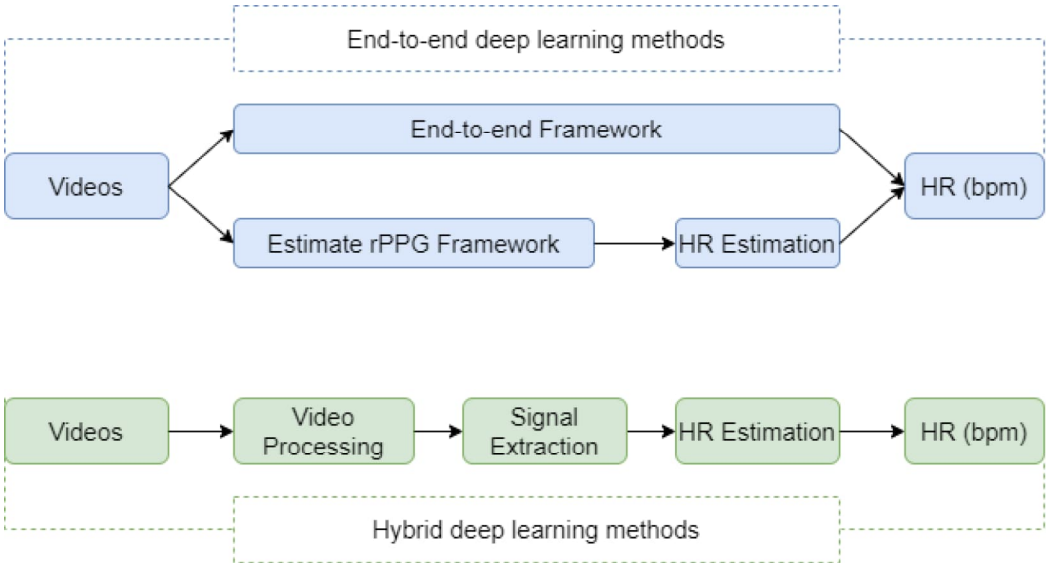


Figure 4. A graphical representation of end-to-end and hybrid DL techniques Redrawn from [25].

2.2.1. End-To-End Deep Learning Methods

A method is considered end-to-end if it directly produces the HR value without any intermediary stages and accepts a sequence of facial video frames as its input. This also includes methods that are specifically created the rPPG signal as output. Despite the convenience of end-to-end DL methods in terms of simple model optimization, they need a substantial amount of training data and can be challenging to verify. Further research is necessary to understand these models and apply them in clinical settings [26]. In the following, we talk about typical DL models in remote HR measurements such as 2D Convolutional Neural Networks (2D CNN), 3D Convolutional Neural Networks (3D CNN), 2D Convolutional Neural Networks + Recurrent Neural Networks (2D CNN + RNN).

2D Convolutional Neural Network rPPG: One of the first rPPG 2D CNN models was introduced by Spetlik et al. [27]. Their work suggested a comprehensive method for estimating HR that produced a single numerical value as its output. Their approach used two-stage CNN which incorporated one signal extractor and one HR estimator known as HR-CNN. The 2D CNN feature extractor was intentionally created to optimize the signal-to-noise ratio (SNR), enabling it to isolate the rPPG signal from a succession of facial video frames. After that, the HR estimator used this signal to generate a prediction of the HR value, with the training procedure aiming to minimize the mean absolute error between the actual and predicted HR values. According to the authors, their proposed method was more effective in dealing with video compression artifacts, which poses a significant challenge for most traditional rPPG signal extraction methods. The authors evaluated their approach on three publicly available datasets, as well as introduced a novel and more demanding dataset (ECG-Fitness) that involved diverse types of motion and lighting conditions. Then another popular 2D CNN DeepPhys model [28] was introduced and showed that DL approaches surpass the previous traditional signal processing approaches. The model,

which integrated a motion and appearance model, was constructed based on the VGG architecture. The motion model received an input motion representation that was obtained by computing the normalized difference between consecutive frames and was built on the DFM, which is used to model color changes and movements. The motion model was directed by the appearance model to understand motion representation by utilizing an attention network. This network generated soft-attention masks from the initial video frames, highlighting areas of the skin that had stronger signals by assigning them higher weights. By using the attention network, it was possible to visualize how the physiological signals were distributed spatially and temporally. According to the author, the utilization of the motion representation and attention network allowed for better capture of physiological signals under different lighting conditions, resulting in increased robustness to changes in illumination and subject movement.

Afterward, Liu et al. [29] also suggested an effective neural structure named MTTS-CAN, which operates on the device and is an enhanced variant of DeepPhys. It enhanced the model's ability to capture temporal information by introducing a temporal shift module (TSM) [30]. This module facilitated information exchange between neighboring frames and eliminated the need for computationally expensive 3D convolution operations by moving tensor segments along the temporal axis. Additionally, instead of using the original video frame, the appearance model was provided with an input frame that was created by taking the average of neighboring multiple frames. Moreover, MTTS-CAN simultaneously predicted both HR and RR by utilizing a multiple-task approach. The network solely relied on 2D CNN, enabling it to achieve on-device inference in just 6 milliseconds per frame, which suggests its applicability for real-time applications. Despite their success, both DeepPhys and MTTS-CAN require several preprocessing steps, such as frame calculation and image normalization as demonstrated in Fig. 5, to accurately extract physiological signals. To address this issue and create a simple and fast on-device camera-based vitals measurement system, Liu et al. [31] proposed the EfficientPhys model which is a one-branch network that includes a custom normalization layer, self-attention module, tensor-shift module, and 2D convolution operation. These components allow for capturing precise and effective spatial and temporal relationships while simplifying the deployment process by inputting raw video frames and outputting a raw PPG signal. By eliminating the need for preprocessing steps, the EfficientPhys model has the capability to measure HR in real-time utilizing on-device cameras in a simple, fast, and precise manner, making it a promising strategy for healthcare, fitness, and wellness monitoring applications.

3D Convolutional Neural Network rPPG: Research authors have suggested using 3D CNN frameworks, referred to as spatiotemporal networks (STNs), to incorporate the temporal information contained in videos, in addition to the spatial information captured by 2D CNNs. STNs have the ability to provide a more comprehensive representation of physiological signals in the video stream. PhysNet [32], one of the first rPPG 3D CNN proposed by Zitong Yu et al. aims to locate the peak of every individual heartbeat and accurately estimate both HR and heart rate variability (HRV). The model processes the initial RGB video frames and generates the rPPG signal as its final output without any intermediate steps. To improve signal trend similarity and reduce peak location inaccuracies, the model uses negative Pearson correlation

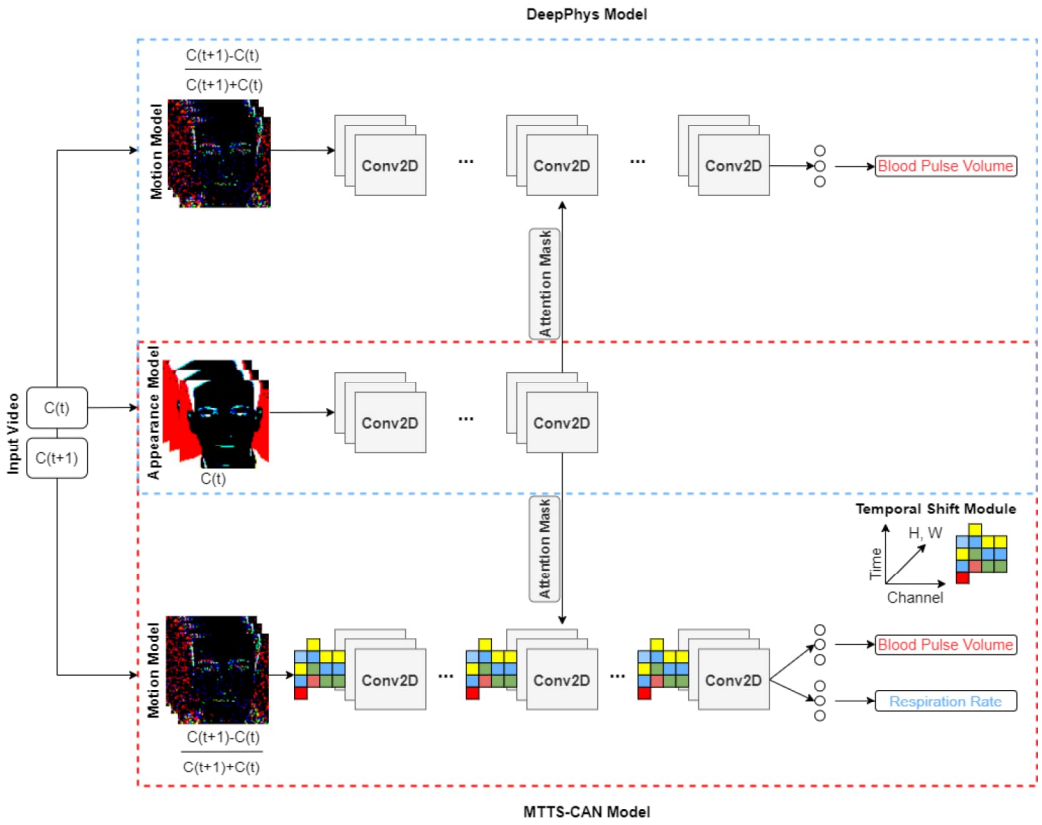


Figure 5. A visual representation of DeepPhys and MTTS-CAN model architectures.

as its loss function. PhysNet's ability to precisely estimate both HR and HRV makes it suitable for more complex applications such as emotion recognition. Later, Zhang and colleagues [33] proposed a speedy and efficient method for estimating HR that requires only 15 seconds of face video. Their approach involves using a multi-hierarchical convolutional network that extracts low-level facial features from RGB face videos using a three-layer 3D CNN. A spatiotemporal stack convolution module is then employed to generate a high-level feature map from the low-level feature maps. Additionally, a skin map is generated from the low-level feature maps to highlight skin areas with strong information while a channel-wise feature map is extracted from the high-level feature map. In the final step, the model generates a weight mask constructed by combining the channel-wise feature map and the skin map, which is used to multiply with the high-level feature map on a channel-by-channel basis, resulting in the extraction of the rPPG signal.

ETA-rPPGNet [34] was introduced with the aim to solve the issue of redundant video information in rPPG signal extraction. To achieve this, the network utilizes a time-domain segment sub-network that models the video's long-term temporal structure. This sub-network consists of several subspace networks, each of which takes a split video segment as input to obtain facial features. An attention mechanism is utilized to acquire essential spatial information. Then the model aggregates the temporal context using an aggregation function to eliminate duplicate video information. This process generates a feature map in each of the separate subspace networks which are integrated and then inputted into the network's backbone to extract the rPPG signal.

The network also includes an attention module to eliminate various noises such as head movements and illumination variations. Finally, a 1D convolution operation is used to effectively describe the local time-domain correlation of the obtained signal. Since the previously mentioned models are complex and do not fully explore the long-range spatiotemporal relationship required for reliable rPPG measurement, Yu et al. proposed a simpler deep-learning neural network called PhysFormer [35]. The model comprises a simple initial section, a component that tokenizes the input data, multiple transformers that work with temporal differences, and a final module that predicts the rPPG. These transformers consist of Temporal Difference Multi-head Self-attention (TD-MHSA) and Spatio-temporal Feed-forward (ST-FF) modules, that improve the global-local spatiotemporal information. The rPPG feature enhancement is improved in PhysFormer through cascaded temporal difference transformer blocks that utilize global spatiotemporal attention. This attention mechanism is based on the small temporal differences in the color of skin. The model addresses the problem of overfitting caused by interference and weak temporal supervision signals by incorporating detailed frequency domain supervision which allows the model to learn intrinsic rPPG-aware features.

2D Convolutional Neural Network + Recurrent Neural Network: Another type of spatiotemporal network combines 2D CNN and RNN which use to capture spatial information and temporal context, respectively. In the same study mentioned in [32], the authors proposed a different version of PhysNet that combines a 2D CNN with different RNNs, such as LSTM, BiLSTM, and ConvLSTM. The objective was to assess the effectiveness of 3D CNN-based PhysNet, RNN-based PhysNet, and various RNN architectures using the same input. Initially, the input was processed by a 2D CNN to capture the spatial characteristics of the RGB video frames. Subsequently, the RNN was utilized to propagate these pieces of information in the temporal domain. The study found that PhysNet based on 3D CNN outperformed PhysNet based on RNN, with the BiLSTM variation performing the poorest suggesting that the backward motion of spatial information is unessential.

In another study, Hu et al. [36] introduced a method for extracting rPPG signals that combine a 2D CNN with a ConvLSTM network and an attention mechanism. The 2D CNN component involved a single trunk and mask branch which was designed similarly to DeepPhys. The trunk branch extracted spatial characteristics from a series of facial images, while the mask branch produced attention masks and transmitted them to the trunk branch to direct feature extraction. Subsequently, the retrieved spatial features were inputted into a ConvLSTM network to leverage the temporal correlation of video frames to obtain rPPG signal.

2.2.2. Hybrid Deep Learning Methods

End-to-end deep learning methods have been widely used due to their ability to automatically learn features from video data. However, these methods face several challenges, such as handling motion artifacts, dealing with low-quality videos, and estimating the HR signal accurately from long video sequences. To address these challenges, researchers have proposed hybrid DL rPPG methods that combine multiple DL models to extract both spatial and temporal features from video frames. These

hybrid methods have shown promising results in estimating HR signals accurately and robustly. The use of hybrid DL techniques for remote HR measurement is described in this section. In this case, DL is only utilized during specific phases of the measurement process. We also mention whether these techniques are applied for video processing, signal extraction, or HR estimation as shown in Fig. 4.

DL-based method for Video Processing: In the majority of current remote HR measurement pipelines usually takes an unprocessed video recorded by a digital camera as input. This implies that it is important to perform face detection or skin segmentation to discard unnecessary information and select ROI by identifying specific skin regions with stronger signals. DL-based methods used for video processing to improve the effectiveness of signal extraction is explained in this section. Tang el at. [37] created a 2D CNN to detect skin in a video database, using manually segmented skin and non-skin regions as positive and negative samples. They evaluated the skin regions using conventional rPPG algorithms and found that their method could capture rPPG signals even with low-cost cameras and single-channel input. Specifically, they identified the RGB channel with the lowest noise in different settings. They also suggested that this method could enhance traditional rPPG methods. However, their approach extracted RPG signals from all facial skin areas, including potentially unnecessary noise was only evaluated on a proprietary yellow skin tone dataset. Paracchini et al. utilized a single-photon avalanche diode camera to record videos in low-light settings in their research [38]. The camera recorded low-resolution grayscale frames which were then fed into a 2D CNN encoder-decoder model. The output of this model was an image with one channel that contains values ranging from 0 to 1. These values indicate the probability that each pixel in the image is categorized as skin. During the training process, transfer learning was used to address the limited availability of data for detecting skin issues. Specifically, the model was initially trained on an extensive set of facial photos with no labels for coloration purposes. Afterward, the model underwent additional training on a skin mask detection dataset, which was generated by applying a threshold to the output and was used for signal extraction.

Video compression also is a necessary step for remote services in order to enable efficient storage and transmission over the Internet. Therefore, it is important to create rPPG techniques that can reliably operate on videos with significant compression. Despite the fact that commercial cameras usually compress videos using different codecs and bitrates most existing rPPG methods do not account for the impact of video compression. To overcome this issue, Yu et al. [39] have introduced a two-stage end-to-end approach that employs hidden rPPG information enhancement and attention networks to counter the loss of rPPG signals resulting from highly compressed facial videos. The first stage of the proposed approach uses a video-to-video generator named Spatio-Temporal Video Enhancement Networks (STVEN) to increase the caliber of compressed videos. The author uses a technique called fine-grained learning, where they assume that different compression bitrates result in different artifact distributions. To tackle this issue, compressed videos are sorted into groups based on their compression bitrate. The model then learns to match the distributions of the videos in each group, resulting in enhanced video sequences. The STVEN model used for this purpose is built on a spatial-temporal convolutional neural network (ST-CNN) that comprises six spatiotemporal blocks sandwiched between

two downsampling layers and two upsampling layers on either end. To increase the model's generalizability, the framework can also use a particular compressing bitrate to compress the original video by utilizing two loss functions: the translation reconstruction loss, which employs mean squared error to handle the loss of video details, and the compression reconstruction loss, that employs L1 loss.

DL-based method for Signal Extraction: The key objective in remote HR measurement is signal extraction, which is the main focus of research in this field. The primary aim of signal extraction is to capture the rPPG wave from videos to estimate HR accurately and to enhance the accuracy of estimation by increasing the quality of the retrieved rPPG signal. To extract high-quality rPPG signals, researchers have proposed various DL methods whose categorization and description can vary depending on the kind of neural network employed. Brian et al. proposed using a Long Short-Term Memory (LSTM) network for signal filtering to improve the quality of the extracted rPPG signal. Since conventional methods for rPPG signal extraction may produce noisy signals, filtering out the noise from the rPPG signal can result in a more precise HR measurement. Initially, the LSTM network underwent training on an extensive artificial dataset and then fine-tuned on real data to enhance its generalization ability. This approach effectively addresses the challenge of limited data availability. In the Deep-HR method [40], a 2D CNN was utilized to capture the color information of the pixels in the ROI. To eliminate any noise in the retrieved data, a GAN-style module was employed. The module consisted of a discriminator that accessed high-quality rPPG signals and provided guidance to a generator for reconstructing a clean rPPG signal, free from noise that can be employed in other rPPG studies to enhance their performance. The Siamese-rPPG [41] approach utilizes a Siamese 3D CNN framework. The underlying concept of this method is that different facial regions may exhibit various noise levels and visual appearances, but they should still contain similar rPPG characteristics. To address this, the ROI was considered to be the forehead and cheek areas as they contain a greater abundance of rPPG information. After selecting the pixels within the aforementioned ROIs, they were subsequently inputted into separate 3D CNNs with identical architectures, one for the forehead and one for the cheek. A weight-sharing mechanism was also utilized to enable the framework to extract signals from the other region in case one of the regions is contaminated with noise, thereby enhancing the overall stability. Subsequently, the outcomes produced by the two separate branches were combined through the use of an additional function. This was accompanied by the implementation of two 1D convolutional processes and an average pooling layer, resulting in the production of the anticipated rPPG signal.

Meta-rPPG [42] employs a transductive meta-learner to adjust weights using unlabeled data during deployment, enabling quick adaptation to various sample distributions. The framework first utilizes a convolutional encoder similar to ResNet to extract latent features from a series of facial images. These features are then fed into a BiLSTM network for the purpose of modeling temporal context. A multilayer perceptron (MLP) was used to estimate the rPPG signals as an additional step. A synthetic gradient generator based on a shallow Hourglass network is also proposed for transductive learning. This generator is further employed in a few-shot learning structure to produce slopes for non-labeled data. PRNet [43] is a spatiotemporal framework for estimating HR from stationary videos, which consists of a 3D CNN extractor for spatial features and local temporal features extraction from ROI, followed

by an LSTM extractor for global temporal features. HR estimation is achieved by applying a fully connected layer to the obtained feature map. In contrast to other remote HR measurement techniques that necessitate 6-30 seconds of video, Huang et al. contended that their framework could anticipate HR using just 60 frames (2 seconds) of video. Song et al. created the PulseGAN model [44] which is a GAN-based framework that generates high-quality, realistic rPPG signals for accurate HR estimation. The model takes a raw rPPG signal extracted from the CHROM method on the ROI as an input and generates the finalized rPPG signal that closely resembles the label signal. Its discriminator maps the raw rPPG signal to a final rPPG signal using the ground truth rPPG signal as guidance. By integrating this framework with other conventional rPPG methods, the quality of the retrieved signal can be improved and lead to more precise HR measurements.

DL-based method for HR Estimation: In the past, HR estimation was typically accomplished using traditional methods such as bandpass filtering and peak detection. However, DL methods can also be used to approach HR estimation as a regression problem and they suggested various features of the HR signal. One approach, described in Reference [45], involves extracting the rPPG signal using traditional methods and applying short-time Fourier transform and bandpass filtering to generate a frequency domain representation. Then it can be coupled with the time domain wave to create a spectrum image that can act as a means of characterizing the signal. This spectrum image is then put into a ResNet18 model that has been trained on the ImageNet dataset to act as an HR estimator. By leveraging the HR estimator's capacity to acquire information from spectrum photos and then translate them to HR without the use of conventional methods, the proposed approach can achieve accurate HR estimation. A different approach to representing the HR signal is through spatiotemporal maps, which require the selection of an ROI. This method involves utilizing color information from the ROI pixels and concatenating them in temporal sequences to form a matrix representation. This spatiotemporal map is then input into a neural network for direct HR estimation. This approach allows for highlighting the HR signal while suppressing irrelevant information. Transfer learning with pre-training using the ImageNet dataset can be used to address the issue of insufficient data. Some examples of this approach include using a combination of 2D CNN and gated recurrent unit (GRU) for HR estimation [46], and employing the Neural Architecture Search (NAS) method to find a lightweight and optimal CNN for HR estimation from spatiotemporal maps [47], along with an attention module to mitigate the effect of different types of noise [48].

3. IMPLEMENTATION

This section presents a comprehensive approach to evaluating video data and signals using traditional and DL methods. Our evaluation process begins by preparing the video data and signals for analysis. This involves utilizing both traditional and DL techniques, which allow us to gain a deeper understanding of the underlying structure and features of the data. Next, we introduce a set of transformed methods that can be used to create transformation videos commonly encountered in practice. By utilizing these methods, we can simulate a variety of real-world scenarios that can have a negative impact on the quality of video data. These scenarios include faces that are occluded by sunglasses or facemasks, as well as common types of image degradation such as noise artifacts and motion blur. After creating these deteriorated videos, we evaluate a selection of the best methods for analyzing them. This involves applying various traditional and DL techniques to the videos and assessing their performance in terms of selected metrics and overall effectiveness. Additionally, we experiment with several restoration methods that can be used to improve the performance of the selected rPPG method in these types of videos. Finally, we fine-tune different combinations of restored datasets using various fine-tuning methods to determine their effectiveness in improving the performance of our rPPG model.

3.1. Data Resampling

This thesis research utilizes three datasets: UBFC-rPPG [49], UCLA-rPPG [50], and UBFC-Phys [51]. Due to the differences in reading signal data and sample rates, UBFC-rPPG was initially divided into two different parts as shown in Table 1.

Table 1. The original parameter of the datasets

	⌊FPS⌋ (frames/s)	Signal Sample Rate (samples/s)	⌊Duration⌋ (second)
UBFC1-rPPG	28	62	42-117
UBFC2-rPPG	29	29-30	46-68
UCLA-rPPG	30	30	59
UBFC-Phys	35	64	180

To ensure uniformity in video duration, computational efficiency, and to evaluate the effectiveness of techniques used for rPPG on low fps video data, the data were split, extracted, and downsampled. A 16-second segment was selected from the middle of each video to exclude any unusual actions that usually occur at the beginning or end. Furthermore, to reduce the amount of data, the selected segment was downsampled to 20 frames per second. To achieve precise resampling, I split the indexes of the 32-second window into 32 smaller chunks, representing frames indexes in seconds, and then split each of these chunks into smaller chunks corresponding to the desired fps rate of 20. The middle frame from each of these smaller chunks was selected to effectively resample the frames from the original fps rate to the desired fps rate. To prepare the video frames for input to pre-trained models, each video was scandalized. As the rPPG signal is periodic, a Fourier-based method was used

for resampling to preserve the information content of the signal and minimize the introduction of artifacts. Resampling can introduce bias or skewness to the signal, so after resampling, standardizing was applied to remove any remaining noise or outliers. Additionally, to ensure consistent analysis across all data, the rPPG signals were downsampled to match the video frames, resulting in a total of 320 samples per subject ($n\_samples\_per\_subject$) as shown in Table 2.

Table 2. The parameter for downsampling dataset segments

	FPS	Signal Sample Rate	Duration
UBFC-rPPG UCLA-rPPG UBFC-Phys	20 frames/s	20 samples/s	16s

3.2. Face Region

In order to detect faces in each frame of the video with high accuracy, a pre-trained face detection model called Multi-Task Cascaded Convolutional Networks (MTCNN) [8] was utilized. To further enhance the precision of face detection, a custom function was developed to adjust the bounding boxes generated by the MTCNN algorithm.

Algorithm 1. Adjust bounding boxes

Input: Bounding boxes $B \in \mathbb{R}^{N\times 4}$
Output: Adjusted bounding box $bbox \in \mathbb{R}^4$
if B *is None* **then**
  $B \leftarrow [0, 0, 0, 0]$
else
  $B' \leftarrow$ flatten(B)
  $B'_1 \leftarrow \max(B'_1, 0), B'_2 \leftarrow \max(B'_2, 0)$
  $B'_3 \leftarrow \max(B'_3, 0), B'_4 \leftarrow \max(B'_4, 0)$
  $L \leftarrow \max(B'_2 - B'_0, B'_3 - B'_1)$
  $C_x \leftarrow \left\lfloor \frac{B'_1+B'_3}{2} \right\rfloor$
  $C_y \leftarrow \left\lfloor \frac{B'_0+B'_2}{2} \right\rfloor$
  $B \leftarrow [C_x - \lfloor \frac{L}{2} \rfloor, C_x + \lfloor \frac{L}{2} \rfloor, C_y - \lfloor \frac{L}{2} \rfloor, C_y + \lfloor \frac{L}{2} \rfloor]$
end

This function is demonstrated in Algorithm 1, which takes the MTCNN bounding boxes as input and returns adjusted bounding boxes that are centered on the face and have equal width and height. The function first checks if the input bounding boxes are valid and if they are not, it returns an empty bounding box. If the input bounding boxes are valid, the function flattens them and calculates the maximum length of the box (L), which is used to set the width and height of the adjusted bounding box. The center point of the input bounding box (C_x, C_y) is then calculated, and the adjusted bounding box is centered on this point. The function returns the adjusted bounding

box as an array of four values representing the x_1, x_2, y_1, y_2 coordinates of the box. By creating square bounding boxes that are centered on the face, the function ensured that the detected faces are more accurately cropped from the input images, which improved the performance of subsequent face recognition and analysis tasks.

To minimize computational costs and processing time, face detection is performed only three times per video, at positions $(0, \lfloor \frac{n\_samples\_per\_subject}{3} \rfloor, \lfloor \frac{n\_samples\_per\_subject*2}{3} \rfloor$, specifically at frame indices 0, 106, and 213. The face region is cropped from the video frame using the face box at these positions, corresponding to the intervals $[0, 106)$, $[106, 213)$, and $[213, 219)$, respectively. To ensure consistency in face region size, the cropped face region is resized to dimensions (72,72), and the whole process is illustrated in Fig. 6.

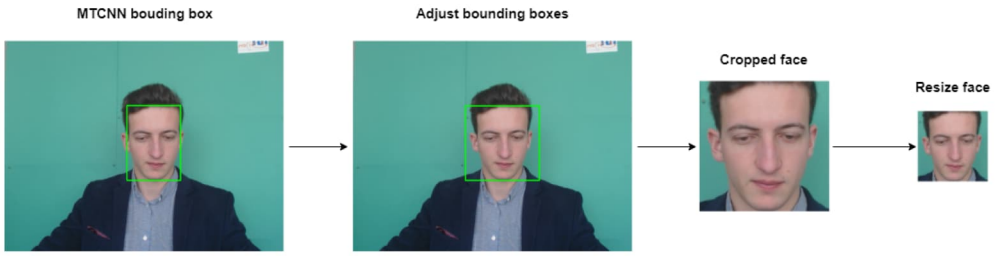


Figure 6. Example of using MTCNN and adjusting the bounding box to crop then resize the facial image.

3.3. Image Transformation

This section explores the image transformation process in-depth, with a specific emphasis on dealing with challenges such as motion blur, noise artifacts, occluded eyes, and facemasks, as demonstrated in Fig. 7.

Motion Blur: To simulate motion blur in the image, a 2D Gaussian filter is used, as given by Equation 8:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{8}$$

Where, $G(x, y)$ is the Gaussian function at position (x, y) in the image, σ is the standard deviation of the Gaussian distribution (to control the degree of blurring), e is

(a) Original image (b) Motion blur (c) Gaussian noise (d) Eyemask (e) Facemask

Figure 7. Four image transformation techniques.

the mathematical constant e (approximately 2.71828), π is the mathematical constant pi (approximately 3.14159), and x, y are the pixel coordinates in the image. The filter is convolved with the image using a kernel of size $k \times k$, where k is an odd integer. In this work, I use a kernel size of k = 15. This filter blurs the image in the direction of motion, simulating the effect of motion blur in real-world scenarios.

Noise Artifacts: To add noise to a colored image, Gaussian noise is used, which is a common technique for simulating real-world scenarios where images may have different types of noise. Gaussian noise is a type of random noise that is often present in electronic devices such as cameras and sensors. It is a statistical noise that has a probability distribution defined by the Gaussian function (Equation 9). where z represents the grey level, μ the mean grey value and σ its standard deviation.

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{9}$$

To add color Gaussian noise to an image, the equation for adding monochrome Gaussian noise is extended to include the three color channels: red (R), green (G), and blue (B). The equations for the red, green, and blue channels are:

$$\begin{aligned} I_R(x,y) &= I_R(x,y) + N_R(x,y) \\ I_G(x,y) &= I_G(x,y) + N_G(x,y) \\ I_B(x,y) &= I_B(x,y) + N_B(x,y) \end{aligned} \tag{10}$$

where $I_R(x,y)$, $I_G(x,y)$, and $I_B(x,y)$ are the values of the red, green, and blue channels, respectively, at pixel (x,y) in the original image. $N_R(x,y)$, $N_G(x,y)$, and $N_B(x,y)$ are the amounts of Gaussian noise to be added to the red, green, and blue channels at that pixel. To generate color Gaussian noise, a mean value ($\mu = 0$), and standard deviation ($\sigma = \sqrt{0{,}004}$) are first defined for each color channel, resulting in three sets of values. For each pixel in the image, a random number is generated for each color channel using a Gaussian distribution with the corresponding mean and standard deviation, and these values are added to the original pixel values for each color channel to produce the final noisy image.

Occluded Eyes: To create a mask in the shape of sunglasses on a human face in an image. The function is created to compute the major and minor axes of the ellipse based on the distance between the eyes, the bridge of the nose, and the middle point of the eyebrows as shown in Fig. 8. These landmarks are detected using dlib's advanced face recognition algorithm built with DL techniques [52, 53]. The resulting mask is then combined with the input image using a bitwise $\vee$ operation to create the final image with the mask in the shape of sunglasses. The function is designed to be robust to changes in the position of the face. It uses facial landmark detection to locate the key points needed to draw the mask, which ensures that even if there is a change in the position of the face, the mask is drawn correctly as long as the face can still be detected. The algorithm returns a new set of images with the sunglasses mask applied to the detected face, enabling easy application of the mask to a large number of images.

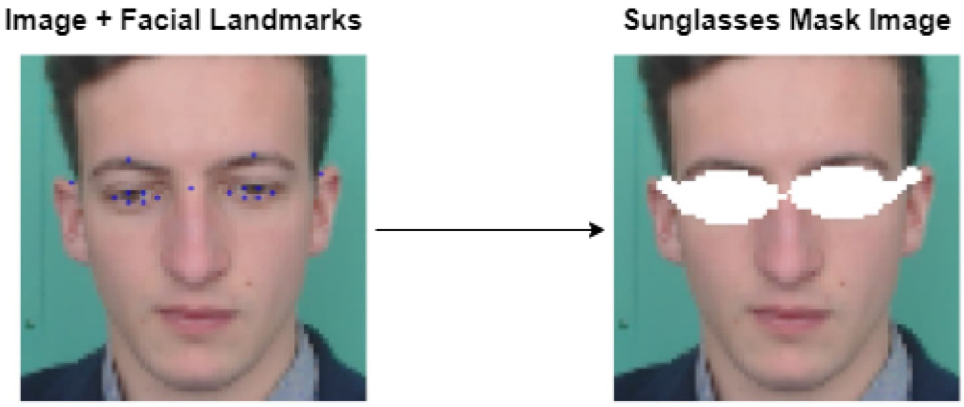


Figure 8. Example of how to create a sunglasses mask for the facial image.

Facemask: To create a facemask for a facial image, the algorithm starts by identifying the facial landmarks such as the chin and nose tip using the same landmarks detected algorithm as above. Then, a polygon contour is created using a subset of the chin landmarks and the second point of the nose bridge. The interior of this contour is then shaded in white to create the mask. Finally, the mask is combined with the input image using a bitwise $\vee$ operation, resulting in the final image with the facemask in place. This approach is simpler than creating the sunglasses mask demonstrated in Fig. 8 but follows a similar overall process.

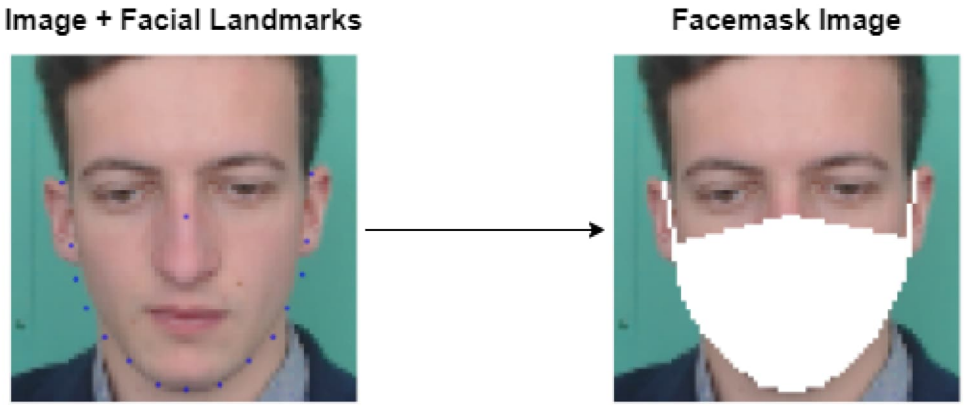


Figure 9. Example of how to create a facemask for the facial image.

3.4. RPPG Methods

To compare and select the most suitable methods for evaluating the performance of image transformations, a variety of traditional rPPG methods and pre-trained DL models are assessed on original data with low frame rates. This evaluation includes a total of nine rPPG methods, consisting of six traditional and three DL methods as listed in the section below.

3.4.1. Traditional RPPG Methods

A detailed explanation of the six methods that have been selected for assessing original data with low frame rates is provided in this section. The six traditional methods are GREEN, LGI, PBV, ICA, CHROM, and POS [54]. These methods are based on different principles, such as color space transformation, temporal normalization, independent component analysis, and projection onto orthogonal planes. By comparing the performance of these approaches on the original data with low frame rates, it is possible to identify the most effective method for detecting the pulse rate on transformation video datasets. Additionally, these methods are relatively simple to implement and do not require extensive training, making them suitable for real-time applications.

GREEN [20]: The GREEN rPPG method has been argued that one of the simplest approaches to estimating HR via rPPG shown in Equation 11 where $x(t)$ is the RGB temporal traces.

$$S(t) = x_g(t) \tag{11}$$

It analyzes changes in the green color channel of the video frames captured by the camera, which are caused by variations in the volume of blood on the human face during the cardiac cycle. The method involves computing the temporal average of the green channel pixel values in the facial image and then extracting the pulse signal by using a bandpass filter to isolate the frequency range of the heartbeat. The method assumes that the pulse signal is present in the green channel and that the face region is motionless during the measurement.

LGI [55]: The objective of the Local Group Invariance (LGI) method is to create a more reliable feature space for rPPG from an original signal $x(t)$, that is less affected by interfering factors like human movements and changes in lighting. To accomplish this, a matrix X is created by vectorizing the pixels of skin in an RGB channel video frame as shown in Equation 12.

$$C = \frac{X^T \cdot X}{N} \tag{12}$$

X has dimensions of N×3, with N is the pixels number. C is then defined by means of eigenvalues $\Lambda = diag(\lambda_1, \lambda_2, \lambda_3)$ and eigenvectors U which is used as a novel coordinate system for skin pixels. A projection operator O is derived from C with I as the identity matrix using Equation 13. The output signal is generated in the final stage by translating the input signal $x(t)$ onto the new space defined by matrix O demonstrated in Equation 14.

$$O = I - UU^T \tag{13}$$

$$S(t) = Ox(t) \tag{14}$$

PBV [22]: According to the authors, the changes in optical absorption due to blood volume variations in the skin occur along a specific vector in the normalized color channel space. This vector is referred to as the Pulse Blood Volume (PBV) vector. The authors explain that the PBV vector is computed using the following formula:

$$P_{bv}(t) = \frac{[\sigma(x_r), \sigma(x_g), \sigma(x_b)]}{\sqrt{\sigma^2(x_r), \sigma^2(x_g), \sigma^2(x_b)}} \tag{15}$$

The pre-processed signal within the selected window $x(t)$ is represented in matrix form as $X = [x_r, x_g, x_b]$, with the standard deviation operator $\sigma(.)$ being applied to it. The output signal is obtained by computing the projection of the pre-processed signal $x(t)$ onto an orthogonal matrix M where P_{bv} is the Pulse Blood Volume vector calculated previously, and k is a normalization factor using the equation below:

$$\begin{gathered} S(t) = Mx(t) \\ \text{where } M = kP_{bv}(XX^t)^{-1} \end{gathered} \tag{16}$$

ICA [13]: The process involves capturing a video of the facial area using a webcam, where the RGB color sensors detect a combination of the reflected plethysmographic signal and other variations in light caused by factors such as movement and changes in ambient lighting conditions. Due to variations in hemoglobin absorptivity throughout the visible and near-infrared spectral range, the signals captured by each color sensor are a combination of the original signals with slight variations in weighting. At time point t, the amplitudes of the recorded signals are represented as x_r, x_g, and x_b which are the averages of all pixels in the facial region. The number of observable sources in conventional ICA cannot be greater than the number of observations. Therefore, it is assumed that there are three underlying source signals represented by S_r, S_g, and S_b. The ICA model assumes that the observed signals are linear mixtures of the sources which can be represented compactly by the mixing Equation below:

$$x(t) = AS(t) \tag{17}$$

Where the column vectors $x(t) = [x_r, x_g, x_b]^T$, and $S(t) = [S_r, S_g, S_b]^T$, and the square 3x3 matrix A contains the mixture coefficients a_{ij}. The objective of ICA is to find a separating or demixing matrix W that approximates the inverse of the original mixing matrix A. The output of the demixing process is an estimate of the vector $S(t)$ that contains the underlying source signals as follows:

$$S(t) = Wx(t) \tag{18}$$

To extract the independent sources, the demixing matrix W must maximize the non-Gaussianity of each source, as the central limit theorem indicates that the sum of independent random variables is more Gaussian than the original variables. In order to achieve this, iterative methods are commonly employed to optimize a given cost function that measures non-Gaussianity, such as kurtosis, negentropy, or mutual information.

CHROM [21]: This approach is suggested to address the limitations of other rPPG methods that suffer from unpredictable normalization errors caused by specular reflections on the skin surface which are not present in contact PPG. The DFM depicted in Fig. 3 provides an explanation for the two underlying components that comprise the reflected light from the skin. The observed color of the light reflected from the skin is a result of the combination of these two components. The proportions of these two components are determined by the angles between the camera lens, human skin, and the source of light, thus resulting in change through time as the subject moves, which creates a weakness in rPPG algorithms that do not account for the additive specular component. To address this issue, the CHROM method utilizes chrominance signals to remove the specular reflection $\mathbf{v_s}(t)$ by exploiting the difference of color. The method starts with projecting normalized RGB values from the RGB traces $x(t)$ into two chrominance vectors, X_C and Y_C, which are orthogonal to each other as follows:

$$\begin{aligned} X_C(t) &= 3x_r(t) - 2x_g(t) \\ Y_C(t) &= 1{,}5x_r(t) + x_g(t) - 1{,}5x_b(t) \end{aligned} \tag{19}$$

Equation 20 is eventually used to compute the rPPG signal outputF where $\alpha = \frac{\sigma(X_C(t))}{\sigma(Y_C(t))}$, and $\sigma(.)$ is the standard deviation.

$$S(t) = X_C(t) - \alpha Y_C(t) \tag{20}$$

POS [11]: The POS method aims to remove specular reflections on the skin surface, similar to the CHROM method, by defining a plane that is orthogonal to the skin tone in the RGB color space that has been temporally normalized. The POS method involves three steps for processing the input signal $x(t)$. After normalized temporal information, the data is projected onto an orthogonal plane to the skin tone as shown in Equation 21:

$$\begin{aligned} X_P(t) &= x_g(t) - b(t) \\ Y_P(t) &= x_g(t) + x_b(t) - 2x_r(t) \end{aligned} \tag{21}$$

Finally, a tuning step is conducted to find the precise direction of projection within the limited area established in the previous stage, where α is the same as CHROM using the below formula:

$$S(t) = X_P(t) + \alpha Y_P(t) \tag{22}$$

In contrast to CHROM, the POS method seeks two projection axes that yield signals in phase, whereas, in CHROM, the two projected signals are in antiphase. Additionally, in order to enhance the SNR of the resulting signal, the video input sequence is partitioned into smaller time intervals, and the pulse rate is calculated from these shorter video segments. Finally, the partial segments are overlap-added to derive the final signal.

3.4.2. Deep Learning RPPG Methods

This section provides a comprehensive understanding of the three deep-learning methods that have been selected to assess original data with low frame rates. The criteria for selecting these methods are based on their simplicity, ease of use, and their ability to operate on a single input. Moreover, the methods are pre-trained, which implies that they are already trained on large datasets and can be fine-tuned on a specific dataset to increase their performance. The application of DL methods is motivated by their ability to learn and extract features automatically from the data, which can then be used to make accurate predictions. These methods are also advantageous because they can handle large amounts of data and can generalize well to unseen data. The three DL methods that are used are EfficientPhys, ContrastPhys, and PhysFormer. Each of these methods has its strengths and weaknesses, and their performance is evaluated on the original data to determine the most effective method for detecting the pulse rate on transformation video datasets. Overall, the use of DL methods provides a promising approach for accurately detecting the pulse rate from original data with low frame rates.

EfficientPhys [31]: The model aims to simplify the process of removing pre-processing modules by providing an all-in-one solution. To achieve this, the author suggests using a customized normalizing module capable of modeling movement among each pair of consecutive original video frames and normalizing them to minimize lighting and motion noise. This module consists of a different layer and a batch norm layer. The difference layer calculates the first forward difference across the temporal direction among the original frames through the subtraction of each pair of neighboring frames. The motion modeling and normalization act as a high-pass filter to reduce global noise from lighting and movement while preserving small PPG variations. The Equation 23 provides an optical basis for the different frames where $\boldsymbol{D}_k(t)$ refers to the contrast between two successive frames, while $I(t)$ represents the brightness intensity, which undergoes modification due to specular reflection $\boldsymbol{v}_s(t)$, diffuse reflection $\boldsymbol{v}_d(t)$, and quantization noise $\boldsymbol{v}_n(t)$ originating from the optical sensor.

$$\begin{aligned}\boldsymbol{D}_k(t) = (I(t) \cdot (\boldsymbol{v}_s(t) + \boldsymbol{v}_d(t)) + \boldsymbol{v}_n(t)) - (I(t-1) \\ \cdot (\boldsymbol{v}_s(t-1) + \boldsymbol{v}_d(t-1)) + \boldsymbol{v}_n(t-1))\end{aligned} \tag{23}$$

When using different frames for feature extraction in video analysis, variations in scale can complicate the network's ability to learn significant characteristics, particularly if the important signal is obscured by minor pixel fluctuations and unwanted noise. To resolve this problem, a batch normalization layer is incorporated after the difference layer. While training, it normalizes the frames with a consistent scale among a batch and provides two learnable parameters for scaling and shifting, which allows the layer to learn the best parameters for increasing pixel variations while minimizing noise as Equation 24 shows. Adding a batch normalization layer helps the network to acquire knowledge of the normalization function to significantly amplify the minor alterations in skin pixels. The output of the batch norm layer contains more

information and is better for skin segmentation post-training compared to hand-crafted normalized frames.

$$\boldsymbol{N}_k(t) = \frac{(\beta_t * \boldsymbol{D}_k(t) + \gamma_t) - \mu_{\boldsymbol{D}_k}}{\sigma_{\boldsymbol{D}_k}} \tag{24}$$

The self-attention-shifted network (SASN) is proposed to effectively extract spatial-temporal features in optical cardiac monitoring. The SASN architecture is constructed on the foundation of the previous state-of-the-art approach, the temporal-shift convolutional attention network (TS-CAN) [29], which has two branches to process the difference and appearance frames separately. However, the attention masks in TS-CAN are able to study with a one-branch end-to-end network. The SASN starts with a normalization module and then performs tensor-shifted convolutional processes as illustrated in Fig. 10.

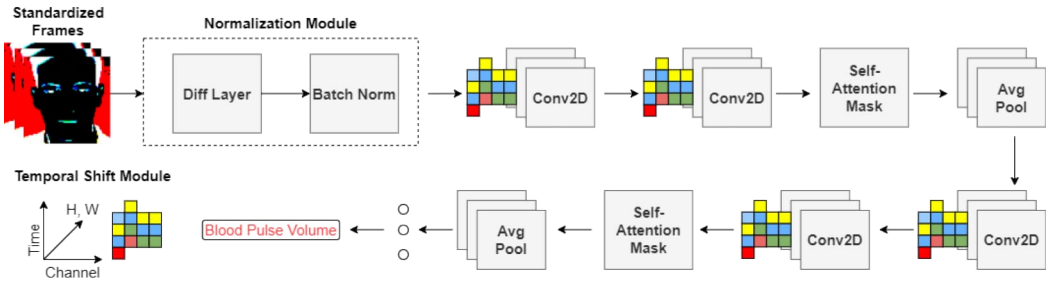


Figure 10. A visual representation of EfficientPhys model architecture Adapted and modified from [31].

Self-attention layers are added after certain convolutional layers to mitigate the detrimental impact caused by temporal shifting, motion, and lighting noise. The self-attention network uses a sigmoid activation function and element-wise multiplication to obtain the final attention mask. The way the self-attention mechanism works can be explained using Equation 25, where it uses the notation $ts(.)$ to represent temporal shift operation, ω_a^c to represent 2D convolutional kernel with temporal shift module, σ as the sigmoid activation function, and ω_a^t to represent the 1×1 convolutional kernel for self-attention mechanism.

$$\left(\omega_c^t ts\left(\boldsymbol{N}_k(t)\right) + b_c^t\right) \odot \frac{H_t W_t \cdot \sigma\left(\omega_a^t \mathbb{X}_\alpha^t + b_a^t\right)}{2\left\|\sigma\left(\omega_a^t \mathbb{X}_\alpha^t + b_a^t\right)\right\|_1} \tag{25}$$

The authors in reference [56] released a pre-trained version of the EfficientPhys model that was trained on the UBFC-rPPG dataset [49] with a frame rate of 30 frames per second. The model is configured with a depth of 10 frames for the TSM, which splits each video and detects the face bounding box 180 times. During training, an AdamW optimizer with a learning rate of $9e$-3 was utilized for a total of 30 epochs, and the batch size was set to 4.

ContrastPhys [57]: The diagram in Fig. 11 provides an overview of ContrastPhys. To obtain the ST-rPPG block representation, the PhysNet model based on 3DCNN is modified by the authors. The input RGB video has dimensions $T \times dim \times dim \times 3$,

where T denotes the frame's number and dim signifies the image's dimension. Adaptive average pooling is used in the last stage to decrease the spatial dimensions and manage the output spatial dimension size. This alteration enables the model to create an ST-rPPG block with dimensions $T \times S \times S$, with S representing the length of the spatial dimension. Each is an assemblage of spatiotemporal rPPG signals and denoted by $P \in \mathbb{R}^{T \times S \times S}$. If a particular spatial location (h, w) is selected from the ST-rPPG block, the associated rPPG signal at that location would be $P(\cdot, h, w)$. This is obtained from the raw video's receptive field of the corresponding spatial location. If the length of spatial dimension S is limited, the receptive field of every spatial location in the ST-rPPG block is broader and encompasses a section of the facial area. This indicates that every position of space within the ST-rPPG block has the ability to encompass rPPG information.

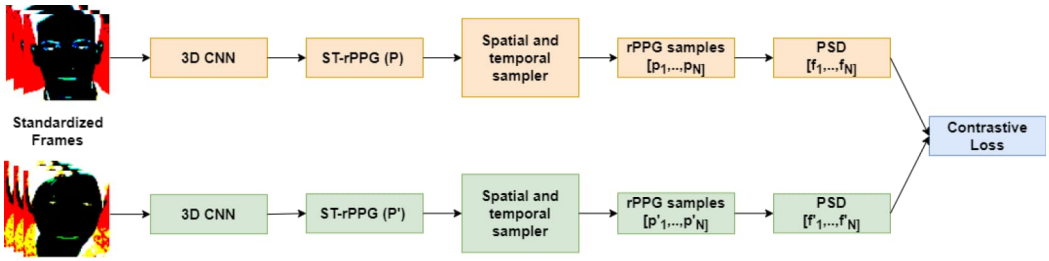


Figure 11. A graphical illustration of ContrastPhys model architecture Adapted and modified from [57].

Spatial information can be extracted by selecting a single spatial position and retrieving the corresponding rPPG signal $P(\cdot, h, w)$. To acquire temporal information, a brief time interval can be sampled from $P(\cdot, h, w)$, generating a spatiotemporal sample denoted by $P(t \to t + \Delta t, h, w)$. With h and w denoting the spatial location, t represents the start time, and Δt signifies the interval of time duration. In the case of an ST-rPPG block, all spatial positions are iterated over and K rPPG clips are sampled for each spatial position with a starting time t chosen at random. Thus, a total of $S \cdot S \cdot K$ rPPG clips can be obtained from the ST-rPPG block. During model training, the aforementioned sampled processes are employed. Once the model has been trained and used for testing, the ST-rPPG can be spatially averaged to obtain the rPPG signal.

The model takes two distinct videos chosen randomly from a dataset as input as depicted in Fig. 11. The first video produces one ST-rPPG block P, along with a series of rPPG samples $[p_1, \ldots, p_N]$ and relating PSDs $[f_1, \ldots, f_N]$. Similarly, the second video produces one ST-rPPG block P', one series of rPPG samples $[p'_1, \ldots, p'_N]$ and relating PSDs $[f'_1, \ldots, f'_N]$. The contrastive loss function aims to bring together PSDs belonging to the same video and separate PSDs from two separate videos. The loss function comprises two terms: the positive and negative loss, and is expressed as $L = L_p + L_n$. These terms are computed based on the following formulas:

$$
\begin{aligned}
L_p &= \frac{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left(\|f_i - f_j\|^2 + \|f'_i - f'_j\|^2 \right)}{(2N(N-1))} \\
L_n &= \frac{-\sum_{i=1}^{N} \sum_{j=1}^{N} \|f_i - f'_j\|^2}{N^2}
\end{aligned}
\tag{26}
$$

The model weight published in this study was trained on the UBFC2-rPPG dataset [49] without changing fps using $K = 4$, where four rPPG samples were selected randomly for each spatial position in the ST-rPPG block. The ST-rPPG block had a spatial resolution of 2 × 2 and the length of time was set to 10s with a time interval Δt of half the block length for each rPPG sample. The training was carried out over 30 epochs with an AdamW optimizer and a learning rate of $1e-5$, with each training iteration including two 10s clips from different videos.

PhysFormer: The architecture of the model is inspired by the study presented in [58], and it comprises a shallow stem $\mathbf{E}_{stem}$, a tube tokenizer $\mathbf{E}_{tube}$, N temporal difference transformer blocks $\mathbf{E}^{i}_{trans}$ $(i, \ldots, N)$, and an rPPG predictor head, as illustrated in Fig. 12.

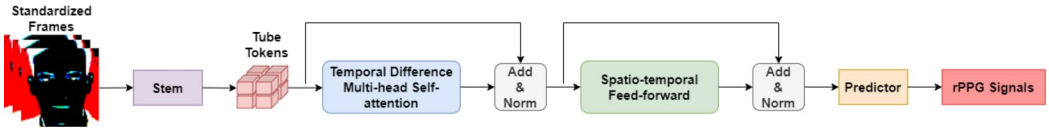


Figure 12. A graphical representation of the Physformer model architecture Adapted and modified from [58].

The shallow stem is designed to extract coarse local spatiotemporal features, enabling faster convergence and clearer global self-attention. Specifically, it comprises three convolutional blocks, each with a different kernel size followed by batch normalization, ReLU activation, and max pooling. The pooling layer reduces the dimension of space by half. If an RGB facial video input $X \in \mathbb{R}^{3\times T\times H\times W}$ is given, the stem generates an output X_{stem}, which is obtained by passing X through the function $\mathbf{E}_{stem}$ and has a dimension of $D \times T \times \frac{H}{8} \times \frac{W}{8}$, where D, T, W, and H denote channel, length, width, and height of sequence, respectively. Subsequently, X_{stem} is divided into spatiotemporal tube tokens X_{tube} by the tube tokenizer $\mathbf{E}_{tube}$. The tube tokens are then passed through N temporal difference transformer blocks to extract the global and local refined rPPG information X_{trans} with similar dimensions as X_{tube}. Lastly, the rPPG predictor head takes the X_{trans} and performs three operations: temporal upsampling, spatial averaging, and projection to a 1D sequence of predicted rPPG values Y.

In the tube tokenization step, the initial coarse feature X_{stem} is split into separate tube tokens using $\mathbf{E}_{tube}(X_{stem})$. This operation combines spatiotemporal neighbor information and helps reduce the computational burden on the subsequent transformers. $\mathbf{E}_{tube}(X_{stem})$ generates the tube token map X_{tube}, which has a length, height, and width corresponding to the desired tube size $T_s \times H_s \times W_s$, which is identical to the step size of the partition in the non-overlapping configuration. The resulting X_{tube} is a tensor of dimensions $D \times T' \times H' \times W'$ which are calculated using the below equation:

$$T' = \lfloor \frac{T}{T_s} \rfloor, \lfloor H' = \frac{H/8}{H_s} \rfloor, \lfloor W' = \frac{W/8}{W_s} \rfloor \tag{27}$$

By projecting the query and key pairs and computing their similarity, the self-attention mechanism models the connection between tokens, which results in an

attention score. Rather than using a point-wise linear projection, a temporal difference convolution (TDC) was employed for projecting the query (Q) and key (K). This approach can extract fine-grained temporal difference information at a local level that describes small color changes. The TDC with a learnable parameter w is expressed as follows:

$$\mathrm{TDC}(x) = \underbrace{\sum_{p_n \in \mathcal{R}} w\left(p_n\right) \cdot x\left(p_0 + p_n\right)}_{\text{vanilla 3D convolution}} + \theta \cdot \underbrace{\left(-x\left(p_0\right) \cdot \sum_{p_n \in \mathcal{R}'} w\left(p_n\right)\right)}_{\text{temporal difference term}} \tag{28}$$

The current spatiotemporal location sampled local (3x3x3) neighborhood and sampled adjacent neighborhood are denoted by p_0, $\mathcal{R}$, and $\mathcal{R}'$, respectively. The query (Q) and key (K) are projected using TDC instead of point-wise linear projection. TDC with a learnable w captures fine-grained local temporal differences information for describing minor color changes. The projected Q and K are then normalized with batch normalization (BN). The value (V) projection employs point-wise linear projection without BN. The Q, K, and V are then flattened into a sequence and separated into h heads. For the i-th head ($i \leq h$), the self-attention (SA) where τ controls the sparsity can be formulated as:

$$SA_i = Softmax(\frac{Q_i K_i^T}{\tau})V_i \tag{29}$$

To obtain the output of TD-MHSA, the self-attention (SA) results from all heads are concatenated and then subjected to a linear projection, represented as $U \in \mathbb{R}^{D \times D}$, The notation TD-MHSA is represented as the concatenation of SA_1, SA_2, ..., SA_h and then projected using the linear operator U. After TD-MHSA, residual connection and layer normalization (LN) are conducted. Rather than using a vanilla feed-forward network consisting of linear transformation layers, a depthwise 3D convolution with BN and nonlinear activation is introduced between the two layers to improve performance at a slight additional computational cost. This ST-FF approach provides a complementary refinement to TD-MHSA for local inconsistencies and noisy features, while also offering richer locality for TD-MHSA to obtain sufficient relative position cues. This approach utilizes label distribution to cover a range of class labels for each instance, allowing facial videos to contribute to both targeted and adjacent HR values, treating the rPPG-based HR measurement as a multi-label classification problem with L classes, where a label distribution $p = \{p_1, p_2, \ldots, p_L\} \in \mathbf{R}^L$ is assigned to each facial video X, with each entry of p being a real value in $[0, 1]$ such that $\sum_{k=1}^{L} p_k = 1$, and a Gaussian distribution centered at the ground truth HR label Y_{HR} with standard deviation σ is used to construct the corresponding label distribution p demonstrated in Equation 30, and the label distribution loss is computed as $\mathcal{L}_{LD} = \mathrm{KL}(\mathbf{p}, \mathrm{Softmax}(\hat{\mathbf{p}}))$, where $\mathrm{KL}(\cdot)$ represents the Kullback-Leibler divergence [59] and $\hat{\mathbf{p}}$ is the power spectral density of the predicted rPPG signals.

$$p_k = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(k-(Y_{HR}-41)^2}{2\sigma^2}} \tag{30}$$

The model also incorporates curriculum learning that follows an easy-to-hard approach to training. For rPPG measurement, the model is supervised with signals from both the temporal and frequency domains, which impose different constraints on the learning process. The model's dynamic loss function, $\mathcal{L}_{overall}$, is a combination of $\mathcal{L}_{time}$ and $\mathcal{L}_{CE}$ as shown in Equation 31, representing the negative Pearson loss and frequency cross-entropy loss, respectively, with hyperparameters α, β_0, and η set to 0.1, 1.0, and 5.0, respectively. Dynamic supervision enables the model to learn better signal trends initially, which facilitates gradual knowledge learning later.

$$\begin{aligned} \mathcal{L}_{\text{overall}} &= \underbrace{\alpha \cdot \mathcal{L}_{\text{time}}}_{\text{temporal}} + \underbrace{\beta \cdot (\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{LD}})}_{\text{frequency}} \\ \beta &= \beta_0 \cdot \left(\eta^{\frac{(\text{Epoch}_{\text{current}} - 1)}{\text{Epoch}_{\text{total}}}} \right) \end{aligned} \tag{31}$$

This thesis uses the PhysFormer was trained on a single fold of 30 fps VIPL-HR dataset [60], a large-scale collection of 2,378 RGB videos featuring 107 subjects under diverse recording scenarios, using the model configuration of N=12, h=4, D=96, and D'=144 along with TD-MHSA settings of θ=0.7 and τ=2.0, and the targeted tube size of 4×4×4. During training, RGB face clips of size 160×128×128 (T×H×W) were utilized with techniques for augmenting data, such as random horizontal flipping and temporal up/downsampling, and the model was optimized using Adam optimizer with a preliminary learning rate and weight decay of $1e$-4 and $5e$-5, correspondingly, and $\alpha = 0{,}1$.

We select sample face clips with dimensions of 320x72x72 ($T \times H \times W$), as this size is suitable for all three pre-trained models: In this context, the EfficientPhys model was specifically trained on samples of this size, making it appropriate for our purposes. The ContrastPhys model can accommodate any sample size, as its structure is not dependent on the input sample's size, allowing us to directly use the same size. In contrast, the PhysFormer model was originally trained using an input size of 160×128×128. After passing through the header modules, the model produces a final feature of size 96x40x4x4, where 96 represents the depth of the features (D), and 40 represents $\frac{T}{4}$. By using an input sample size of 320x72x72, we can maintain the same performance as the previous modules while altering only the size of the final feature to 96x80x2x2. This sample size does not impact the performance of the previous modules in any way. Finally, passing the final feature through the predictor module of the PhysFormer model produces the rPPG signal.

3.5. Image Restoration Methods

This section focuses on discussing methods for improving the accuracy of HR estimates by removing noise and inpainting facemasks region from facial images. The objective is to improve the effectiveness of the method that has been found to yield the best results in preprocessing low fps datasets, as well as transformation datasets. To this end, a restoration method is performed on both noise and facemask datasets based on the results table that is presented in the next section. To remove noise from the datasets, two approaches are utilized. The first one is a traditional method that involves

using well-known denoising algorithms. The second approach is to use state-of-the-art pre-trained DL models to denoise the images. However, for facemask datasets, only the traditional method is used for inpainting, as DL-based inpainting models trained on entirely different data can lead to unrealistic and deformed faces. By employing these noise and facemask removal methods, we aim to improve the overall performance of the HR estimation model by minimizing the impact of external factors that may interfere with the accurate estimation of HR from facial videos. The results of this preprocessing step are presented in the following section, where the impact of these methods on the precision and reliability of the HR estimation pre-trained DL model is discussed.

3.5.1. Non-Local Mean Denoising

This study uses the non-local means (NLM) algorithm [61] which is a popular image-denoising technique that is widely used in the field of signal processing and computer vision. It is a non-parametric method that uses redundancy in natural image statistics to remove noise from an image. The algorithm is based on the principle that similar patches in an image have similar noise characteristics, and therefore, averaging these patches can effectively remove the noise while preserving the image structure. The NLM algorithm has been shown to be effective in removing Gaussian, impulse, and mixed noise from images and has been applied in various applications, such as medical imaging, video processing, and computer graphics. The objective of image-denoising techniques is to restore the original image from a noisy measurement represented as Equation 32, where $v(i)$ is the observed value, $u(i)$ is the actual value, and $n(i)$ represents the noise perturbation at pixel i. A common approach to represent noise in digital images is to add Gaussian white noise, where the values of $n(i)$ are independent and identically distributed Gaussian values with a zero mean and a variance of σ^2.

$$v(i) = u(i) + n(i) \tag{32}$$

The algorithm outlines a denoising technique, D_h, which decomposes the noisy image v into $D_h v$ and $n(D_h, v)$. Here, h is a filtering parameter that typically relies on the noise's standard deviation. The ideal scenario is that $D_h v$ is smoother than v, while $n(D_h, v)$ is similar to white noise. To explain further, the NLM algorithm estimates a denoised value for a pixel i in a discrete noisy image v, denoted as $NL[v](i)$. This value is obtained by taking a weighted average of all the pixels in the image, where the weights $w(i, j)$ depend on the similarity between pixels i and j. These weights satisfy the usual conditions, such as being non-negative and summing up to 1. The similarity between two pixels i and j is based on the similarity of their intensity gray level vectors $v(\mathcal{N}_i)$ and $v(\mathcal{N}_j)$, where $\mathcal{N}_i$ and $\mathcal{N}_j$ are square neighborhoods centered at pixels i and j, respectively. To apply the Euclidean distance to the noisy neighborhood, the NLM algorithm utilizes a fixed-size square neighborhood, denoted as $\mathcal{N}_k$, centered at a pixel k. This similarity is determined by a decreasing function of the weighted Euclidean distance between the intensity vectors, i.e., $\|v\left(\mathcal{N}_i\right) - v\left(\mathcal{N}_j\right)\|_{2,a}^2$, where $a >$ 0 represents the standard deviation of the Gaussian kernel which leads to the following equation:

$$E\left\|v\left(\mathcal{N}_i\right)-v\left(\mathcal{N}_j\right)\right\|_{2,a}^2=\left\|u\left(\mathcal{N}_i\right)-u\left(\mathcal{N}_j\right)\right\|_{2,a}^2+2\sigma^2 \tag{33}$$

This equation demonstrates the effectiveness of the NLM algorithm in maintaining the order of similarity between pixels by using the Euclidean distance to calculate the weights between pixels. Pixels that have similar grey-level neighborhoods to the pixel being denoised are given higher weights in the weighted average. The normalizing constant is represented by the value of $Z(i)$, while the filtering degree is controlled by the parameter h, which determines the decay rate of the exponential function and the weights as a function of the Euclidean distance. The weights and normalizing constant are determined through the following formula:

$$\begin{aligned} w(i,j)&=\frac{1}{Z(i)}e^{-\frac{\left\|v\left(\mathcal{N}_i\right)-v\left(\mathcal{N}_j\right)\right\|_{2,a}^2}{h^2}}\\ Z(i)&=\sum_j e^{-\frac{\left\|v\left(\mathcal{N}_i\right)-v\left(\mathcal{N}_i\right)\right\|_{2,a}^2}{h^2}} \end{aligned} \tag{34}$$

In this research, the NLM algorithm is applied by utilizing the fastNlMeansDenoisingColored function in OpenCV as illustrated in Fig. 13. The parameter which controls the filter strength for the luminance component and color is set to 30. The size of the template patch used for computing weights is 7 pixels and the size of the window used for computing the weighted average for a given pixel is 21 pixels. The image is first converted to the CIELAB colorspace and then the NLM algorithm is applied separately to denoise the L and AB components using the specified parameters.

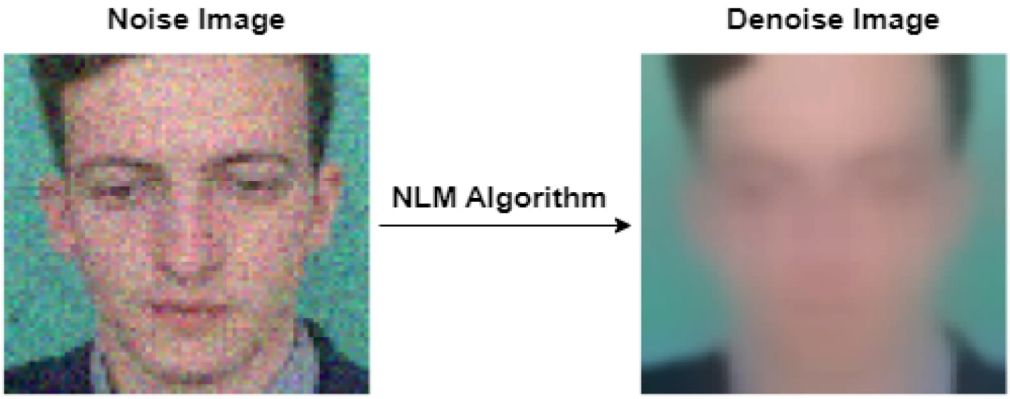


Figure 13. An instance of a denoised image using NLM algorithm.

3.5.2. NAFNet Model Denoising

The NAFNet model [62] is a cutting-edge DL network that has demonstrated remarkable performance with low inter-block and intra-block complexity for various multi-image restoration tasks, including denoising, deblurring, and super-resolution.

In this particular context, our focus is on its denoising capabilities. The model was designed by decomposing SOTA methods and extracting essential components, which were used to create the baseline model, a classic single-stage U-shaped architecture [63] with skip connections. Upon further analysis, the author found that the baseline model can be simplified by removing nonlinear activation functions. This resulted in the creation of NAFNet, a nonlinear activation-free network that uses a simple gate consisting of an element-wise product of feature maps, replacing the GELU and Channel Attention Module demonstrated in Fig. 14. This allowed for a reduction in complexity while maintaining high accuracy, making the NAFNet a promising model for various applications.

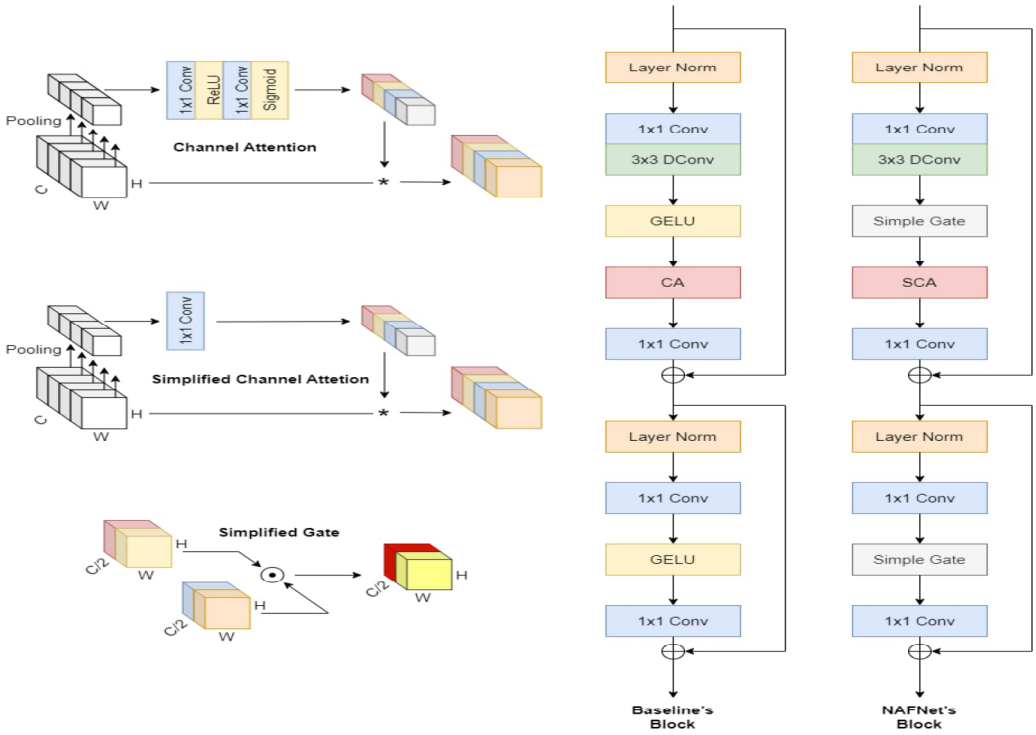


Figure 14. Illustration of Intra-block structure Adapted and modified from [62].

In this thesis, the NAFNet pre-trained model with 32-layer widths was employed for denoising shown in Fig. 15. The model was trained on the SIDD dataset [64] using a batch size of 64 and a total of 400K training iterations. To enhance the model's performance, random crop augmentation is applied during training.

3.5.3. Fast Marching Inpainting

The fast marching method (FMM) [65] is an image restoration technique used to fill in missing or damaged regions of an image. The FM algorithm is a partial differential equation-based approach that propagates image information from the surrounding areas of the missing region to fill in the gaps. It achieves this by iteratively solving the Laplace equation and using a diffusion process to gradually fill in the missing pixels. Compared to other inpainting methods, the FF algorithm is relatively fast and

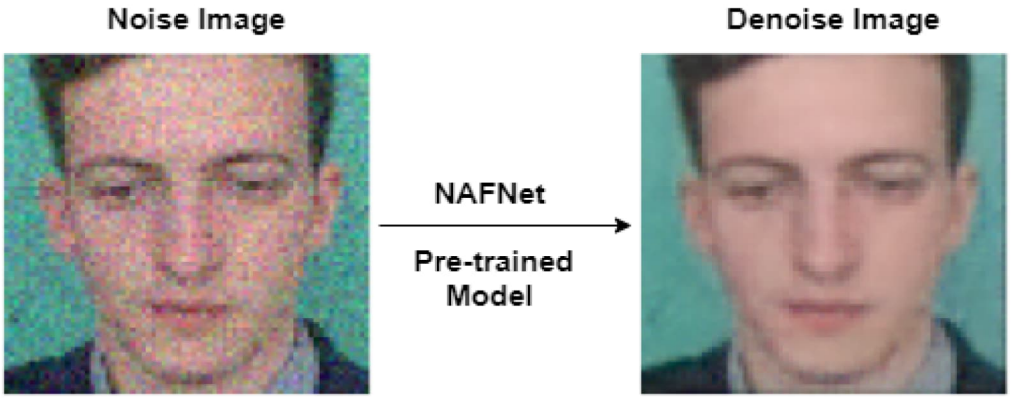


Figure 15. An example of a denoised image using NAFNet pre-trained model.

produces high-quality results, making it a popular choice for image restoration tasks. The FFM is applied by using the INPAINT_TELEA function in OpenCV as illustrated in Fig. 16. To describe in detail FMM, let's refer to Fig. 17, which shows a point p located on the boundary of a region to be inpainted.

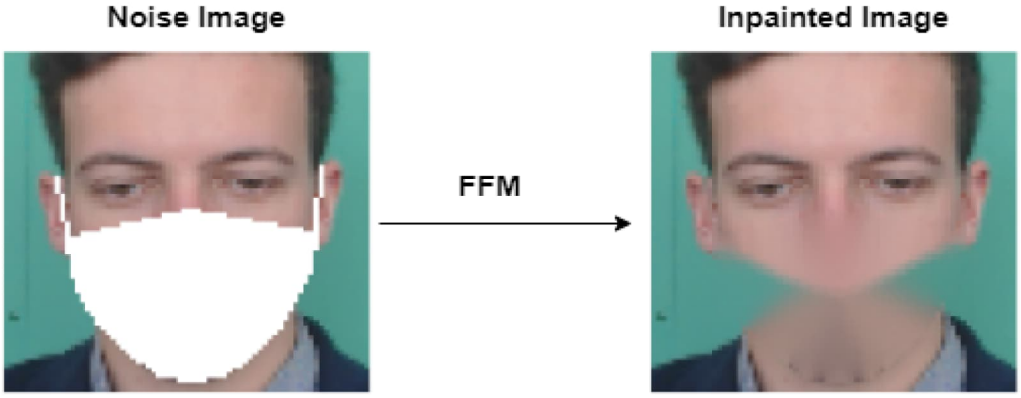


Figure 16. Example of an inpainted image using FFM.

To inpaint p, FFM considers a small neighborhood $B_\varepsilon(p)$ of size ε around the known image. The method initially focuses on gray-value images and can be extended to color images. By choosing a sufficiently small value of ϵ, the image in point p can be approximated using a first-order approximation $I_q(p)$, where q represents the image and gradient $\nabla I(q)$ values of a point within the neighborhood $B_\epsilon(p)$ that is closest to p. The next step is to fill in the missing point p by taking into account all points q within the neighborhood $B_\varepsilon(p)$. This is done by calculating the estimates of all points q, and then weighting them with a normalized weighting function $w(p,q)$ before summing them up as demonstrated in the below formula:

$$\begin{aligned} I_q(p) &= I(q) + \nabla I(q)(p-q) \\ I(p) &= \frac{\sum_{q\in B_\varepsilon(p)} w(p,q) I_q(p)}{\sum_{q\in B_\varepsilon(p)} w(p,q)} \end{aligned} \tag{35}$$

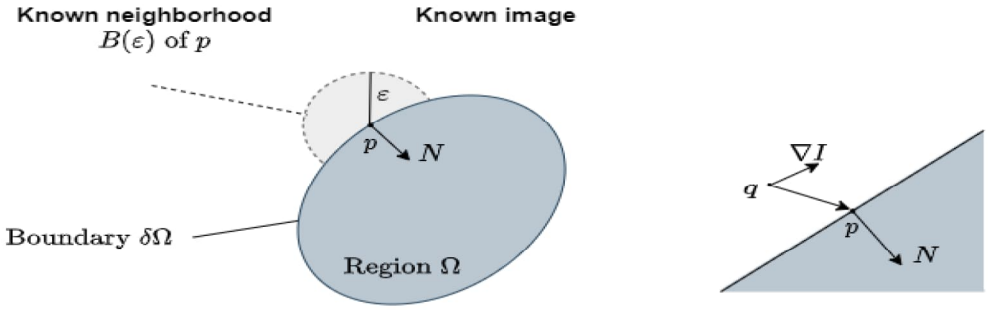


Figure 17. The principle of FFM Redrawn from [65].

3.6. Fine-Tuning Pre-Trained Models

This section discusses the experimental process of fine-tuning the best-performing pre-trained model, EfficientPhys. We start by outlining the basic parameters used, including frame depth, learning rate, and batch size. The frame depth is set to 10, the learning rate to $1e$-3, and the batch size to 640 video frames, equivalent to 2 videos. Based on EfficientPhys's structure, I divide it into 3 different fine-tuning methods as shown in Fig. 18: fine-tune all layers, fine-tune from the second convolution layer cluster, and fine-tune only the dense layers cluster.

Our focus is on improving denoise and inpaint face mask data, which are fine-tuned on the whole UBFC-rPPG data since EfficientPhys has been pre-trained on this data with 30 fps. The UBFC-rPPG data was split with a ratio of 8:2 for training, validation, and testing on the other datasets. Fine-tune data was created in two ways: fine-tune by data type (denoise or inpaint) or combine denoise and inpaint face mask data to fine-tune resulting in a total of six experimental approaches for each transformational data type.

Figure 18. Strategies for fine-tuning pre-trained EfficientPhys.

4. EXPERIMENTS

This section provides a detailed overview of the three datasets used in this study: UBFC-rPPG, ULCA-rPPG, and UBFC-Phys as shown in Table 3. Additionally, it discusses the four commonly used metrics for HR estimation, which include mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), Pearson correlation coefficient (ρ), and two widely used quality assessment for image restoration. Finally, the section presents the results of the rPPG methods used in this study for low frame rates original data, and transformed data.

Table 3. A overview of rPPG datasets

	Subjects	Video per subject	Total
UBFC-rPPG	50	1 video	50 videos
UCLA-rPPG	98	4-5 videos	488 videos
UBFC-Phys	56	3 videos	168 videos

4.1. Experimental Setup

4.1.1. Databases

UBFC-rPPG [49]: The dataset used in this study contains 50 videos that were synchronized with a pulse oximeter finger clip sensor to establish ground truth. Each video has a length of approximately 2 minutes and was recorded at a frame rate of 28-30 Hz, with a resolution of 640x480 in uncompressed 8-bit RGB format. To create two distinct subsets, the authors divided the dataset into two groups. The first subset, UBFC1-rPPG, consists of eight videos in which participants were instructed to remain stationary with a signal sample rate of 62. The second subset, UBFC2-rPPG, includes 42 videos with a signal sample rate of 29-30 in which participants played a time-sensitive mathematical game designed to increase their HR while also simulating a realistic human-computer interaction scenario. In both subsets, participants sat facing a camera positioned at a distance of approximately one meter. Fig. 19 demonstrates some images of subjects in the datasets.

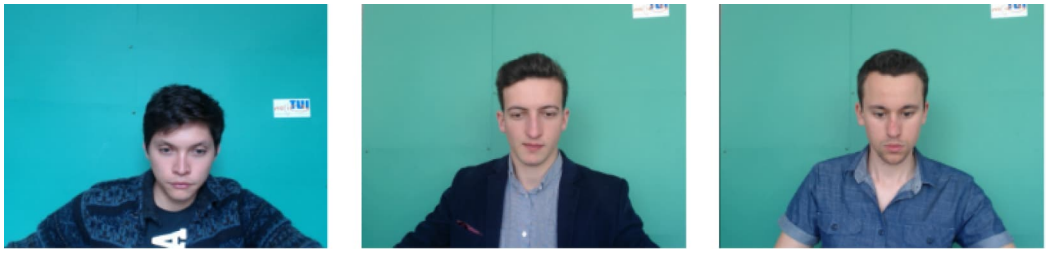


Figure 19. UBFC-rPPG image samples.

UCLA-rPPG [50]: The dataset includes a diverse set of 98 subjects, spanning a range of skin tones, ages, genders, ethnicities, and races. The subjects were classified

using the Fitzpatrick (FP) skin type scale, which ranges from 1-6. For each subject, 4-5 videos were recorded, each lasting about 1 minute and consisting of 1790 frames at 30fps. After removing any erroneous videos, a total of 488 videos were included in the dataset. All videos in the dataset are uncompressed and synchronized with ground truth HR data. To reduce redundancy in the dataset, only one video per subject was selected for analysis. Specifically, the second video from each subject was used, resulting in a total of 98 videos. Multiple instances of dataset subjects for demonstration are shown in Fig. 20, with the subjects' eyes being obscured as per request.

Figure 20. UCLA-rPPG image samples.

UBFC-Phys [51]: The dataset consists of videos of 68 undergraduate psychology students. The participants were recorded with an EO-23121C RGB digital camera by Edmund Optics, which used MotionJPEG compression and a frame rate of 35 frames per second. The signal was sampled at a rate of 65, and the frame resolution was 1024 x 1024 pixels. An artificial light source was used to ensure consistent lighting conditions for all participants. During the experiment, participants experienced social stress through a three-step process involving a resting task (T1), a speech task (T2), and an arithmetic task (T3). All videos for each subject were included in the analysis and divided into three subsets based on the task performed: UBFC-Phys T1, UBFC-Phys T2, and UBFC-Phys T3. The figure presented below shows several image examples from UBFC-Phys:

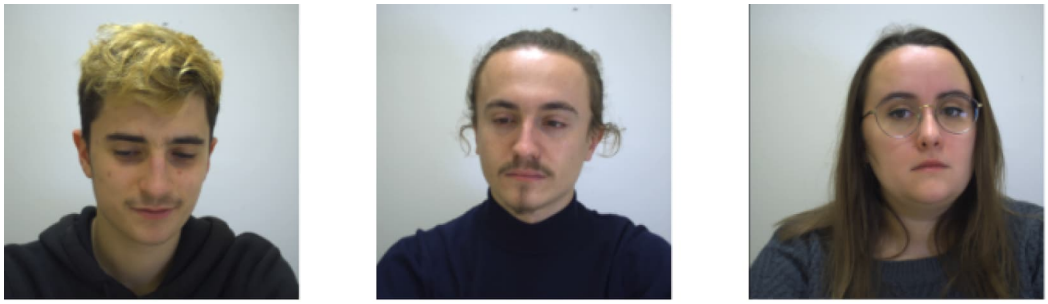


Figure 21. UBFC-Phys image samples.

4.1.2. Evaluation Metrics

This context provides a brief summary of the four common metrics used to assess the performance of rPPG methods, as well as two regularly used metrics to evaluate image quality.

rPPG Evaluation Metrics: The estimated rPPG waveform obtained from the rPPG method is filtered by applying a 2nd-order Butterworth filter with cut-off frequencies of 0.75 and 2.5 Hz to achieve the predicted HR. To measure the quality of the estimated HR (in BPM), the predicted HR is denoted as $\hat{h}(t)$, with respect to the ground truth HR which is $h(t)$. The estimation performance in each video was evaluated using the following measures:

MAE is computed by dividing the sum of the absolute errors by the signal size which can be expressed mathematically as $\mathrm{MAE} = \frac{1}{N}\sum_t \left|\hat{h}(t) - h(t)\right|$.

RMSE is a measure of the difference between quantities, expressed as the square root of the average of the squared differences. Mathematically, it can be represented as $\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_t(\hat{h}(t) - h(t))^2}$.

MAPE, also known as the mean absolute percentage deviation (MAPD), is a metric that quantifies the accuracy as a ratio computed using the below equation: $\mathrm{MAPE} = \frac{100\%}{N}\sum_t \left|\frac{h(t)-\hat{h}(t)}{h(t)}\right|$.

PPC or ρ is a measure of the linear correlation between the estimate $\hat{h}(t)$ and the ground truth $h(t)$. It is calculated using the following formula

$$\mathrm{PCC} = \frac{\sum_t(\hat{h}(t) - \hat{\mu})(h(t) - \mu)}{\sqrt{\sum_t(\hat{h}(t) - \hat{\mu})^2}\sqrt{\sum_t(\hat{h}(t) - \mu)^2}}$$

where $\hat{\mu}$ and μ are the mean values of the estimates and ground truth, respectively.

Image Quality Metrics: In this study, the quality of images before and after restoration is assessed using the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

PSNR is used in engineering to refer to the proportion of the maximum power of a signal to the power of any noise that can corrupt its accuracy. The PSNR between a reference image f and a test image g, both of which have a size of $M\mathrm{x}N$, is determined by the following formula:

$$\mathrm{PSNR}(f,g) = 10log_{10}(\tfrac{255^2}{\mathrm{MSE}(f,g)})$$
$$\mathrm{MSE}(f,g) = \tfrac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2$$

SSIM is utilized to estimate the perceived quality of various digital images and videos, including those used in television, cinema, and other domains. The evaluation of image quality is based on a reference image that is uncompressed or free from distortions as shown below equation:

$$\mathrm{SSIM}(f,g) = l(f,g)c(f,g)s(f,g)$$
$$\begin{cases} l(f,g) = \frac{2\mu_f\mu_g+C_1}{\mu_f^2+\mu_g^2+C_1} \\ c(f,g) = \frac{2\sigma_f\sigma_g+C_2}{\sigma_f^2+\sigma_g^2+C_2} \\ s(f,g) = \frac{\sigma_{fg}+C_3}{\sigma_f\sigma_g+C_3} \end{cases}$$

Where the first function, $l(f, g)$, calculates the similarity between the mean luminance of the two images (μ_f and μ_g). The second function, $c(f, g)$, determines the similarity between the contrasts of the images based on the standard deviations (σ_f and σ_g). The third function, $s(f, g)$, assesses the similarity in structure between the two images, measuring the correlation coefficient between f and g. The positive constants C_1, C_2, and C_3 are used to avoid a zero denominator.

4.2. Experimetal Results

4.2.1. Evaluation Original Datasets

In this section, we conducted a comparison of various rPPG methods with different measures of original rPPG datasets. The results, as shown in Table 4 and Table 5, reveal that the EfficientPhys pre-trained model surpasses other methods in terms of MAE, RMSE, MAPE, and ρ on UBFC-rPPG, UCLA-rPPG, and UBFC-Phys T1 datasets. Additionally, the model also achieved quite good results on UBFC-Phys T2 and T3 datasets compared to other methods. Among the traditional methods, POS appears to be the best performer, with the lowest values for all four metrics on UBFC-rPPG and UCLA-rPPG datasets. However, CHROM methods slightly outperformed POS on UBFC-Phys T1 datasets. It is noteworthy that the results on all methods for UBFC-Phys T2 and T3 datasets, which correspond to speech and arithmetic actions, are not very satisfactory. It should be noted that according to Table 5, the PhysFormer pre-trained model produced the lowest result on UBFC-Phys T2 and T3 datasets. However, the model's results are not valid as it provides almost the same result of approximately 72-77 HR (BPM) for different videos as inputs, suggesting that the model may have to overfit its training data and is unable to generalize to this new data.

Table 4. The result of various rPPG methods on the original UBFC-rPPG & UCLA-rPPG datasets

	UBFC-rPPG				UCLA-rPPG			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Traditional Method								
GREEN [20]	29.77	40.08	27.64	-0.14	8.99	15.70	10.98	0.28
LGI [55]	27.75	38.85	26.05	0.03	4.26	11.15	5.04	0.55
PBV [22]	26.16	38.48	24.25	0.00	7.94	14.32	9.97	0.39
ICA [13]	21.14	32.29	19.35	0.12	4.57	10.73	5.45	0.58
CHROME [21]	10.12	21.56	9.04	0.46	2.15	5.63	2.72	0.87
POS [11]	<u>9.75</u>	<u>19.27</u>	<u>8.88</u>	<u>0.57</u>	<u>1.46</u>	<u>4.46</u>	<u>1.93</u>	<u>0.92</u>
DL Method								
EfficientPhys [31]	**5.11**	**13.05**	**4.75**	**0.78**	**0.57**	**1.74**	**0.81**	**0.99**
ContrastPhys [57]	26.02	37.79	23.25	-0.29	5.43	12.12	6.30	0.45
PhysFormer* [58]	23.81	27.96	23.11	0.17	9.23	11.53	13.00	-0.02

Note: The bold number represents the best result obtained for each dataset, while the underline number represents the second-best result.

After analyzing the results presented in the two tables above, we have identified the two methods that achieved the best overall performance on the original data. These

Table 5. The result of various rPPG methods on the original UBFC-Phys T1, T2, T3 datasets

	UBFC-Phys T1				UBFC-Phys T2				UBFC-Phys T3			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Traditional method												
GREEN [20]	16.95	25.04	19.45	-0.01	18.5	22.79	23.75	0.02	19.63	23.74	24.11	0.30
LGI [55]	7.99	15.81	9.68	0.51	17.2	21.89	23.31	0.13	14.86	20.53	**18.32**	**0.37**
PBV [22]	12.39	21.53	14.63	0.12	16.78	20.97	22.03	**0.25**	20.13	25.41	25.55	0.06
ICA [13]	10.09	17.99	12.62	0.43	16.62	21.19	22.63	-0.04	18.58	22.96	23.15	0.08
CHROME [21]	6.03	13.06	8.45	0.58	**15.03**	**19.76**	**22.57**	-0.05	14.94	21.17	21.88	-0.01
POS [11]	6.03	13.93	8.52	0.54	15.57	20.55	22.64	0.05	**14.65**	**20.42**	21.97	0.04
DL method												
EfficientPhys [31]	**4.35**	**9.46**	**6.65**	**0.83**	15.32	19.91	23.99	0.15	15.65	20.38	23.04	0.05
ContrastPhys [57]	11.55	19.70	13.76	0.09	15.65	20.03	20.71	0.00	16.41	21.08	20.74	-0.05
PhysFormer* [58]	12.89	15.98	16.06	-0.10	11.89	14.26	17.77	0.12	12.85	15.76	17.86	0.05

Note: The bold number represents the best result obtained for each dataset, while the underline number represents the second-best result.

methods are the EfficentPhys pre-trained model and the POS traditional method. Both methods have demonstrated outstanding performance in terms of different metrics on the UBFC-rPPG, UCLA-rPPG, and UBFC-Phys T1 datasets. To further evaluate the effectiveness of these two methods, we apply them to transformed datasets. These transformed datasets contain motion blur, noise artifacts, occluded eyes, and facemasks which are known to be more challenging to deal with. By the effectiveness of the EfficentPhys pre-trained model and the POS traditional method on these transformed datasets, we gain a better understanding of their capabilities and limitations.

4.2.2. Evaluation Transformed Datasets

Based on the comparison of the results presented in Tables 6 and 7, we can observe that the EfficientPhys pre-trained model delivers the best results for UBFC-rPPG, UCLA-rPPG, and UBFC-Phys T1 datasets when the participant is at rest, regardless of whether the data is blurred or occluded eyes are present. However, for noisy and face mask data, both EfficientPhys and POS methods are affected, but the POS method performs relatively better than EfficientPhys, except for the UBFC-rPPG dataset with face masks. Regarding the UBFC-Phys T2 and T3 datasets, some noteworthy observations can be made. Initially, the results of the POS method showed a slight improvement when face masks were applied to the original data. This suggests that when there is significant head movement in the images, applying face masks can further enhance the performance of the POS method. Despite this, the improvement is not significant, and the results remain relatively poor for datasets with substantial head movements.

4.2.3. Evaluation Restored Datasets

Table 8 illustrates that the NAFNet pre-trained model denoise method leads to a significant enhancement in image quality, while the NLM method does not provide much improvement, possibly due to the luminance component and color set to 30 as mentioned earlier. Nonetheless, the EfficientPhys pre-trained model performs well on blur datasets as shown in earlier tables, resulting in significant improvements for most datasets, even if the denoised image is blurred, with the UBFC-rPPG dataset achieving the best results. However, its improvement is not significant for the UBFC-Phys T2

Table 6. The result of EfficientPhys and POS method on the transformed UBFC-rPPG & UCLA-rPPG datasets

	UBFC-rPPG				UCLA-rPPG			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Original Data								
EP	**5.11**	**13.05**	**4.75**	**0.78**	0.57	1.74	0.81	**0.99**
POS	9.75	19.27	8.88	0.57	1.46	4.46	1.93	0.92
Gaussian Blur Data								
EP	5.30	13.18	4.97	0.76	**0.53**	**1.61**	**0.76**	**0.99**
POS	10.22	20.32	9.32	0.54	1.48	4.57	2.05	0.92
Gaussian Noise Data								
EP	19.69	24.71	20.21	0.08	14.54	20.37	21.80	0.06
POS	18.66	26.62	16.92	0.28	4.4	8.29	6.02	0.69
Eyes Mask Data								
EP	5.39	13.66	5.01	0.74	1.29	3.74	1.78	0.94
POS	12.38	21.63	11.61	0.50	1.46	4.07	1.89	0.93
Face Mask Data								
EP	10.69	18.71	10.56	0.55	13.13	23.91	18.83	0.32
POS	15.38	25.28	13.86	0.31	2.92	6.74	3.78	0.81

Note: EP = EfficientPhys. For each dataset, the bold number represents the best result out of five types of data, while the underline number represents the best result between the EfficientPhys and POS methods for each transformation.

Table 7. The result of EfficientPhys and POS method on the transformed UBFC-Phys T1, T2, T3 datasets

	UBFC-Phys T1				UBFC-Phys T2				UBFC-Phys T3			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Original Data												
EP	4.35	9.46	6.65	**0.83**	15.32	19.91	23.99	0.15	15.65	20.38	23.04	0.05
POS	6.03	13.93	8.52	0.54	15.57	20.55	22.64	0.05	14.65	20.42	21.97	0.04
Gaussian Blur Data												
EP	4.65	9.99	7.25	0.80	14.15	19.61	22.37	0.17	15.23	20.22	22.75	0.06
POS	6.03	13.93	8.52	0.54	16.11	20.97	23.58	0.04	14.44	20.29	21.65	0.07
Gaussian Noise Data												
EP	12.64	17.27	17.14	0.16	17.54	20.75	16.13	0.02	17.08	21.11	24.60	-0.01
POS	8.37	15.27	10.90	0.35	14.69	18.90	22.12	0.10	14.40	18.29	21.37	0.04
Eyes Mask Data												
EP	**4.31**	**9.40**	**6.62**	**0.83**	15.40	20.46	22.28	0.13	13.94	20.24	21.01	0.21
POS	6.07	13.10	8.32	0.57	13.14	17.70	19.20	**0.28**	15.99	21.22	23.04	-0.06
Face Mask Data												
EP	11.84	20.74	15.68	0.37	19.71	24.44	28.61	0.13	18.37	23.11	25.02	0.10
POS	6.19	12.74	8.77	0.59	**12.56**	**17.10**	**18.54**	0.20	**13.02**	**17.23**	**18.84**	**0.28**

Note: EP = EfficientPhys. For each dataset, the bold number represents the best result out of five types of data, while the underline number represents the best result between the EfficientPhys and POS methods for each transformation.

dataset. In terms of the POS method, the NAFNet and NLM methods do not offer much improvement in the UBFC-rPPG and UCLA-rPPG datasets, and only a slight improvement is observed with the NAFNet method in the UBFC-Phys T1 dataset. Although the NLM method helps improve the performance of POS for datasets T2 and T3, the improvement is only limited. These findings are summarized in Tables 9 and 10.

The use of the FFM for inpainting on facemask data has led to a slight improvement in image quality compared to the original, as indicated in Table 8. When combined with the EfficientPhys pre-trained model, Table 9 and 10 show that the FFM produces significant improvements in most datasets, with the exception of UBFC-Phys T2,

Table 8. The evaluation of the perceived quality of transformed images compared to original images

	UBFC-rPPG		UCLA-rPPG		UBFC-Phys T1		UBFC-Phys T2		UBFC-Phys T3	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Gaussian Noise & Denoise Data										
ND	29.06	0.64	29.17	0.67	29.07	0.64	29.07	0.65	29.07	0.65
NLM	29.32	0.63	29.79	0.66	29.7	0.66	29.64	0.66	29.67	0.66
NAFNet	**31.62**	**0.89**	**30.71**	**0.84**	**31.29**	**0.87**	**31.12**	**0.87**	**31.17**	**0.87**
Facemask & Inpainting Data										
FM	33.58	0.63	33.33	0.61	33.4	0.64	33.49	0.66	33.52	0.65
FFM	**34.24**	**0.78**	**33.99**	**0.76**	**33.89**	**0.78**	**33.8**	**0.76**	**33.79**	**0.76**

Note: ND = Noise Data, FM = Facemask Data, the bold number represents the best result of each transformation.

Table 9. The results of the restoration method's performance on the UBFC-rPPG and UCLA-rPPG datasets after transformation

	UBFC-rPPG				UCLA-rPPG			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Gaussian Noise Data								
POS	18.66	26.62	16.92	0.28	**4.40**	**8.29**	**6.02**	**0.69**
NN+POS	20.25	26.27	18.80	0.44	5.12	9.34	6.88	0.60
NLM+POS	19.97	27.40	18.51	0.25	5.52	9.92	7.32	0.56
EP	19.69	24.71	20.21	0.08	14.54	20.37	21.80	0.06
NN+EP	16.59	22.06	17.26	0.23	17.60	23.27	26.01	-0.01
NLM+EP	**12.05**	**17.60**	**11.48**	**0.63**	8.80	13.65	13.10	0.43
Face Mask Data								
POS	15.38	25.28	13.86	0.31	**2.92**	**6.74**	**3.78**	**0.81**
FFM+POS	14.67	23.84	13.50	0.39	3.99	8.97	5.47	0.68
EP	10.69	18.71	10.56	0.55	13.13	23.91	18.83	0.32
FFM+EP	**8.67**	**16.56**	**8.50**	**0.71**	5.07	10.43	6.50	0.64

Note: NN = NAFNet, EP = EfficientPhys. The bold number represents the best result obtained for each type of transformation, while the underline number represents the best improvement achieved by applying the restoration method.

which only shows minor improvement. Notably, the UBFC-Phys T1 and T3 datasets produce results that are close to or even better than those achieved by the POS method in some metrics. Furthermore, the combination of the FFM with the EfficientPhys pre-trained model performs much better than the POS method on the UBFC-rPPG dataset, while the UCLA-rPPG dataset shows more than double the improvement, though not as good as the POS method. However, combining the FFM with the POS method does not appear to be effective, as the results show little to no improvement.

4.2.4. Evaluation Fine-Tuned Models

As previously mentioned, we performed experiments to fine-tune EfficientPhys in six different ways. These involved fine-tuning all layers, fine-tuning from the second

Table 10. The results of the restoration method's performance on the UBFC-Phys T1, UBFC-Phys T2, and UBFC-Phys T3 datasets after transformation

	UBFC-Phys T1				UBFC-Phys T2				UBFC-Phys T3			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Gaussian Noise Data												
POS	8.37	15.27	10.9	0.35	14.69	18.9	22.12	0.10	14.4	18.29	21.37	0.04
NN+POS	**7.37**	**12.65**	**9.87**	**0.57**	15.19	19.53	23.28	0.04	14.19	17.73	20.22	0.02
NLM+POS	9.58	16.19	12.04	0.25	**14.15**	**17.94**	**21.28**	0.13	**13.77**	**17.47**	**20.55**	0.14
EP	12.64	17.27	17.14	0.16	17.54	20.75	16.13	0.02	17.08	21.11	24.6	-0.01
NN+EP	16.53	22.85	22.32	-0.14	20.47	24.99	30.06	-0.25	17.03	21.04	25.30	0.09
NLM+EP	10.76	15.92	14.97	0.32	19.55	23.41	29.07	**0.14**	16.16	20.90	23.68	**0.19**
Face Mask Data												
POS	**6.19**	12.74	8.77	0.59	**12.56**	**17.1**	**18.54**	0.2	**13.02**	**17.23**	**18.84**	0.28
FFM+POS	**6.19**	13.12	**8.72**	0.58	13.39	18.55	20.47	**0.23**	14.4	19.28	20.91	0.10
EP	11.84	20.74	15.68	0.37	19.71	24.44	28.61	0.13	18.37	23.11	25.02	0.10
FFM+EP	6.57	**11.81**	9.10	**0.69**	18.46	22.28	26.57	-0.24	13.81	18.08	19.09	**0.33**

Note: NN = NAFNet, EP = EfficientPhys. The bold number represents the best result obtained for each type of transformation, while the underline number represents the best improvement achieved by applying the restoration method.

convolution layer cluster, and fine-tuning only dense layers on combined data. We also fine-tuned all layers, from the second convolution layer cluster, and the dense layers cluster on denoise or inpaint data. The results of these experiments are presented in Table 11 and 12.

Table 11. The evaluation of different fine-tuned methods on the restored UCLA-rPPG and UBFC-Phys T1 datasets

	UCLA-rPPG				UBFC-Phys T1			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Gaussian Noise data								
POS	**4.4**	**8.29**	**6.02**	**0.69**	-	-	-	-
NN+POS	-	-	-	-	**7.37**	**12.65**	**9.87**	**0.57**
NLM+EP	8.8	13.65	13.1	0.43	10.76	15.92	14.97	0.32
EP-FT1-DI	11.31	14.87	15.81	0.32	16.45	21.24	21.39	0.02
EP-FT2-DI	13.18	17.23	18.01	0.09	14.77	20.21	18.63	0.02
EP-FT3-DI	12.56	16.29	18.35	0.11	13.48	17.51	18.8	0.1
EP-FT1-D	10.91	14.14	15.92	0.16	12.18	15.83	16.57	0.23
EP-FT2-D	11.72	15.67	17.06	0.09	14.31	18.89	18.85	-0.01
EP-FT3-D	11.93	15.07	17.32	0.09	13.73	18.02	18.36	0.03
Face Mask Data								
POS	**2.92**	**6.74**	**3.78**	**0.81**	**6.19**	12.74	**8.77**	0.59
FFM+EP	5.07	10.43	6.5	0.64	6.57	**11.81**	9.1	**0.69**
EP-FT1-DI	10.31	14.8	12.97	0.26	16.11	21.7	19.47	0.03
EP-FT2-DI	10.74	15.82	13.91	0.02	15.53	20.71	18.79	0.17
EP-FT3-DI	5.60	11.25	7.05	0.54	8.75	15.25	11.44	0.45
EP-FT1-I	5.86	11.34	7.78	0.51	7.87	14.08	10.62	0.54
EP-FT2-I	6.19	12.28	7.90	0.43	9.46	16.03	12.24	0.42
EP-FT3-I	5.38	11.59	6.94	0.50	8.87	15.47	11.53	0.45

Note: FT 1, 2, 3 = Type of fine-tuned methods based on model structure, DI = Denoise + Inpainting Face Mask Data, D = Denoise Data, I = Inpainting Face Mask Data. The bold number represents the best result.

Our findings suggest that using only 320 frames per subject for fine-tuned is inadequate, but this is currently challenging to improve as my laptop lacks the computing power to handle larger datasets. Despite trying various fine-tuned methods,

Table 12. The evaluation of different fine-tuned methods on the restored UBFC-Phys T2 and UBFC-Phys T3 datasets

	UBFC-Phys T2				UBFC-Phys T3			
	MAE	RMSE	MAPE	ρ	MAE	RMSE	MAPE	ρ
Gaussian Noise Data								
NLM+POS	**14.15**	**17.94**	**21.28**	0.13	**13.77**	**17.47**	**20.55**	0.14
NLM+EP	19.55	23.41	29.07	**0.14**	16.16	20.90	23.68	**0.19**
EP-FT1-DI	16.53	20.07	23.66	0.12	16.66	20.26	22.18	0.13
EP-FT2-DI	15.61	19.53	21.99	0.03	18.96	22.88	25.80	-0.0
EP-FT3-DI	15.69	19.85	22.58	0.02	17.45	21.33	24.22	-0.09
EP-FT1-D	15.90	18.88	23.23	0.00	16.99	20.63	23.33	-0.06
EP-FT2-D	15.19	18.97	22.21	0.12	16.36	20.06	23.07	0.04
EP-FT3-D	16.91	20.11	24.57	-0.11	15.28	18.78	21.44	0.11
Face Mask Data								
POS	**12.56**	**17.10**	**18.54**	0.20	**13.02**	**17.23**	**18.84**	0.28
FFM+POS	-	-	-	**0.23**	-	-	-	-
FFM+EP	18.46	22.28	26.57	-0.24	13.81	18.08	19.09	**0.33**
EP-FT1-DI	14.56	18.98	20.38	-0.11	18.96	23.25	24.56	-0.12
EP-FT2-DI	16.45	20.14	23.09	-0.21	18.96	23.47	24.52	-0.11
EP-FT3-DI	14.44	18.41	21.10	0.10	16.78	21.59	22.80	-0.13
EP-FT1-I	14.23	19.13	20.42	0.00	14.40	19.34	18.65	0.23
EP-FT2-I	14.98	20.22	22.38	-0.04	14.73	18.75	19.41	0.27
EP-FT3-I	13.98	18.67	20.32	0.09	14.44	18.41	19.70	0.20

Note: FT 1, 2, 3 = Type of fine-tuned methods based on model structure, DI = Denoise + Inpainting Face Mask Data, D = Denoise Data, I = Inpainting Face Mask Data. The bold number represents the best result.

the results not only failed to improve but also decreased in some cases. Although there was a slight improvement in some metrics for UBFC-Phys T2, this improvement was not statistically significant. Moreover, we found that fine-tuning by combining denoise and inpaint data is not recommended. Our results show that the denoise data negatively impacts the inpaint data, resulting in low performance for both types of data.

4.3. Discussion

This section explores some intriguing details regarding the results tables presented earlier. Firstly, as mentioned before, in Table 6 and 7, transformed images with eyemask or with blur do not have a negative impact on the POS and EfficientPhys methods. It is worth noting that the transformed UCLA-rPPG datasets are less affected than other datasets when using the POS method. From personal observations, it appears that the UCLA-rPPG datasets have the best sitting posture with minimal external body movements, except for facial and eye muscles.

Moreover, Table 7 demonstrates that UBFC-Phys T2 and UBFC-Phys T3 yield unsatisfactory results on the original data, and thus, the application of POS and EfficientPhys combined with image restoration methods also produces unfavorable outcomes. However, it is evident that detecting the face region only three times for these datasets is inadequate since head movements can cause the cropped face to

be misplaced, affecting the estimated HR. Fig. 22 provides examples of misplaced cropped face images to illustrate this issue.

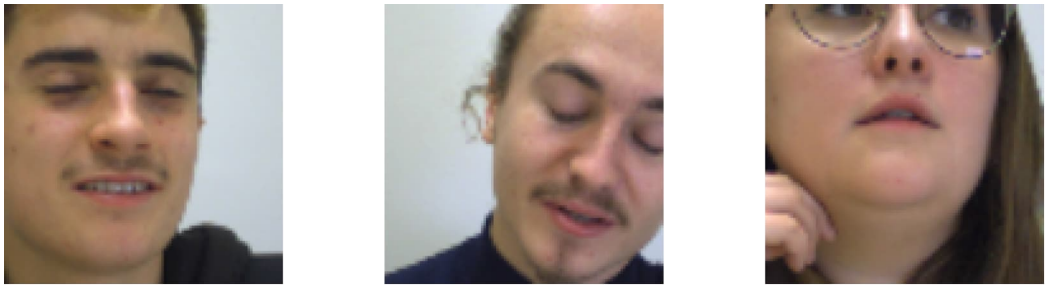
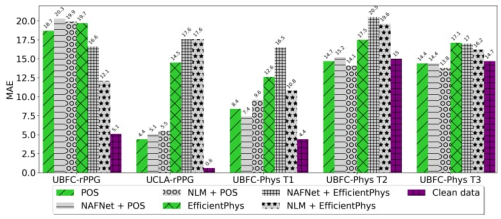
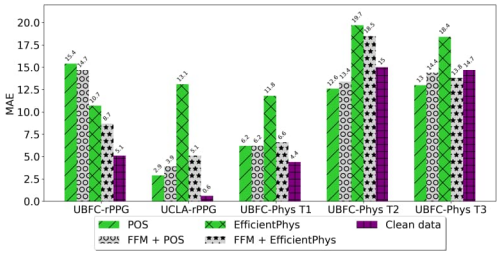


Figure 22. Illustrations of misplaced cropped face images.

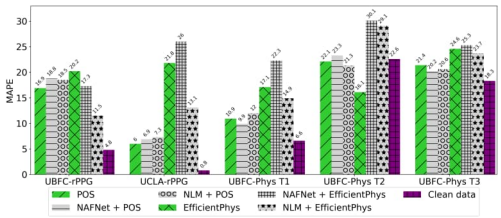
Regarding the rPPG results on degraded and occluded data, the most striking findings indicate that transformed images with noise or facemasks have an adverse impact on the data. Nevertheless, combining the EfficientPhys pre-trained model with NLM or FFM leads to significant improvements in datasets with less head movement, such as UBFC-rPPG, UCLA-rPPG, and UBFC-Phys T1. The result of these two types of transformation on three metrics was summarised in Fig. 23. The POS method is considered superior to the deep learning approach because it extracts the pulse signal from skin regions, making it less affected by facemasks compared to the EfficientPhys method. However, combining the FFM and POS methods may not be the best approach, according to the results.



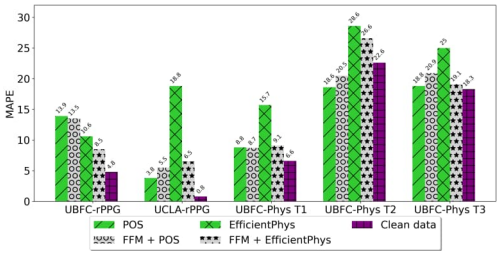
(a) Gaussian Noise MAE



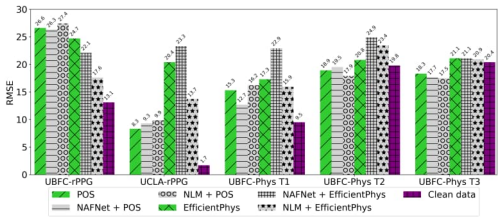
(b) Facemask MAE



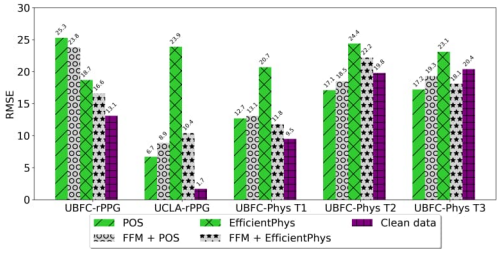
(c) Gaussian Noise MAPE



(d) Facemask MAPE



(e) Gaussian Noise RMSE



(f) Facemask RMSE

Figure 23. Results with Gaussian Noise and Facemask Data: Restoration Impact

5. CONCLUSION

Estimating HR using rPPG methods has gained significant attention in recent years, but research on their performance in various real-world scenarios, such as occluded faces due to sunglasses or facemasks and image degradation, remains limited. Therefore, this work aims to provide a comprehensive evaluation of different rPPG methods using custom-created datasets with various image transformations and restoration methods. We reviewed previous rPPG methods, including traditional and DL-based approaches, and evaluated their performance on our custom datasets.

After reviewing various methods, we selected two of the best-performing ones, the EfficientPhys and POS methods, and studied their performance on custom-created transformation datasets. During our evaluation, we found that EfficientPhys pre-trained models perform efficiently on original, blurred, and occluded image datasets, but are heavily affected by noisy and facemask datasets. On the other hand, the POS method yields decent results consistently across all datasets but after applying the image restoration method, it did not improve much, unlike EfficientPhys, which improved when combined with NLM and FFM methods. This combination improved the results on almost all datasets, and in some cases, the results were equal to those of the POS method, showing promise for further research. However, both methods did not perform well on datasets with significant head movements, such as UBFC-Phys T2 and T3. Additionally, we observed that using DL-based methods for image restoration was not suitable for improving the results of estimating HR, despite improving the quality of face images. Finally, our attempts to fine-tune EfficientPhys on restored data did not yield satisfactory results, likely due to the relatively small size of the fine-tuned dataset compared to the original pre-trained data. Unfortunately, at present, our computing resources are limited, and we cannot run larger datasets on our current laptop.

However, we plan to address current limitations in the future by experimenting with more powerful computational equipment, exploring suitable methods for denoising and inpainting data, and collecting additional data for fine-tuning and transfer learning. Moreover, the accuracy of the speech and arithmetic actions dataset can be enhanced by detecting face regions every second instead of three times, as this will better capture variations in HR and improve the reliability of rPPG methods. In addition to these improvements, there are several potential areas for exploration to further enhance the accuracy and robustness of HR estimation using rPPG methods. For instance, investigating different frame rates can determine whether a higher or lower frame rate improves HR estimation accuracy. Using better face landmark detection methods can divide the face into multiple regions of interest, enabling the evaluation of different rPPG methods on combinations of transformations, such as degradation + occlusion, noise + blur, and facemasks + eyemasks, to better understand how each transform affects different parts of the face in rPPG methods and how combinations of transformations degraded performance of rPPG methods.

Our study provides valuable insights into the performance of rPPG methods in different scenarios and highlights areas for further research to enhance the accuracy and robustness of estimating HR using rPPG methods.

6. REFERENCES

[1] Nina B., Kristiana S.R., Lorraine P.B., Ron M. & Jaime N. (2018) Pearson's comprehensive medical assisting: Administrative and clinical competencies 4th Edition, Pearson, chap. Vital Signs. pp. 798–828.

[2] Allen J. (2007) Photoplethysmography and its application in clinical physiological measurement. Physiological Measurement. DOI: https://doi.org/10.1088/0967-3334/28/3/R01.

[3] Olson J.A., Sandra D.A., Élissa S. Colucci, Al Bikaii A., Chmoulevitch D., Nahas J., Raz A. & Veissière S.P. (2022) Smartphone addiction is increasing across the world: A meta-analysis of 24 countries. Computers in Human Behavior. DOI: https://doi.org/10.1016/j.chb.2021.107138.

[4] Markus H. & Vladimir B. (2002) Contactless mapping of rhythmical phenomena in tissue perfusion using PPGI. In: A.V. Clough & C.T. Chen (eds.) Medical Imaging 2002: Physiology and Function from Multidimensional Images, pp. 110–117. DOI: https://doi.org/10.1117/12.463573.

[5] Sun Y. & Thakor N. (2016) Photoplethysmography Revisited: From Contact to Noncontact, From Point to Imaging. IEEE Transactions on Biomedical Engineering. DOI: https://doi.org/10.1109/TBME.2015.2476337.

[6] Maestre R., Rivera-Roman T., Fernandez-Jaramillo A., Guerrón N. & Serrano Olmedo J. (2020) A Non-Contact Photoplethysmography Technique for the Estimation of Heart Rate via Smartphone. Applied Sciences. DOI: https://doi.org/10.3390/app10010154.

[7] Nguyen N., Nguyen L., Alvarez Casado C., Silven O. & Bordallo Lopez M. (2023), Non-Contact Heart Rate Measurement from Deteriorated Videos. ArXiv preprint arXiv: 2304.14789.

[8] Zhang K., Zhang Z., Li Z. & Qiao Y. (2016) Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters. DOI: https://doi.org/10.1109/LSP.2016.2603342.

[9] Khanam F.T.Z., Al-Naji A.A. & Chahl J. (2019) Remote Monitoring of Vital Signs in Diverse Non-Clinical and Clinical Scenarios Using Computer Vision Systems: A Review. Applied Sciences. DOI: https://doi.org/10.3390/app9204474.

[10] Revanur A., Li Z., Ciftci U.A., Yin L. & Jeni L.A. (2021) The First Vision For Vitals (V4V) Challenge for Non-Contact Video-Based Physiological Estimation. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 2760–2767. DOI: https://doi.org/10.1109/ICCVW54120.2021.00310.

[11] Wang W., den Brinker A.C., Stuijk S. & de Haan G. (2017) Algorithmic Principles of Remote PPG. IEEE Transactions on Biomedical Engineering. DOI: https://doi.org/10.1109/TBME.2016.2609282.

[12] Chen X., Cheng J., Song R., Liu Y., Ward R. & Wang Z.J. (2019) Video-Based Heart Rate Measurement: Recent Advances and Future Prospects. IEEE Transactions on Instrumentation and Measurement. DOI: https://doi.org/10.1109/TIM.2018.2879706.

[13] Poh M.Z., McDuff D.J. & Picard R.W. (2010) Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. Opt. Express. DOI: https://doi.org/10.1364/OE.18.010762.

[14] Balakrishnan G., Durand F. & Guttag J. (2013) Detecting Pulse from Head Motions in Video. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3430–3437. DOI: https://doi.org/10.1109/CVPR.2013.440.

[15] Irani. R., Nasrollahi. K. & Moeslund. T.B. (2014) Improved pulse detection from head motions using dct. In: Proceedings of the 9th International Conference on Computer Vision Theory and Applications - Volume 3: VISAPP, (VISIGRAPP 2014), pp. 118–124. DOI: https://doi.org/10.5220/0004669001180124.

[16] Lin Y.C., Chou N.K., Lin G.Y., Li M.H. & Lin Y.H. (2017) A Real-Time Contactless Pulse Rate and Motion Status Monitoring System Based on Complexion Tracking. Sensors. DOI: https://doi.org/10.3390/s17071490.

[17] Guo Z., Wang Z.J. & Shen Z. (2014) Physiological parameter monitoring of drivers based on video data and independent vector analysis. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4374–4378. DOI: https://doi.org/10.1109/ICASSP.2014.6854428.

[18] Qi H., Guo Z., Chen X., Shen Z. & Jane Wang Z. (2017) Video-based human heart rate measurement using joint blind source separation. Biomedical Signal Processing and Control. DOI: https://doi.org/10.1016/j.bspc.2016.08.020.

[19] Chen X., Peng H., Yu F. & Wang K. (2017) Independent Vector Analysis Applied to Remove Muscle Artifacts in EEG Data. IEEE Transactions on Instrumentation and Measurement. DOI: https://doi.org/10.1109/TIM.2016.2608479.

[20] Verkruysse W., Svaasand L.O. & Nelson J.S. (2008) Remote plethysmographic imaging using ambient light. Opt. Express. DOI: https://doi.org/10.1364/OE.16.021434.

[21] de Haan G. & Jeanne V. (2013) Robust Pulse Rate From Chrominance-Based rPPG. IEEE Transactions on Biomedical Engineering. DOI: https://doi.org/10.1109/TBME.2013.2266196.

[22] Haan G. & Leest A. (2014) Improved motion robustness of remote-PPG by using the blood volume pulse signature. Physiological measurement. DOI: https://doi.org/10.1088/0967-3334/35/9/1913.

[23] Wang W., den Brinker A.C., Stuijk S. & de Haan G. (2017) Robust heart rate from fitness videos. Physiological Measurement. DOI: https://doi.org/10.1088/1361-6579/aa6d02.

[24] Casado C.A. & López M.B. (2022), Face2PPG: An unsupervised pipeline for blood volume pulse extraction from faces.

[25] Cheng C.H., Wong K.L., Chin J.W., Chan T.T. & So R.H.Y. (2021) Deep Learning Methods for Remote Heart Rate Measurement: A Review and Future Research Agenda. Sensors. DOI: https://doi.org/10.3390/s21186296.

[26] Zhan Q., Wang W. & de Haan G. (2020) Analysis of CNN-based remote-PPG to understand limitations and sensitivities. Biomed. Opt. Express. DOI: https://doi.org/10.1364/BOE.382637.

[27] Spetlik R., Franc V., Cech J. & Matas J. (2018) Visual Heart Rate Estimation with Convolutional Neural Network. In: British Machine Vision Conference, p. 84.

[28] Chen W. & McDuff D. (2018) DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks. In: V. Ferrari, M. Hebert, C. Sminchisescu & Y. Weiss (eds.) Computer Vision – ECCV 2018, pp. 356–373. DOI: https://doi.org/10.1007/978-3-030-01216-8_22.

[29] Liu X., Fromm J., Patel S. & McDuff D. (2020) Multi-Task Temporal Shift Attention Networks for On-Device Contactless Vitals Measurement. In: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan & H. Lin (eds.) Advances in Neural Information Processing Systems, pp. 19400–19411.

[30] Lin J., Gan C. & Han S. (2019) TSM: Temporal Shift Module for Efficient Video Understanding. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7082–7092. DOI: https://doi.org/10.1109/ICCV.2019.00718.

[31] Liu X., Hill B., Jiang Z., Patel S. & McDuff D. (2023) EfficientPhys: Enabling Simple, Fast and Accurate Camera-Based Cardiac Measurement. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 4997–5006. DOI: https://doi.org/10.1109/WACV56688.2023.00498.

[32] Yu Z., Li X. & Zhao G. (2019) Recovering remote Photoplethysmograph Signal from Facial Videos Using Spatio-Temporal Convolutional Networks. CoRR DOI: https://doi.org/10.48550/arXiv.1905.02419.

[33] Zhang P., Li B., Peng J. & Jiang W. (2021) Multi-hierarchical Convolutional Network for Efficient Remote Photoplethysmograph Signal and Heart Rate Estimation from Face Video Clips. CoRR DOI: https://doi.org/10.48550/arXiv.2104.02260.

[34] Hu M., Qian F., Guo D., Wang X., He L. & Ren F. (2021) ETA-rPPGNet: Effective Time-Domain Attention Network for Remote Heart Rate Measurement. IEEE Transactions on Instrumentation and Measurement. DOI: https://doi.org/10.1109/TIM.2021.3058983.

[35] Yu Z., Shen Y., Shi J., Zhao H., Torr P. & Zhao G. (2022) PhysFormer: Facial Video-based Physiological Measurement with Temporal Difference Transformer. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4176–4186. DOI: https://doi.org/10.1109/CVPR52688.2022.00415.

[36] Hu M., Guo D., Wang X., Ge P. & Chu Q. (2019) A Novel Spatial-Temporal Convolutional Neural Network for Remote Photoplethysmography. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–6. DOI: https://doi.org/10.1109/CISP-BMEI48845.2019.8966034.

[37] Tang C., Lu J. & Liu J. (2018) Non-contact Heart Rate Monitoring by Combining Convolutional Neural Network Skin Detection and Remote Photoplethysmography via a Low-Cost Camera. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1390–13906. DOI: https://doi.org/10.1109/CVPRW.2018.00178.

[38] Paracchini M., Marcon M., Villa F., Zappa F. & Tubaro S. (2020) Biometric Signals Estimation Using Single Photon Camera and Deep Learning. Sensors. DOI: https://doi.org/10.3390/s20216102.

[39] Yu Z., Peng W., Li X., Hong X. & Zhao G. (2019) Remote Heart Rate Measurement From Highly Compressed Facial Videos: An End-to-End Deep Learning Solution With Video Enhancement. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 151–160. DOI: https://doi.org/10.1109/ICCV.2019.00024.

[40] Sabokrou M., Pourreza M., Li X., Fathy M. & Zhao G. (2021) Deep-HR: Fast heart rate estimation from face video under realistic conditions. Expert Systems with Applications. DOI: https://doi.org/10.1016/j.eswa.2021.115596.

[41] Tsou Y.Y., Lee Y.A., Hsu C.T. & Chang S.H. (2020) Siamese-RPPG Network: Remote Photoplethysmography Signal Estimation from Face Videos. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing, pp. 2066—-2073. DOI: https://doi.org/10.1145/3341105.3373905.

[42] Lee E., Chen E. & Lee C.Y. (2020) Meta-rPPG: Remote Heart Rate Estimation Using a Transductive Meta-learner. In: A. Vedaldi, H. Bischof, T. Brox & J.M. Frahm (eds.) Computer Vision – ECCV 2020, pp. 392–409. DOI: https://doi.org/10.1007/978-3-030-58583-9_24.

[43] Huang B., Lin C.L., Chen W., Juang C.F. & Wu X. (2021) A novel one-stage framework for visual pulse rate estimation using deep neural networks. Biomedical Signal Processing and Control. DOI: https://doi.org/10.1016/j.bspc.2020.102387.

[44] Song R., Chen H., Cheng J., Li C., Liu Y. & Chen X. (2021) PulseGAN: Learning to Generate Realistic Pulse Waveforms in Remote Photoplethysmography. IEEE Journal of Biomedical and Health Informatics. DOI: https://doi.org/10.1109/JBHI.2021.3051176.

[45] Yang W., Li X. & Zhang B. (2018) Heart Rate Estimation from Facial Videos Based on Convolutional Neural Network. In: 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), pp. 45–49. DOI: https://doi.org/10.1109/ICNIDC.2018.8525602.

[46] Niu X., Shan S., Han H. & Chen X. (2020) RhythmNet: End-to-End Heart Rate Estimation From Face via Spatial-Temporal Representation. IEEE Transactions on Image Processing. DOI: https://doi.org/10.1109/TIP.2019.2947204.

[47] Lu H. & Han H. (2021) NAS-HR: Neural architecture search for heart rate estimation from face videos. Virtual Reality & Intelligent Hardware. DOI: https://doi.org/10.1016/j.vrih.2020.10.002.

[48] Niu X., Zhao X., Han H., Das A., Dantcheva A., Shan S. & Chen X. (2019) Robust Remote Heart Rate Estimation from Face Utilizing Spatial-temporal Attention. In: 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), pp. 1–8. DOI: https://doi.org/10.1109/FG.2019.8756554.

[49] Bobbia S., Macwan R., Benezeth Y., Mansouri A. & Dubois J. (2019) Unsupervised skin tissue segmentation for remote photoplethysmography. Pattern Recognition Letters. DOI: https://doi.org/10.1016/j.patrec.2017.10.017.

[50] Wang Z., Ba Y., Chari P., Bozkurt O.D., Brown G., Patwa P., Vaddi N., Jalilian L. & Kadambi A. (2022) Synthetic Generation of Face Videos with Plethysmograph Physiology. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20555–20564. DOI: https://doi.org/10.1109/CVPR52688.2022.01993.

[51] Meziati Sabour R., Benezeth Y., De Oliveira P., Chappe J. & Yang F. (2021) UBFC-Phys: A Multimodal Database For Psychophysiological Studies Of Social Stress. IEEE Transactions on Affective Computing. DOI: https://doi.org/10.1109/TAFFC.2021.3056960.

[52] King D.E. (2009) Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research. DOI: https://dl.acm.org/doi/10.5555/1577069.1755843.

[53] Geitgey A. (2019) Machine Learning is Fun! 2nd Edition. Machine Learning Is Fun! URL: https://www.machinelearningisfun.com/get-the-book/.

[54] Boccignone G., Conte D., Cuculo V., D'Amelio A., Grossi G. & Lanzarotti R. (2020) An Open Framework for Remote-PPG Methods and their Assessment. IEEE Access. DOI: https://doi.org/10.1109/ACCESS.2020.3040936.

[55] Pilz C.S., Zaunseder S., Krajewski J. & Blazek V. (2018) Local Group Invariance for Heart Rate Estimation from Face Videos in the Wild. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1335–13358. DOI: https://doi.org/10.1109/CVPRW.2018.00172.

[56] Liu X., Zhang X., Narayanswamy G., Zhang Y., Wang Y., Patel S. & McDuff D. (2022), Deep Physiological Sensing Toolbox. DOI: https://doi.org/10.48550/arXiv.2210.00716.

[57] Sun Z. & Li X. (2022) Contrast-Phys: Unsupervised Video-Based Remote Physiological Measurement via Spatiotemporal Contrast. In: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella & T. Hassner (eds.) Computer Vision – ECCV 2022, pp. 492–510. DOI: https://doi.org/10.1007/978-3-031-19775-8_29.

[58] Xiao T., Singh M., Mintun E., Darrell T., Dollar P. & Girshick R. (2021) Early Convolutions Help Transformers See Better. In: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang & J.W. Vaughan (eds.) Advances in Neural Information Processing Systems, pp. 30392–30400.

[59] Gao B.B., Xing C., Xie C.W., Wu J. & Geng X. (2017) Deep Label Distribution Learning With Label Ambiguity. IEEE Transactions on Image Processing. DOI: https://doi.org/10.1109/TIP.2017.2689998.

[60] Niu X., Han H., Shan S. & Chen X. (2019) VIPL-HR: A Multi-modal Database for Pulse Estimation from Less-Constrained Face Video. In: C. Jawahar, H. Li, G. Mori & K. Schindler (eds.) Computer Vision – ACCV 2018, pp. 562–576. DOI: https://doi.org/10.1007/978-3-030-20873-8_36.

[61] Buades A., Coll B. & Morel J.M. (2005) A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 60–65. DOI: https://doi.org/10.1109/CVPR.2005.38.

[62] Chen L., Chu X., Zhang X. & Sun J. (2022) Simple Baselines for Image Restoration. In: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella & T. Hassner

(eds.) Computer Vision – ECCV 2022, pp. 17–33. DOI: https://doi.org/10.1007/978-3-031-20071-7_2.

[63] Ronneberger O., Fischer P. & Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: N. Navab, J. Hornegger, W.M. Wells & A.F. Frangi (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234–241. DOI: https://doi.org/10.1007/978-3-319-24574-4_28.

[64] Abdelhamed A., Lin S. & Brown M.S. (2018) A high-quality denoising dataset for smartphone cameras. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1692–1700. DOI: https://doi.org/10.1109/CVPR.2018.00182.

[65] Telea A. (2004) An Image Inpainting Technique Based on the Fast Marching Method. Journal of Graphics Tools. DOI: https://doi.org/10.1080/10867651.2004.10487596.