Karlsruher Institut für Technologie

**Institut für Telematik**
**Pervasive Computing Systems / TECO**
**Leiter: Prof. Dr.-Ing. Michael Beigl**

# Investigation of Image Processing Techniques for Camera-Based Respiratory Rate Measurement with Machine Learning

## Master's Thesis

## by

**Patrick Zierahn**

**Department of Informatics**

| | |
|---|---|
| Responsible Supervisor: | Prof. Dr. Michael Beigl |
| Supervising Staff: | M.Sc. Gergely Biri/ Dr. Till Riedel |
| Project Period: | 17.04.2024 - 17.10.2024 |

Karlsruher Institut für Technologie

**Erklärung**

Hiermit erkläre ich, dass ich die vorliegende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und weiterhin die Richtlinien des KIT zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Karlsruhe, den _____

_____

Unterschrift

# Abstract

In recent years, non-contact health monitoring has emerged as a promising solution for the growing need for remote, unobtrusive medical assessments. Respiratory rate, a critical vital sign and early indicator of health complications, is traditionally monitored using invasive or uncomfortable sensors. This thesis investigates algorithms and machine learning models for extracting respiratory signals from video data, enabling contactless, reliable, and continuous monitoring. It compares various techniques including remote photoplethysmography (rPPG), optical flow, and transformer-based models using the VitalCam dataset. The results demonstrate that optical flow-based models, such as Lucas-Kanade and FlowNet2, outperform other approaches for robust respiration signal extraction. Transformer-based architectures, including SimpleViT and RhythmFormer, also achieve competitive accuracy. Furthermore, the thesis evaluates the efficacy of a hybrid loss function for training respiration models, identifying the cross-entropy Power Spectral Density (PSD) loss as the most effective. A novel scenario utilizing normalized facial data was introduced, but it performed poorly relative to settings that capture chest movement and background information. These findings contribute to the advancement of non-contact respiratory rate monitoring and open new avenues for remote healthcare applications.

# Contents

**Abstract** i

**1. Introduction** 1
1.1. Motivation . . . 1
1.2. Objectives . . . 2
1.3. Structure . . . 2

**2. Background** 5
2.1. Respiration as Vital Sign . . . 5
2.2. Neuronal Networks . . . 6
2.2.1. Convolutional Neural Networks . . . 7
2.2.2. Vision Transformers . . . 7
2.3. Optical Flow . . . 9
2.4. VitalCamSet Dataset . . . 9
2.5. Pearson's Correlation Coefficient . . . 10
2.6. T-Test . . . 11
2.7. Fourier Analysis . . . 12
2.8. Fast Fourier Transform . . . 13
2.9. Butterworth Filters . . . 14
2.10. Frequency Extraction from Vital Signs . . . 15
2.10.1. Power Spectral Density . . . 15
2.10.2. Peak Count . . . 16
2.10.3. Crossing Point . . . 17
2.10.4. Negative Feedback Crossing Point . . . 18

**3. Literature Review** 21
3.1. Respiration Extraction . . . 21
3.1.1. Optical Flow . . . 21
3.1.2. Remote Photoplethysmography . . . 22
3.2. Optical Flow Models . . . 23
3.2.1. Pixel Intensity . . . 23
3.2.2. Lucas-Kanade Algorithm . . . 23
3.2.3. FlowNet and FlowNet2 . . . 25
3.2.4. Recurrent All-Pairs Field Transforms . . . 26
3.3. rPPG Models . . . 27
3.3.1. DeepPhys . . . 27

3.3.2. EfficientPhys . . . 28
3.3.3. MTTS-CAN / TS-CAN . . . 28
3.3.4. BigSmall . . . 29
3.3.5. RhythmFormer . . . 30
3.4. Taxonomy of Extraction Methods . . . 31
3.5. The Research Gap . . . 32

**4. Contribution** **35**
4.1. Methodology . . . 35
4.1.1. Project Structure . . . 35
4.1.2. Implementation and Model Sources . . . 37
4.1.3. Pre-processing . . . 38
4.1.4. Filtering and Normalization . . . 40
4.1.5. Frequency Extraction . . . 41
4.2. Extraction respiration with RAFT . . . 43
4.2.1. Chest Detection . . . 43
4.2.2. Calculating Motion Vectors . . . 43
4.2.3. Extracting Vertical Motion Components . . . 44
4.2.4. Filtering and Normalization . . . 45
4.2.5. Frequency Extraction . . . 46
4.3. Respiration Transformer . . . 47
4.3.1. Model Architectures . . . 47
4.3.2. Scenarios . . . 48
4.3.3. Hybrid Loss Function . . . 50
4.3.4. Training . . . 52

**5. Analysis and Results** **55**
5.1. Direct Approach Comparison . . . 55
5.1.1. Correlation . . . 56
5.1.2. Bland-Altman plots . . . 57
5.1.3. Visualisation . . . 58
5.2. Group Comparison . . . 60
5.3. SimpleViT vs RhythmFormer . . . 64
5.4. Influence of Hybrid Loss Function components . . . 65
5.5. Setting Influence . . . 68
5.5.1. Direct Comparsion . . . 68
5.5.2. Grouped Comparison . . . 69
5.5.3. Conclusion . . . 70
5.6. Threats to Validity . . . 70

**6. Conclusion** **73**
6.1. Summery . . . 73
6.2. Future Work . . . 74
6.2.1. Spiking Neural Networks . . . 74

6.2.2. More Datasets . . . . . 74
6.2.3. Training PPG Models on VitalCam . . . . . 75
6.2.4. Mobile Deployment . . . . . 75

**A. Appendix** **77**
A.1. Model Metrics . . . . . 77
A.1.1. Optical Flow . . . . . 77
A.1.2. Respiration-RhythmFormer . . . . . 78
A.1.3. SimpleViT . . . . . 78
A.1.4. Pretrained rPPG Models . . . . . 78
A.1.5. Pretrained Respiration Models . . . . . 78
A.1.6. Random . . . . . 78

**Bibliography** **85**

# 1. Introduction

## 1.1. Motivation

In recent years, the healthcare sector has faced significant challenges, particularly in rural areas where access to medical care is often limited due to a shortage of doctors and medical facilities. This disparity in healthcare availability highlights the need for innovative solutions that can bridge the gap between patients and medical professionals, especially for routine monitoring of vital signs.

Among the various vital signs, respiratory rate stands out as a critical indicator of a person's health status. The respiratory rate is not only essential to assess cardiovascular health, but also serves as a valuable predictor of whether immediate medical attention is required [1]. As noted by Fieselmann et al., respiratory rate is a significant early indicator of conditions such as cardiopulmonary arrest [2]. Despite its importance, the respiratory rate remains one of the least frequently measured vital signs in clinical settings [3].

Traditional methods of measuring respiratory rate often involve contact-based sensors, which can be uncomfortable and impractical for continuous monitoring, especially in home environments. These conventional approaches may cause discomfort and lead to false alarms due to sensor detachment or motion artifacts [1]. Consequently, there is a growing demand for non-intrusive, contactless methods of respiratory rate measurement that can enable more frequent and comfortable monitoring. Camera-based systems have emerged as a promising solution due to their accessibility and cost-effectiveness [4].

The integration of machine learning algorithms with image processing techniques has further enhanced the capabilities of camera-based respiratory monitoring systems. Machine learning approaches can improve the accuracy of respiratory rate estimation and enable more robust performance in various real-world conditions [5]. Machine learing is especially good at recognizing complex patterns in visual data and generalizing across diverse environments and populations. The adaptability and scalability of machine learning thus open the door to increasingly sophisticated and personalized healthcare applications.

The development of reliable, non-contact respiratory rate measurement systems has significant implications for remote healthcare. This technology could enable a more widespread monitoring of patients in their homes, reducing the need for frequent

doctor visits, and allowing for earlier detection of potential health issues. This is particularly valuable in rural areas where access to medical facilities is limited [6].

## 1.2. Objectives

In light of these considerations, this thesis aims to investigate and develop image processing techniques combined with machine learning approaches for camera-based respiratory rate measurement. The primary contributions of this research are as follows:

1. Exploration, implementation and evaluation of existing video-based respiratory extraction methods on the VitalCamSet Dataset.
2. Evaluation of how well remote Photoplethysmography (rPPG) models, can be utilized to extract respiratory signals from videos.
3. Development of a respiratory extraction method that utilizes transformer-based optical flow.
4. Training and evaluation of end-to-end transformers that can extract respiratory signals from videos.

By addressing these contributions, this research seeks to enhance non-intrusive vital sign monitoring technologies, potentially improving access to healthcare and enabling more proactive health management for individuals in diverse settings.

## 1.3. Structure

The thesis is organized into several chapters, each focusing on a distinct aspect of the research, and collectively building a comprehensive exploration of non-contact respiratory rate monitoring. The structure is as follows:

1. **Introduction:** This chapter sets the stage by providing an overview of the motivation, objectives, and significance of research in the realm of non-contact respiratory monitoring. It also outlines the overall structure of the thesis.
2. **Background:** This chapter delves into fundamental concepts and related technologies, offering detailed insights into respiration as a vital sign, the basics of neural networks, optical flow techniques, photoplethysmography, and the datasets used. It sets the theoretical foundation required for the research conducted in subsequent chapters.

3. **Literature Review:** Here, the focus is on reviewing existing methodologies and advancements in respiratory signal extraction from video data. It discusses various optical flow and rPPG-based models and identifies research gaps, laying the groundwork for the contributions made by this study.
4. **Contribution:** This chapter elaborates on the methodologies and implementations specific to this research. It covers aspects such as data processing, model architectures, and the new approaches developed for improving respiration signal extraction from video datasets.
5. **Analysis and Results:** In this chapter, the performance of the different models and techniques is evaluated. Detailed results, statistical analyses, and visualizations are presented to assess the efficacy of each method. Comparison of models and critical insights into their performance are discussed.
6. **Conclusion:** Concluding the thesis, this chapter summarizes the key findings and contributions. It also outlines recommendations for future research endeavors and the potential impact of continued developments in non-contact respiratory monitoring technology.
7. **Appendix:** Supplementary materials, including detailed metrics, additional figures, and related data, are provided in the appendix. This section supports the main body of research with comprehensive evaluation metrics and model performance data.

Each chapter is carefully designed to build on the previous one, ensuring a logical flow and a clear understanding of the research conducted. This structured approach allows readers to follow the progression from theoretical foundations to applied research and findings, ultimately providing valuable insights into the advancement of non-contact respiratory monitoring techniques.

# 2. Background

The Background chapter provides a foundational understanding of concepts and technologies relevant to non-contact respiration monitoring. It begins with an overview of respiration as a vital sign, emphasizing its clinical significance. The chapter then delves into the basics of neural networks, including Convolutional Neural Networks (CNNs) and Vision Transformers. This is followed by a detailed explanation of optical flow methods, the VitalCamSet dataset, and the process of extracting respiration signals using Pearson's Correlation Coefficient, t-tests, Fourier analysis, Fast Fourier Transform (FFT), and Butterworth filters. Lastly, it covers various techniques for frequency extraction from vital signs, such as Power Spectral Density (PSD), Peak Count (PC), Crossing Point (CP), and Negative Feedback Crossing Point (NFCP).

## 2.1. Respiration as Vital Sign

The respiratory rate is a crucial vital sign that provides valuable information on a person's health status. It is considered one of the most sensitive indicators of physiological deterioration in patients. Changes in respiratory rate can precede and predict serious clinical events, making it an essential parameter for the early detection of deterioration in the patient [7].

The relevance of respiratory rate as a vital sign is underscored by its ability to predict adverse outcomes. Studies have shown that elevated respiratory rate is a specific predictor of cardiac arrest and unplanned intensive care unit admission [7]. Furthermore, the respiratory rate has been found to be more discriminatory between stable and unstable patients than other vital signs such as pulse rate [8].

Despite its importance, the respiratory rate remains the least measured vital sign in clinical practice [8]. This paradox can be attributed to several factors. Traditionally, respiratory rate measurement has relied on manual counting or contact-based sensors, which can be time consuming, prone to error, and potentially uncomfortable for patients [9]. The intrusiveness of conventional measurement techniques, such as nasal cannulas or chest straps, often leads to poor patient compliance and inconsistent monitoring [10].

The challenges associated with accurate and consistent respiratory rate measurement have animated research into non-contact monitoring methods. These include video-based techniques that analyze chest wall movements or subtle skin color changes associated with breathing [10]. Such contactless approaches aim to provide continuous, unobtrusive monitoring of respiratory rate, potentially improving the frequency and accuracy of measurements in both clinical and home settings.

## 2.2. Neuronal Networks

Neuronal networks are computational models inspired by the biological neural networks found in the brains of animals [11]. These networks consist of interconnected nodes or "neurons" organized in layers that process and transmit information. The fundamental structure of a typical neural network includes an input layer, one or more hidden layers, and an output layer.

The input layer is responsible for receiving and processing the initial data, with each node representing a feature or attribute of the input. The hidden layers, located between the input and output layers, perform most of the computational work in the network. Each hidden unit receives input from the previous layer, processes this information, and passes it to the next layer. The output layer produces the results of the computations, with the number of units depending on the specific task the network is designed to perform.

A crucial aspect of neural networks is the introduction of non-linearity, which allows them to learn and approximate complex patterns in data. This non-linearity is primarily achieved through the use of activation functions in the hidden layers. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). These functions determine whether and to what extent a neuron's signal should be propagated to influence the next layer.

The learning process in neural networks is guided by a loss function, also known as the cost function or error function. This function measures the difference between the network's predictions and the actual target values, quantifying how well the network is performing. Common loss functions include mean squared error for regression tasks and cross-entropy for classification tasks.

The primary technique used to train neural networks is called backpropagation. This process involves two main steps: forward propagation and backward propagation. In forward propagation, input data is passed through the network, generating predictions at the output layer. During backward propagation, the error is calculated using the loss function, and this error is propagated backwards through the network. The gradient of the error with respect to each weight is computed, allowing for weight updates that minimize the error. This process is repeated iteratively, gradually improving the network's performance on the given task.

The power of neural networks lies in their ability to approximate complex functions and learn intricate patterns in data. By passing information through interconnected layers of neurons and using non-linear activation functions to process this information, neural networks can adapt to a wide variety of tasks. The backpropagation algorithm enables these networks to learn from their errors and continuously refine their performance. This versatile architecture has made neural networks powerful tools for a wide range of machine learning applications, from image and speech recognition to natural language processing and beyond.

### 2.2.1. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialized type of neural network designed to process grid-like data, particularly images. CNNs have become the dominant architecture for most computer vision tasks due to their ability to efficiently handle spatial information and learn hierarchical features [11].

The key innovation of CNNs is the use of convolution operations in place of general matrix multiplication in at least one of their layers. This allows the network to process local patterns and maintain spatial relationships within the data. A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply learnable filters to the input, detecting features at different scales. The pooling layers downsample the feature maps, providing a degree of translation invariance. The fully connected layers at the end of the network integrate global information for final decision making [11].

CNNs have several advantages that have contributed to their widespread use. They are relatively efficient in terms of parameter usage due to weight-sharing in convolutional layers. They also exhibit a degree of translation invariance, allowing them to recognize patterns regardless of their position in the image. Furthermore, hierarchical feature learning in CNNs often results in more interpretable intermediate representations compared to fully connected networks [11].

Despite their strengths, CNNs are not without limitations. They can struggle to capture long-range dependencies in images, and their built-in assumptions about the nature of images (such as translation equivariance) may not always be optimal for all tasks. These limitations have motivated ongoing research into alternative architectures, including the development of Vision Transformers [12].

### 2.2.2. Vision Transformers

Vision Transformers (ViTs) are a type of neural network architecture designed to process image data using self-attention mechanisms, originally introduced for natural language processing tasks. Unlike convolutional neural networks (CNNs), which have

been the dominant approach in computer vision, ViTs operate directly on sequences of image patches without using convolutions [13].

The basic architecture of a Vision Transformer, as described by Dosovitskiy et al. in their work "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" [13], consists of the following key components:

1. **Patch Embedding:** The input image is divided into fixed-size patches (e.g., 16x16 pixels), which are then linearly projected into a lower-dimensional embedding space.
2. **Position Embeddings:** Learnable position embeddings are added to the patch embeddings to retain spatial information, as the self-attention operation itself is permutation-invariant.
3. **Transformer Encoder:** A stack of Transformer encoder layers processes the sequence of embedded patches. Each encoder layer contains:
   - **Multi-Head Self-Attention (MHSA):** Allows patches to attend to other patches across the entire image.
   - **Multi-Layer Perceptron (MLP):** Further processes the attention outputs.
   - **Layer Normalization and Residual Connections:** Applied before and after each sub-layer, respectively.
4. **Classification Head:** For image classification tasks, a learnable "CLS" token is prepended to the sequence of patch embeddings, and the final representation of this token is used for classification.

The self-attention mechanism in ViTs enables the model to capture long-range dependencies between different parts of the image without the need for multiple layers of convolutions and pooling operations. This allows ViTs to effectively model the global context and relationships between distant image regions.

One of the key advantages of ViTs is their ability to scale efficiently with increased model and dataset sizes. When pre-trained on large-scale datasets, ViTs have shown excellent performance on various computer vision tasks, often surpassing CNN-based approaches.

However, ViTs also have some limitations. They lack the inductive biases present in CNNs, such as translation equivariance and locality. This means that ViTs may require larger datasets and longer training times to learn these properties from scratch. Additionally, the quadratic complexity of self-attention with respect to the number of patches can be computationally expensive for high-resolution images.

Recent research has focused on addressing these limitations through various architectural modifications, such as introducing hierarchical structures, local attention

mechanisms, and hybrid CNN-Transformer models. These improvements aim to combine the strengths of both CNNs and Transformers for more efficient and effective visual representation learning.

## 2.3. Optical Flow

Optical flow is a fundamental concept in computer vision that describes the apparent motion of objects, surfaces, and edges in a visual scene relative to an observer. It was first formalized by Horn and Schunck in 1981, who defined optical flow as the distribution of apparent velocities of brightness patterns in an image. The basic assumption is that the brightness of a particular point in the pattern remains constant as it moves [14].

Traditionally, optical flow estimation has been approached as an optimization problem, often using variational methods [15]. However, recent advancements in deep learning have led to the development of convolutional neural network (CNN) based approaches for optical flow estimation. Fischer et al. introduced FlowNet in 2015, demonstrating that CNNs could be trained end-to-end to predict optical flow directly from image pairs [16]. This was further improved by Ilg et al. with FlowNet 2.0, which stacked multiple FlowNet modules for better accuracy [17].

More recent architectures, such as RAFT (Recurrent All-Pairs Field Transforms) by Teed and Deng, have further pushed the state of the art by using iterative refinement and maintaining a single high-resolution flow field [15]. These deep learning approaches have shown competitive accuracy while achieving real-time or near-real-time performance on standard benchmarks such as Sintel and KITTI [15].

Despite these advances, challenges remain in optical flow estimation, particularly in handling occlusions, large displacements, and small fast-moving objects [15]. Ongoing research continues to address these issues and improve the accuracy and efficiency of optical flow algorithms for various applications in computer vision and robotics.

## 2.4. VitalCamSet Dataset

The VitalCamSet [18] dataset is a comprehensive collection of video recordings and physiological measurements designed to evaluate Photoplethysmography Imaging (PPGI) algorithms. It includes synchronized recordings of various vital signs, including electrocardiography (ECG), photoplethysmography (PPG), oxygen saturation (SpO2), and notably respiration signals from the abdominal and thoracic regions.

The dataset comprises recordings from 26 subjects (20 men, 6 women) aged 23 to 33 years. Each subject participated in 10 different scenarios, each scenario lasting two minutes. This resulted in a total of 1040 minutes of video footage, captured simultaneously by both an RGB color camera and a monochrome near-infrared camera [18].

All data, including video recordings, vital sign measurements, and contextual parameters such as brightness and head movement, are stored in the unisens data format. This format ensures easy access and integration of the dataset for researchers [18].

The scenarios in VitalCamSet were carefully designed to address various challenges in PPGI. They include different lighting conditions (natural, artificial, abrupt changes, and slow changes), as well as specific illumination scenarios using green and infrared LED lighting. Additionally, the dataset incorporates motion scenarios to assess the impact of head movements (rotatory, scaling, and translatory) and a text writing scenario to simulate realistic, random movements. This diverse set of scenarios allows researchers to evaluate PPGI algorithms under various challenging conditions, making VitalCamSet a valuable resource to advance the field of remote vital sign monitoring [18].

## 2.5. Pearson's Correlation Coefficient

The Pearson correlation coefficient is a fundamental statistical measure used to assess the strength and direction of the linear relationship between two continuous variables [19]. It is widely employed in scientific research across various disciplines to quantify the degree of association between variables of interest.

The Pearson correlation, denoted as $r$, is calculated using the following formula:

$$r = \frac{\mathrm{Cov}(x, y)}{s_x \cdot s_y}$$

Where $\mathrm{Cov}(x, y)$ represents the covariance between variables $x$ and $y$, and $s_x$ and $s_y$ are the sample standard deviations of $x$ and $y$, respectively [20].

The correlation coefficient ranges from $-1$ to $+1$, with the sign indicating the direction of the relationship and the magnitude reflecting its strength. A positive correlation ($r > 0$) suggests that as one variable increases, the other tends to increase as well. In contrast, a negative correlation ($r < 0$) implies that as one variable increases, the other tends to decrease. A correlation of 0 indicates no linear relationship between the variables [20].

It is important to note that the Pearson correlation is a measure of linear association and may not capture non-linear relationships between variables. Furthermore,

correlation does not imply causation and researchers should be cautious when interpreting correlation results in the context of their studies.

The significance of the correlation coefficient can be evaluated using statistical tests, such as Fisher's $z$ transformation, which is particularly useful for normally distributed variables. For non-parametric relationships or when the assumption of normality is violated, alternative measures such as Spearman's rho or Kendall's tau may be more appropriate [20].

In scientific research, the Pearson correlation is often used in exploratory data analysis (EDA) to identify potential relationships between variables and guide further investigation. It can be complemented by visual representations, such as scatter plots, to provide a more comprehensive understanding of the data's structure and patterns.

The statistical significance of a Pearson's correlation coefficient can be assessed using a p-value. The p-value represents the probability of obtaining a correlation coefficient as extreme as the observed one, assuming that there is no true correlation between the variables in the population (i.e. the null hypothesis is true) [20]. A small p-value (typically less than a predetermined significance level, such as 0.05) suggests strong evidence against the null hypothesis, indicating that the observed correlation is likely not due to chance. When choosing a significance level, researchers must balance the risk of Type I errors (falsely rejecting the null hypothesis) with Type II errors (failing to detect a true correlation). Common choices for significance levels include 0.05, 0.01, and 0.001, with 0.05 widely used in many fields [20]. However, it is important to note that while a statistically significant correlation indicates a likely relationship between variables, it does not necessarily imply a practically meaningful or causal relationship. Researchers should consider both the p-value and the magnitude of the correlation coefficient when interpreting the results.

By quantifying the degree of linear association between variables, the Pearson correlation coefficient serves as a valuable tool in various scientific disciplines, enabling researchers to uncover relationships, formulate hypotheses, and design more targeted experiments or analyses.

## 2.6. T-Test

The Independent (Two-Sample) T-Test is a statistical method used to compare the means of two independent groups to determine if there is a significant difference between them [20]. This test is widely used in various fields of research, including medicine, psychology, and social sciences.

The t-value represents the difference between the two means of the group in relation to the variation in the data. It is calculated by dividing the difference between the

two means by the standard error of the difference. A higher absolute t-value suggests a greater difference between the two groups.

- Positive t-value: Indicates that the mean of the first sample is larger than the mean of the second sample (or larger than the population mean in a one-sample T-Test).
- Negative t-value: Indicates that the mean of the first sample is smaller than the mean of the second sample (or smaller than the population mean).

The p-value, also known as the probability value, is crucial in hypothesis testing. It represents the probability of observing data as extreme as or more extreme than what was actually observed, assuming the null hypothesis is true. In the context of a t-test:

- A small p value (typically $\leq 0.05$) suggests strong evidence against the null hypothesis, indicating a statistically significant difference between the two group means.
- A large p value ($> 0.05$) suggests weak evidence against the null hypothesis, indicating that the difference between the two group means is not statistically significant.

It is important to note that, while statistical significance is crucial, it should not be the sole factor in interpreting results. The practical significance of the findings should also be considered in the context of the research question and field of study.

## 2.7. Fourier Analysis

Fourier analysis is a fundamental mathematical technique that allows complex signals to be decomposed into simpler sinusoidal components [21]. At its core, the Fourier transform converts a signal from the time or space domain to the frequency domain, revealing the various frequency components that make up the signal [22].

The basic principle of Fourier analysis is that any periodic function can be represented as a sum of sine and cosine waves of different frequencies [22]. For non-periodic functions, the Fourier transform extends this idea to an integral over a continuous spectrum of frequencies. The Fourier transform pair defines the relationship between a function g(t) and its frequency domain representation G(f):

$$\hat{G}(f) = \int_{-\infty}^{\infty} g(t) e^{-2\pi i f t} dt$$

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{2\pi i f t} df$$

These equations allow us to move between the spatial/time domain and the frequency domain representations of a signal [22].

Fourier analysis has numerous applications in science and engineering. In signal processing, it enables the filtering, compression, and analysis of complex waveforms. In optics, it provides a powerful framework for understanding diffraction and imaging systems [22]. The Fast Fourier Transform (FFT) algorithm, developed by Cooley and Tukey in 1965 [22], revolutionized the field by providing an efficient computational method to perform Fourier transforms on discrete data.

By revealing the frequency content of signals, Fourier analysis allows to gain deep insights into the nature of physical phenomena and to develop sophisticated processing techniques for a wide range of applications [22].

## 2.8. Fast Fourier Transform

The Fast Fourier Transform (FFT) is a fundamental algorithm in signal processing and computational science that efficiently computes the Discrete Fourier Transform (DFT) of a sequence or a signal. The FFT is essential for analyzing the frequency content of signals, performing convolutions, and solving various problems in fields such as image processing, audio analysis, and data compression.

The DFT of a sequence of $N$ complex numbers $x[n]$ is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i2\pi kn/N} \tag{2.1}$$

where $k = 0, 1, \ldots, N-1$.

Naively computing the DFT requires $O(N^2)$ operations, making it computationally expensive for large datasets. The FFT algorithm, introduced by Cooley and Tukey in 1965, reduces this complexity to $O(N \log N)$ operations [22]. This significant reduction in computational complexity has made the FFT a cornerstone of modern digital signal processing.

The key insight of the FFT algorithm is to exploit the symmetry and periodicity of complex exponential factors (twiddle factors) in the DFT computation. By recursively dividing the input sequence into smaller subsequences, the FFT algorithm reuses intermediate results, eliminating redundant calculations.

One of the most common FFT implementations is the radix-2 Cooley-Tukey algorithm, which works efficiently for input sizes that are powers of two. However, other variants exist for arbitrary input sizes, such as the split-radix FFT and the prime-factor algorithm.

The FFT has several important properties that make it particularly useful in various applications:

1. Linearity: The FFT of a linear combination of signals is the linear combination of their individual FFTs.
2. Circular convolution: The FFT can efficiently compute circular convolutions, which are essential in many signal processing applications.
3. Parseval's theorem: The energy of a signal in the time domain is equal to its energy in the frequency domain, allowing for energy-based analyses.
4. Spectral leakage and windowing: The FFT assumes periodic signals, which can lead to spectral leakage for non-periodic signals. Windowing techniques are often used to mitigate this effect.

In practice, the FFT is implemented in many software libraries and hardware systems. For example, the FFTW (Fastest Fourier Transform in the West) library is a widely used, highly optimized software implementation of the FFT [22].

The FFT has numerous applications across various scientific and engineering disciplines. In optics and imaging, it is used for analyzing diffraction patterns, image filtering, and phase retrieval in holography [22]. In audio processing, FFT is fundamental for spectral analysis, noise reduction, and audio compression algorithms. In telecommunications, it plays a crucial role in implementing efficient modulation schemes like OFDM (Orthogonal Frequency Division Multiplexing).

## 2.9. Butterworth Filters

Butterworth filters are a fundamental type of digital filter utilized in signal processing to attenuate unwanted frequency components of a signal. These filters are renowned for their maximally flat frequency response in the passband, ensuring no ripples within this frequency range, and a smooth transition to the stopband [23].

Butterworth filters are characterized by a smooth and monotonically decreasing frequency response. The absence of ripples in the passband is a hallmark of these filters' design, which strives for a maximally flat magnitude response. This characteristic is particularly beneficial in applications where phase linearity and minimal distortion are critical [23].

The roll-off rate of Butterworth filters is governed by the filter order. Higher order corresponds to a steeper roll-off from the passband to the stopband. However, this also results in an increased phase delay, which can be a determining factor in the filter design depending on the application's requirements [23].

Butterworth filters can be categorized based on their frequency-selective properties:

- *Lowpass Filters*: Permit frequencies below the cutoff frequency to pass, while higher frequencies are attenuated.
- *Highpass Filters*: Allow frequencies above the cutoff frequency to pass, attenuating lower frequencies.
- *Bandpass Filters*: Allow a specific range of frequencies to pass through and attenuate frequencies outside this range.
- *Bandstop Filters*: Attenuate a narrow range of frequencies while allowing others to pass [23].

The filtering process in Butterworth filters involves transforming the signal into the frequency domain, applying the filter's frequency response, and attenuating frequencies in the stopband while maintaining those within the passband. Finally, the filtered signal is converted back into the time domain [23].

In practice, Butterworth filters are often implemented in digital signal processing as either Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters. IIR filters are typically preferred due to their computational efficiency [23].

Designing a Butterworth filter involves balancing trade-offs between filter order, cutoff sharpness, and phase response. Higher-order filters offer sharper cutoffs, but at the expense of increased phase delay, impacting transient response and possibly introducing artifacts [23].

## 2.10. Frequency Extraction from Vital Signs

Extracting frequency information from vital signs such as photoplethysmography (PPG) and respiration signals is crucial to estimating physiological parameters such as heart rate and respiratory rate. Several methods can be used to extract these frequencies, each with its own advantages and considerations. This section discusses four prominent techniques: Power Spectral Density (PSD), Peak Count (PC), Crossing Point (CP), and Negative Feedback Crossing Point (NFCP).

### 2.10.1. Power Spectral Density

Power Spectral Density (PSD) is a frequency-domain technique that estimates how the power of a signal is distributed across different frequencies. For vital signs, PSD can reveal the dominant frequencies corresponding to physiological processes:

1. The signal is first transformed from the time domain to the frequency domain, typically using methods like Fast Fourier Transform (FFT) or autoregressive (AR) modeling [24].

2. The resulting spectrum shows peaks at frequencies with high signal power. For respiratory rate estimation, the frequency of the highest peak within the expected respiratory frequency range (e.g., 0.1-0.5 Hz) is often identified as the respiratory rate [25].

3. PSD can be applied to raw vital sign signals or to derived respiratory signals extracted from ECG or PPG, such as respiratory-induced intensity variation (RIIV), amplitude variation (RIAV), or frequency variation (RIFV) [24].

The frequency spectrum of the preprocessed signal, as determined by PSD, can be visualized to identify the dominant frequencies, as shown in Figure 2.1.

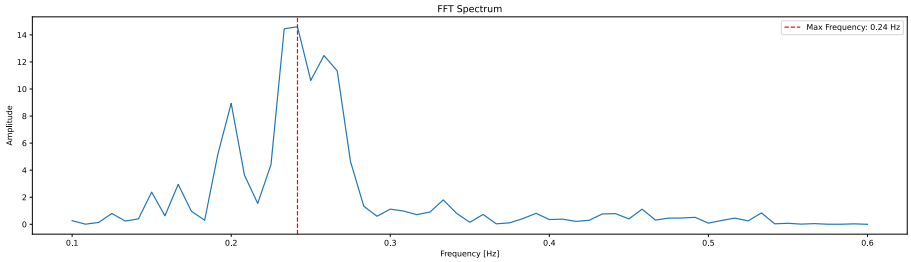


Figure 2.1.: Frequency spectrum of the preprocessed signal.

### 2.10.2. Peak Count

Peak Count (PC) is a time-domain method that estimates frequency by counting notable features in the signal over a given time window:

1. The signal is first preprocessed to remove noise and emphasize the oscillations of interest, often using bandpass filtering [25].

2. Peaks (local maxima) in the filtered signal are identified.

3. The frequency is estimated by counting the number of peaks within a specified time window and dividing by the window duration [26].

4. To improve robustness, additional criteria may be applied, such as setting amplitude thresholds for peak detection or requiring a minimum time between successive peaks [25].

The peaks counted from the preprocessed signal are visualized in Figure 2.2.

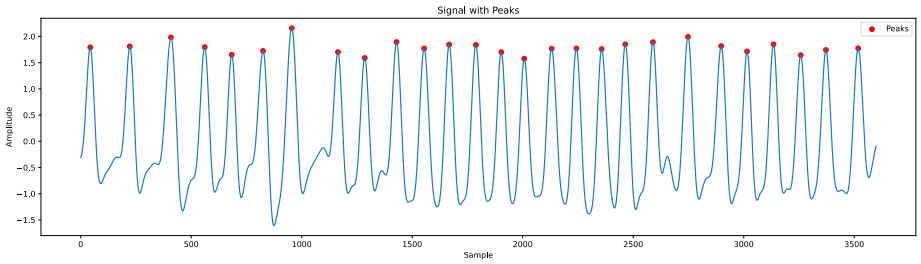


Figure 2.2.: Counted peaks of the preprocessed signal.

### 2.10.3. Crossing Point

The Crossing Point (CP) method estimates the respiratory frequency by analyzing when the respiratory signal crosses a reference level. This technique is particularly useful for the analysis of respiratory waveforms in the time domain. The process involves the following steps:

1. **Signal preprocessing:** The respiratory signal is typically preprocessed to remove noise and emphasize respiratory oscillations, often using bandpass filtering [27].
2. **Reference level definition:** A reference level is defined, commonly as the signal mean or zero after detrending the signal.
3. **Signal shifting:** The original respiratory waveform is shifted to the right by a fixed number of points (w), creating two waveforms - the original and the shifted version [27].
4. **Crossover point detection:** The algorithm identifies the points where the original signal intersects with the shifted signal. These intersections are the crossover points.
5. **Respiratory cycle identification:** Ideally, crossover points appear around the peaks and troughs of the respiration signal. Each crossover point represents a transition from exhalation to inhalation or vice versa. Two consecutive crossover points typically represent a complete respiratory cycle [27].
6. **Respiratory rate calculation:** The respiratory rate (RR) is calculated using the following equation:

$$RR = \frac{C_w/2}{N/f_s} * 60 \tag{2.2}$$

where $C_w$ is the number of crossover points, N is the total number of data points, and $f_s$ is the sampling frequency of the respiratory signal [27].

Figure 2.3 shows the crossover points and the corresponding curves of the preprocessed signal.

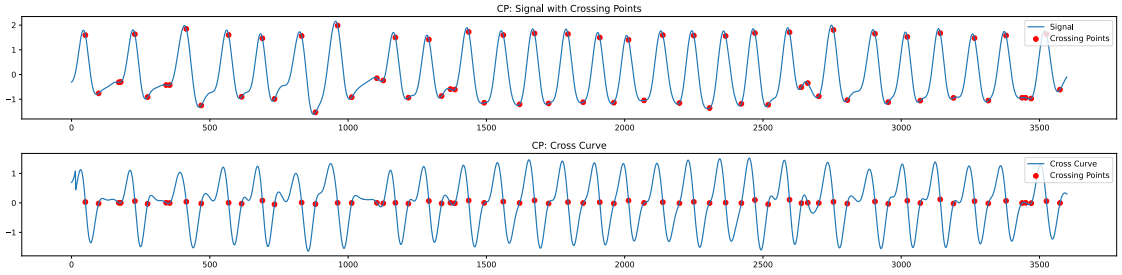


Figure 2.3.: Crossing points and the cross-curve signal of the preprocessed signal.

### 2.10.4. Negative Feedback Crossing Point

The Negative Feedback Crossing Point (NFCP) method is an advanced version of the CP method that aims to improve accuracy by adaptively adjusting to different waveform shapes and reducing errors from spurious crossings. This method incorporates a feedback mechanism to refine the detection of valid crossings. The NFCP method involves the following steps:

1. **Initial processing:** The signal is preprocessed and a reference level is defined as in the CP method.
2. **Crossover point detection:** Initial crossover points are identified between the original and shifted waveforms.
3. **Adaptive threshold calculation:** An adaptive threshold is calculated based on the intrinsic respiration rate to determine the minimum acceptable interval between breaths:

$$N_{span} = \frac{60}{RR} * f_s \tag{2.3}$$

$$N_{minspan} = \frac{N_{span} * q}{2} \tag{2.4}$$

where $N_{span}$ is the average number of data points per breath, RR is the initial respiratory rate estimate, $f_s$ is the sampling frequency, $N_{minspan}$ is the minimum breath data interval, and q is an adjustable parameter (typically set to 0.6) [28].

4. **Crossover point validation:** The algorithm compares the intervals between consecutive crossover points with $N_{minspan}$. If an interval is smaller than $N_{minspan}$, the corresponding crossover points are deemed too close and are removed.
5. **Iterative refinement:** Steps 3-4 are repeated, updating the respiratory rate estimate each time, until all intervals between crossover points are larger than $N_{minspan}$.
6. **Final respiratory rate calculation:** The final respiratory rate is calculated using the validated crossover points, similar to the CP method.

The precision of this method in capturing valid crossing points is illustrated in Figure 2.4.

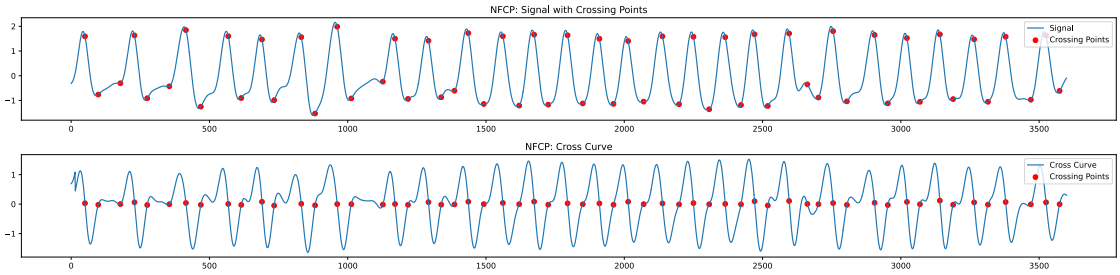


Figure 2.4.: NFCP points and the cross-curve signal of the preprocessed signal.

Each of these methods has its strengths and limitations. PSD provides a comprehensive view of the frequency content but may struggle with non-stationary signals. Time-domain methods like PC, CP, and NFCP can be more robust to non-stationarity but may be more sensitive to noise and artifacts. In practice, combining multiple methods or using ensemble approaches can lead to a more reliable frequency estimation of vital signs [24].

# 3. Literature Review

This chapter explores existing methodologies and advances in the extraction of respiratory signals from video data. It is structured to discuss two primary approaches: the use of optical flow to capture chest movements and remote photoplethysmography (rPPG) to derive respiration rates from PPG signals. It reviews various models and techniques, including optical flow algorithms like Lucas-Kanade and FlowNet2, as well as rPPG models like DeepPhys and EfficientPhys. In addition, the chapter identifies research gaps, emphasizing the potential for transformer-based optical flow techniques and end-to-end transformer models for respiration extraction and highlighting areas for future exploration in the field.

## 3.1. Respiration Extraction

This section will explain how respiration signals can be extracted from videos. Generally speaking, there are two ways to extract the respiration signal. The first is to observe the movement of the chest that is associated with breathing. The second is to calculate the respiration rate from the PPG signal. Both are explained in more detail below.

### 3.1.1. Optical Flow

Extracting respiration using optical flow is a non-contact method that relies on capturing the subtle motions of the chest and abdomen associated with breathing. The process begins by selecting a region of interest (ROI) in the upper body of the subject where respiratory movements are most prominent [2]. Within this ROI, optical flow algorithms are applied to track the motion between consecutive video frames [29].

The optical flow computation generates motion vectors for each tracked point or pixel, representing their displacement over time. As respiration manifests itself primarily as vertical movement of the chest and abdomen, the vertical components of these motion vectors are of particular interest [30]. These components of vertical motion are aggregated throughout ROI, typically by averaging or summing, to produce a respiratory signal that varies over time [27].

Filtering techniques are used to isolate the respiratory frequency band, typically in the range of 0.1 to 0.5 Hz (corresponding to normal breathing rates of approximately 6 to 30 breaths per minute) [31]. After filtering, the processed signal is analyzed to determine the respiration rate. This analysis can be performed using various methods, including Peak Counting (PC), Power Spectral Density (PSD), Crossing Point (CP), or Negative Feedback Crossing Point (NFCP).

### 3.1.2. Remote Photoplethysmography

Remote Photoplethysmography (rPPG) is a non-invasive optical technique used to detect blood volume changes in the microvascular bed of tissue. In videos, PPG can best be monitored on the face, particularly in regions such as the forehead, cheeks, and around the eyes [32]. PPG works on the principle of light absorption by hemoglobin in the blood [6]. When light is emitted onto the skin, a fraction is absorbed by the blood, while the residual light is detected by a photosensor. Remote photoplethysmography (rPPG) and imaging photoplethysmography (iPPG) are contactless variants of PPG that use cameras to extract physiological signals like heart rate from video recordings of a person's skin.

Respiration modulates the PPG signal through several mechanisms, allowing the calculation of the respiration rate from the PPG waveform [33]:

1. Respiratory-induced frequency variation (RIFV): Changes in heart rate that are synchronized with the respiratory cycle, known as respiratory sinus arrhythmia.

2. Respiratory-induced intensity variation (RIIV): Changes in the baseline PPG signal due to variations in intrathoracic pressure during the respiratory cycle.

3. Respiratory-induced amplitude variation (RIAV): Changes in pulse amplitude caused by decreased cardiac output during inspiration.

The connection between respiration and the circulatory system is evident in these modulations. Respiration affects intrathoracic pressure, which in turn influences venous return and cardiac output. This respiratory-circulatory interaction is reflected in the PPG signal, which allows estimation of the respiratory rate from PPG recordings [32].

The extraction is similar to the extraction from motion signals. The PPG signal is filtered to isolate the respiratory signal. A bandpass filter is commonly used to do this in the frequency range from 0.1 to 0.5 Hz [31]. Following this, the processed signal is analyzed to determine the respiration rate. This analysis can be conducted using methods such as Peak Counting (PC), Power Spectral Density (PSD), Crossing Point (CP), or Negative Feedback Crossing Point (NFCP).

## 3.2. Optical Flow Models

This section provides an overview of different methodologies for using optical flow to detect breathing, detailed are specific models such as Pixel Intensity, Lucas-Kanade Algorithm, FlowNet and FlowNet2, and Recurrent All-Pairs Field Transforms (RAFT).

### 3.2.1. Pixel Intensity

The simplest approach is to use the pixel intensity in the region of interest (ROI). Lucy and Suha effectively used this method in their 2021 study [34]. The video frames are processed to extract the pixel intensity information from the ROI. This typically involves decomposing each frame into separate color channels (red, green, and blue), calculating the average intensity values for each color channel within the ROI for every frame, and combining the color channel information to create a single intensity signal.

### 3.2.2. Lucas-Kanade Algorithm

The Lucas-Kanade algorithm [35], introduced by Bruce D. Lucas and Takeo Kanade in 1981, is a well-known method used in computer vision for image registration, particularly in determining optical flow. It is designed to track the motion between two image frames taken at different times or from different viewpoints by aligning them. This algorithm operates under the assumption that the displacement of image content between frames is small.

A key aspect of the Lucas-Kanade algorithm is its reliance on tracking feature points. Feature points are typically corner points with strong gradients in multiple directions. These points are chosen because they are easier to localize and track across frames compared to flat or smooth regions. The strong gradients at these points provide a clear, distinctive pattern that can be consistently matched in successive images. As illustrated in Figure 3.1, feature points are identified in the chest area for applications such as respiration monitoring. This allows for precise tracking of movements that might indicate breathing patterns.

The algorithm focuses on finding the disparity between two frames. Given functions $F(x)$ and $G(x)$, which represent the intensity values of two consecutive image frames at each pixel location $x$, the goal is to identify a vector $h$ that minimizes the difference between $F(x + h)$ and $G(x)$.

The Lucas-Kanade method uses spatial intensity information (i.e., gradient) to compute the displacement. By examining the intensity changes over a small region of interest, a differential technique akin to the Newton-Raphson method is applied.

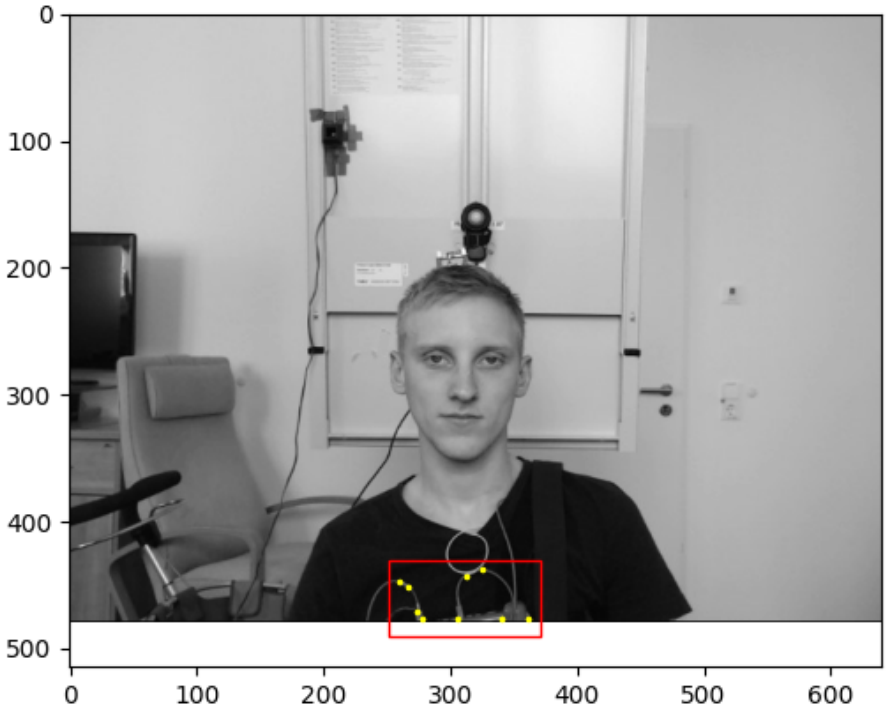


Figure 3.1.: Feature points of the Lucas-Kanade algorithm in the chest area

Starting with an initial estimate of the disparity, the method refines this estimate by iteratively adjusting it based on calculated differences and gradients. This iterative process aims to converge towards the correct disparity efficiently.

Using gradients, the algorithm performs fewer comparisons than exhaustive search methods, which significantly reduces computational cost, especially as it operates with $O(M^2 logN)$ complexity under typical circumstances. This approach can be extended to handle various transformations beyond simple translations, including rotations and scalings.

The success of the Lucas-Kanade algorithm largely depends on the assumption of small movements and relies on the availability of a good initial estimate, which can be obtained using hierarchical or multi-resolution approaches. This method has become a foundational technique in image registration due to its robustness and computational efficiency.

This method has been used for respiration extraction by many different approaches [36, 28, 26, 37, 38, 29] for respiration extraction.

### 3.2.3. FlowNet and FlowNet2

FlowNet and FlowNet2 are notable approaches in applying deep learning to optical flow estimation. These architectures explore the use of convolutional neural networks to predict optical flow directly from pairs of input images. The authors aimed to achieve comparable accuracy to traditional methods while potentially offering faster computation times.

The original FlowNet [16] introduced two main architectures:

1. FlowNetSimple (FlowNetS): A straightforward encoder-decoder architecture that takes two stacked images as input and directly outputs the flow prediction.
2. FlowNetCorrelation (FlowNetC): Uses two separate processing streams for the two images and includes an explicit correlation layer to match features between the two frames before estimating the flow.

Both architectures use an encoder with strided convolutions to compress the spatial resolution, followed by a decoder with upconvolutions to produce the full-resolution flow estimate. They also employ skip connections from encoder to decoder layers to preserve fine details.

FlowNet was trained on a synthetic dataset of rendered chair images called Flying Chairs. Although it achieved promising results, its accuracy still lagged behind state-of-the-art traditional methods, especially for small displacements and real-world data.

FlowNet2 [17] built upon the original FlowNet through several key modifications:

1. Stacked network architecture: Multiple FlowNet networks are stacked sequentially, with each subsequent network refining the previous flow estimate. This allows for iterative improvement of the flow field.
2. Warping layer: Introduces warping of the second image based on the current flow estimate, allowing each network in the stack to focus on estimating the residual flow.
3. Small displacement network: A specialized FlowNetSD network is introduced to handle small motions, which are then fused with the output of the stacked networks.
4. Improved training schedule: A carefully designed curriculum of datasets and learning rates leads to better generalization.

The complete FlowNet2 architecture consists of the following components:

- FlowNet2-C: Initial flow estimation using FlowNetC
- FlowNet2-CS: Refinement using stacked FlowNetS

- FlowNet2-CSS: Further refinement with another stacked FlowNetS
- FlowNet2-SD: Specialized small displacement network
- Fusion network: Learns to fuse FlowNet2-CSS and FlowNet2-SD outputs

This stacked and fused architecture allows FlowNet2 to achieve higher accuracy than the original FlowNet, aiming to match or exceed the performance of state-of-the-art traditional methods while maintaining a speed advantage.

Additionally, FlowNet2 introduced variants with different speed-accuracy trade-offs:

- FlowNet2-s: Single network, fastest but least accurate
- FlowNet2-ss: Two stacked networks
- FlowNet2-css: Three stacked networks with lower capacity
- FlowNet2-CSS: Full three stacked networks

These variants allow users to choose the appropriate model based on their specific requirements for speed and accuracy.

Through these architectural modifications and improved training techniques, FlowNet2 aimed to demonstrate that deep learning approaches could achieve competitive results on optical flow estimation while running at interactive framerates. This research explored new possibilities for using neural network-based optical flow in real-time applications.

The FlowNet2 model was used by Guo et al. [39] to extract breathing motions from videos.

### 3.2.4. Recurrent All-Pairs Field Transforms

RAFT (Recurrent All-Pairs Field Transforms) represents a state-of-the-art deep learning algorithm for optical flow estimation. The method begins with a feature extraction step, utilizing a feature encoder to extract per-pixel features from both input images, creating rich, multi-level representations that can be used for matching. One of RAFT's key innovations lies in the construction of a 4D correlation volume, which is created by computing the inner product between all pairs of feature vectors from the two images. This comparison of features in pairs enables the algorithm to consider the global context when estimating the flow [15].

The 4D correlation volume is subsequently pooled at multiple scales to create a set of multi-scale volumes, facilitating the handling of large displacements and providing a hierarchical representation of motion. RAFT employs a recurrent update operator to iteratively refine the flow estimates. Starting from an initial flow field of zero,

the algorithm performs multiple update steps. In each iteration, the current flow estimate is used to look up the values of the correlation volumes, which, along with other contextual features, are fed into a GRU-based update operator to produce an update to the flow field [15].

The recurrent architecture allows RAFT to refine its estimates over multiple iterations, mimicking the behavior of traditional optimization-based approaches. Instead of explicitly defining an optimization objective, RAFT learns to perform updates that converge to accurate flow estimates, combining the benefits of learned features with the iterative refinement characteristic of classical methods. Despite the computational complexity implied by the all-pairs correlation, RAFT includes an efficient implementation that scales linearly with the number of pixels and iterations, making it practical for real-world applications [15].

RAFT demonstrates several key advantages, including state-of-the-art accuracy on benchmark datasets such as KITTI and Sintel, strong generalization to novel scenes even when trained only on synthetic data, and high efficiency in terms of inference time, training speed, and parameter count. These characteristics position RAFT as a significant advancement over previous optical flow methods, effectively combining the strengths of deep learning with principles inspired by classical optimization-based approaches. The algorithm's ability to perform all-pair matching and iterative refinement enables it to produce highly accurate flow estimates in a wide range of scenarios, marking a substantial step forward in the field of optical flow estimation [15].

RAFT models have not been used so far to extract respiration signals from videos.

## 3.3. rPPG Models

This section will describe various rPPG models and briefly explain their architecture. It should be noted that many of these models can be adopted to extract respiration instead of PPG.

### 3.3.1. DeepPhys

The DeepPhys model [40], introduced by Chen and McDuff, is an end-to-end convolutional attention network (CAN) designed for non-contact video-based physiological measurement. This approach aims to recover the heart rate (HR) and the breathing rate (BR) from RGB or infrared videos.

The model architecture consists of two main components: a motion model and an appearance model. The motion model uses a normalized frame difference as input, based on a skin reflection model. This representation is intended to capture

physiological motions while accounting for varying lighting conditions. The appearance model processes raw video frames to inform the motion estimation through an attention mechanism.

DeepPhys employs a VGG-style CNN architecture with some modifications. It uses average pooling instead of max pooling and utilizes hyperbolic tangent (tanh) activation functions. To address potential overfitting, the network incorporates dropout layers and employs ensemble learning over training checkpoints.

A key feature of DeepPhys is its attention mechanism, which allows the model to learn soft-attention masks. This approach aims to perform automatic region of interest (ROI) selection without explicit skin segmentation. The attention mechanism is derived from appearance information to guide motion estimation.

### 3.3.2. EfficientPhys

EfficientPhys [41] is a model designed for camera-based cardiac measurement that processes raw video frames without extensive pre-processing. The model consists of two main variants: convolution-based and transformer-based.

The core of EfficientPhys includes a normalization module with a difference layer and a batch normalization layer. The difference layer computes the first forward difference along the temporal axis of raw video frames, while the batch normalization layer adjusts the scale of these differences.

The convolution-based EfficientPhys uses a self-attention-shifted network (SASN) that combines tensor-shifted convolutional operations with self-attention mechanisms. This structure is designed to model spatial-temporal information in the video frames.

The transformer-based EfficientPhys adapts the Swin transformer architecture, incorporating a tensor-shift module (TSM) before each Swin transformer block. This design aims to facilitate information exchange across the temporal axis of the video.

Both variants of EfficientPhys have been evaluated on several benchmark datasets, with the convolution-based model showing lower computational requirements. The model's design allows for on-device applications, which may have implications for accessibility in various settings.

### 3.3.3. MTTS-CAN / TS-CAN

The extraction of respiration signals using MTTS-CAN (Multi-Task Temporal Shift Convolutional Attention Network) represents a significant advancement in

non-contact physiological measurement. MTTS-CAN employs a sophisticated two-branch architecture incorporating a spatial attention module, as described by Liu et al. [42]. This structure consists of an appearance branch that processes a single averaged frame and a motion branch that utilizes normalized frames derived from adjacent frames in the video sequence.

A key innovation in MTTS-CAN is the implementation of a Temporal Shift Module (TSM) within the motion branch. This module effectively mimics the functionality of 3D convolutions, enabling information exchange among neighboring frames without increasing the parameter count of the network. The architecture also incorporates an attention module that serves as a bridge between the appearance and motion branches. This module generates soft-attention masks, focusing the network's processing on pixels that exhibit stronger physiological signals, thereby improving the accuracy of respiration signal extraction [42].

One of the distinguishing features of MTTS-CAN is its multitask learning approach. The network simultaneously estimates both pulse and respiration signals, leveraging shared intermediate representations between these related physiological processes [42]. This approach not only improves computational efficiency, but also potentially enhances the accuracy of both measurements through mutual information sharing.

### 3.3.4. BigSmall

The BigSmall model employs a novel dual-branch architecture to extract respiratory signals from video [43]. This approach leverages spatial and temporal information to effectively capture the complex nature of respiratory signals in video data. The model's Small branch processes low-resolution (9x9 pixel) normalized difference frames using convolutional layers enhanced with Wrapping Temporal Shift Modules (WTSM). This branch is specifically designed to model the temporal dynamics of high-frequency, low-resolution signals such as respiration. Complementing this, the Big branch processes high-resolution (144x144 pixel) raw frames through convolutional layers, capturing crucial spatial features that provide context for the respiratory signal.

The outputs from these two branches are then fused, allowing the model to integrate spatial and temporal information for a comprehensive analysis of the respiratory signal. This fused representation is subsequently processed through fully connected layers to predict the respiratory waveform. The model's output undergoes post-processing, including filtering with a second-order Butterworth filter (cutoff frequencies of 0.08-0.5 Hz) and peak detection on the Fourier spectrum to derive the breathing rate.

The key innovation of the BigSmall model lies in its efficient dual-branch architecture, which simultaneously captures the spatial and temporal aspects of the respiratory signal. Using the Small branch for temporal dynamics and the Big branch for spatial

context, the model can effectively extract respiration information from video frames. This approach represents a significant advancement in video-based physiological signal extraction, offering a more comprehensive and efficient method for respiratory signal analysis [43].

### 3.3.5. RhythmFormer

The RhythmFormer [44] is a novel approach for extracting remote photoplethysmography (rPPG) signals based on a Hierarchical Temporal Periodic Transformer. This method explicitly leverages the quasi-periodic nature of rPPG signals to design a periodic sparse attention mechanism, which finely extracts periodic rPPG features across multiple temporal scales. The core of the RhythmFormer is the Hierarchical Temporal Periodic Transformer, consisting of three stages, each containing a Temporal Periodic Transformer (TPT) Block with a sampling coefficient of n. This structure allows for the reinforcement of inter-frame rPPG features at smaller time scales before modeling periodic features at larger time scales, effectively reducing noise interference.

A key innovation in the RhythmFormer is the fusion stem module, which integrates the difference frame with the raw frame. This integration enables frame-level awareness of Blood Volume Pulse (BVP) wave variations, enhancing rPPG features, and guiding the transformer to focus on rPPG signals. The fusion stem can be easily transferred to existing methods, significantly improving their performance.

The RhythmFormer employs a Hybrid Loss function to address the challenge of the potential mixture of information and noise in the ground truth BVP. This loss function combines constraints from both the temporal and frequency domains, with an additional HR distance characterization. Specifically, the overall loss is expressed as:

$$Loss_{overall} = \alpha * Loss_{Time} + \beta * Loss_{Freq} + \gamma * Loss_{HR}$$

where $Loss_{Time}$ is computed using the negative Pearson correlation coefficient, $Loss_{Freq}$ is calculated using the cross-entropy between the predicted BVP wave's frequency domain and the heart rate derived from the actual BVP wave, and $Loss_{HR}$ represents the Kullback-Leibler divergence between the ground truth HR distribution and the predicted HR distribution.

The RhythmFormer has demonstrated state-of-the-art performance in comprehensive experiments, surpassing previous approaches across various datasets with fewer parameters and reduced computational complexity. This makes it a promising new baseline for fully supervised rPPG tasks and opens up possibilities for applications in complex real-world environments.

## 3.4. Taxonomy of Extraction Methods

In the domain of non-contact respiration monitoring, respiration signals can be extracted from video data using a variety of methods. Each method takes advantage of distinct techniques and focuses on different physiological indicators related to breathing. The three primary approaches for extracting respiration signals are as follows:

- **Optical Flow:** This method involves observing the breathing motion, particularly in the chest area, to detect subtle displacements that occur during respiration. Techniques based on optical flow rely on tracking these physical movements to infer the breathing rate.
- **Remote Photoplethysmography (rPPG):** By calculating respiration from the Photoplethysmography (PPG) signal, this method derives respiratory information from variations in skin color correlated with blood volume changes. These fluctuations in the PPG signal are affected by respiratory-induced changes in blood flow.
- **Direct Extraction via Neuronal Networks:** Leveraging sophisticated neural network models, this method extracts respiration signals directly from video data. The exact mechanisms by which these networks operate remain somewhat ambiguous; they may focus on chest-wall motion, detect changes related to the PPG signal, or both. The complexity and heightened learning capacity of neuronal networks enable them to autonomously ascertain respiration-related information without explicit instructions on which modality to prioritize.

The following table provides a classification of various models and their respective methods for respiration signal extraction, including whether they utilize transformer architectures:

| **Model** | **Method** | **Transformer** |
|---|---|---|
| Pixel Intensity | Optical Flow | No |
| Lucas-Kanade | Optical Flow | No |
| Flownet | Optical Flow | No |
| BigSmall | Directly | No |
| MTTS-CAN | Directly | No |
| TS-CAN | rPPG | No |
| DeepPhys | rPPG | No |
| EfficientPhys | rPPG | No |
| RhythmFormer | rPPG | Yes |

Table 3.1.: Taxonomy of Models for Respiration Signal Extraction

As presented in Table 3.1, various approaches have been developed to address the challenge of extracting video-based respiration signal. Each method and model

presents unique capabilities and characteristics, reflecting the evolving landscape of research and technological advancements in non-contact respiratory monitoring. Understanding and categorizing these methods are essential as they provide insight into the underlying mechanisms and guide future development in the field.

The column "Transformer" in Table 3.1 indicates whether a model utilizes transformer architecture as part of its structure. Transformers are a type of deep learning model originally designed for tasks in natural language processing but have been increasingly adapted for computer vision applications due to their ability to efficiently handle complex data patterns. Models with a "Yes" in the Transformer column, such as RhythmFormer, incorporate transformer-based mechanisms to analyze video data. This approach can be advantageous for capturing long-range dependencies and temporal relationships within video sequences, potentially improving the extraction of subtle respiratory signals. In contrast, models marked with "No" in this column rely on alternative architectures, such as convolutional neural networks (CNNs) or more traditional computer vision techniques that can focus on local spatial features rather than broad dependencies. The distinction provided by this column helps researchers and practitioners identify models that leverage the enhanced learning capacity and flexibility of transformer architectures in the context of non-contact respiration monitoring.

## 3.5. The Research Gap

The field of non-contact respiration extraction has witnessed significant advancements in recent years, yet several key areas remain unexplored or underdeveloped, presenting important opportunities for future research. One notable gap in the current literature is the lack of approaches that leverage transformer-based optical flow techniques for respiration extraction. Despite the success of transformer architectures in various computer vision tasks, their potential in capturing the complex spatial and temporal dependencies inherent in respiratory movements has not been fully explored. This represents a significant opportunity to enhance the accuracy and robustness of respiration signal extraction methods.

Furthermore, while transformer-based models have been applied to various physiological signal extraction tasks, such as in RhythmFormer for rPPG signals, there is currently no end-to-end transformer-based model specifically designed for extracting respiration signals from video data. This absence is particularly noteworthy given the potential of transformer models to outperform existing convolutional and hybrid approaches by fully leveraging their strengths in capturing long-range dependencies and multiscale features. The development of such a model could mark a significant advancement in the field of non-contact respiration monitoring.

In addition to these methodological gaps, there is room for further evaluation of existing approaches across different datasets. For example, the VitalCam dataset, among others, has not been extensively used in testing current respiration extraction methods. Although not critical, expanding the range of datasets used for evaluation could provide additional insight into the performance of these methods under varied conditions. This broader test could potentially reveal areas for refinement and improvement, contributing to the overall robustness of respiration extraction techniques.

Addressing these research gaps could lead to significant advancements in the field of non-contact respiration monitoring. By exploring transformer-based optical flow techniques, developing end-to-end transformer models for respiration extraction, and rigorously evaluating these approaches on diverse datasets like VitalCam, researchers could potentially improve the accuracy, robustness, and applicability of these techniques in various clinical and non-clinical settings. Such advances would not only contribute to the theoretical understanding of respiration extraction methods but also have practical implications for real-world applications in healthcare monitoring and beyond.

# 4. Contribution

The "Contribution" chapter explains the methodologies and processes used in the research for extracting and predicting respiration signals from video data. It covers the project structure, implementation and models, pre-processing steps, and the process for filtering and normalizing signals. The chapter also details the techniques for extracting respiratory rates and describes the use of RAFT to detect respiration, including the steps from face detection to signal calculation. Additionally, the training of transformer models like SimpleViT and RhythmFormer, along with the application of a hybrid loss function, is examined to evaluate their performance in non-contact respiratory monitoring.

## 4.1. Methodology

The Methodology section will describe how the experiments and research was conducted and structured. The source code, including all the notebooks and evaluation results, is available in a public GitHub repository[1].

### 4.1.1. Project Structure

The project architecture is designed to facilitate a coherent and reproducible workflow, encompassing everything from model training to comprehensive data analysis. The primary components of the project structure include several key directories: Data, Notebooks, Outputs, and Respiration.

The Data directory serves as a repository for all static data, including pretrained models and essential configuration files. This ensures that all necessary resources are readily available for the project's execution.

The Notebooks directory is a vital part, containing all Jupyter notebooks developed during the course of this work. It is organized into several subdirectories, each serving a distinct purpose:

- **00-Training**: Includes notebooks dedicated to model training and fine-tuning.

[1] https://github.com/pzierahn/respiration

- **01-Extractors**: Houses notebooks that demonstrate how to run inference with various models. These notebooks also detail preprocessing steps and their effects, primarily used for debugging and parameter tuning.
- **02-Experiments**: Instrumental in generating raw data for analysis. It applies models to extract respiration signals from all subjects in the VitalCam dataset, with the resultant data stored in the Outputs directory.
- **03-Analysis**: Contains notebooks that analyze different methodological approaches. These notebooks apply consistent post-processing methods to all signals to extract frequency data from both the ground truth and the predicted respiration signal.
- **04-Support**: Features miscellaneous notebooks designed to explore the dataset and demonstrate the use of specific libraries like opencv or unisens.

The Outputs directory is a comprehensive repository where all raw extracted signals from various models are stored, along with analysis results and relevant figures. This centralized storage enables a systematic approach to data management and retrieval.

The Respiration directory contains the Python modules essential for the project, organized into several sub-modules:

- **Analysis**: Includes algorithms for filtering and frequency extraction, such as the Butterworth filter and Power Spectral Density (PSD) methods.
- **Dataset**: Provides functions for reading and parsing the VitalCam datasets, including both video data and ground truths.
- **Extractor**: Implements all models and methods for the extraction of respiration signals.
- **ROI**: Region of Interrest (ROI) contains functions to detect faces and identify the chest region.
- **Training**: Supports the training process with functions and classes, such as the hybrid loss function and a scenario loader to segment training videos.
- **Utils**: Contains auxiliary functions for tasks such as loading videos, reading metadata, and preprocessing videos.

Each model is assigned a unique model identifier (model ID) and is stored under the Models directory. Alongside the model weights, each model directory includes a manifest JSON file that contains all relevant training information. This includes:

- **Training Scenarios**: Specific scenarios from the dataset used for training.
- **Testing Scenarios**: Scenarios used for evaluating the model.

- **Loss Function Configuration**: Details about the loss functions used, including any custom settings.
- **Optimizer**: Information about the optimizer applied during training.
- **Training Time**: Total time taken to train the model.
- **Frame Size**: Dimensions of the video frames used.
- **Learning Rate**: Learning rate settings used during training.

This structured approach allows for easy tracking and reproducibility of model training conditions and outcomes, ensuring that model results can be consistently replicated and analyzed.

To ensure that our methodologies can be reliably reproduced, the project includes Dockerfiles for both CPU and CUDA environments. This ensures that the project can be set up and executed consistently across different computational environments.

### 4.1.2. Implementation and Model Sources

This work used respiration and photoplethysmography (PPG) models with publicly available source code and pre-trained models. The implementation and sources of these models were carefully selected to ensure the reproducibility and comparability of the results.

The primary source for several rPPG models, including BigSmall, DeepPhys, EfficientPhys, and TS-CAN, was the *rPPG-Toolbox: Deep Remote PPG Toolbox*[2] repository [45]. This comprehensive toolbox provides implementations and pre-trained models for these algorithms, which were trained on various datasets such as BP4D, MA-UBFC, PURE, SCAMPS, and UBFC. The availability of these pre-trained models allowed for consistent and fair comparisons across different approaches.

The MTTS-CAN model, a multitask variation of TS-CAN, was implemented differently from the other rPPG models. Although the rPPG-Toolbox includes the TS-CAN version, the MTTS-CAN approach was sourced from a separate GitHub repository[3] [42]. This repository contains a pre-trained model capable of extracting both PPG and respiratory signals. However, it is worth noting that the code in this repository required some error fixes, particularly in the preprocessing of input frames, which initially did not match the description in the original paper.

For optical flow estimation two state-of-the-art models were utilized. The Flownet2 implementation was obtained from the original NVIDIA repository[4] [17] The RAFT model, meanwhile, is integrated as part of the standard torchvision library.

[2] `https://github.com/ubicomplab/rppg-toolbox`
[3] `https://github.com/xliucs/MTTS-CAN`
[4] `https://github.com/NVIDIA/flownet2-pytorch`

Lastly, the RhythmFormer model, along with its pretrained models for the MMPD, PURE, and UBFC datasets, was obtained from the authors' GitHub repository[5] [44]. This inclusion allowed for the evaluation of a transformer-based approach in the context of rPPG signal extraction.

In addition to these deep learning models, two traditional methods were implemented from scratch to provide a baseline comparison. The Lucas-Kanade method was implemented using the OpenCV library, leveraging its efficient computer vision algorithms. The Pixel Intensity method was also custom-implemented, allowing for a direct comparison with more advanced techniques.

### 4.1.3. Pre-processing

The videos in the VitalCam dataset consist of 3600 frames, with each frame having a resolution of 640x480 pixels. These videos have a sampling rate of 30 frames per second and span 120 seconds, recorded in an uncompressed AVI format. Each video file is approximately 3.1GB in size, which necessitates efficient pre-processing steps to reduce the computational load for training and inference while maintaining relevant features from the video frames. Each of the models used in this study requires specific input formats and dimensions to ensure compatibility and optimal performance.

For both the DeepPhys and the TS-CAN models, the raw video frames are downscaled to 72x72 pixels before being processed. In these models, time-dependent changes are essential to capture the variance introduced by physiological signals. Therefore, each input to the model consists of a time-difference frame along with the down-scaled raw frame. The difference frame is calculated by subtracting the previous frame from the current frame, reflecting the changes between consecutive frames (as shown in Figure 4.1).

The EfficientPhys models require that each input frame be scaled down to 72x72 pixels. Unlike DeepPhys and TS-CAN, EfficientPhys models do not need a time-difference frame. Each preprocessed frame is simply a downscaled version of the original image, capturing appearance-based features with the assumption that respiration-related signals can be derived directly from pixel intensities.

This simpler input format improves computational efficiency while preserving the necessary spatial information (as depicted in Figure 4.2).

The BigSmall architecture requires a more complex preprocessing pipeline, where each input consists of two different frames: A big frame of resolution 144x144 pixels and a small frame of resolution 36x36 pixels. The big frame is a resized version of the original frame with a color normalization applied. For the small frame, a temporal difference frame is created, capturing pixel-wise changes over time, similar

[5] https://github.com/zizheng-guo/RhythmFormer

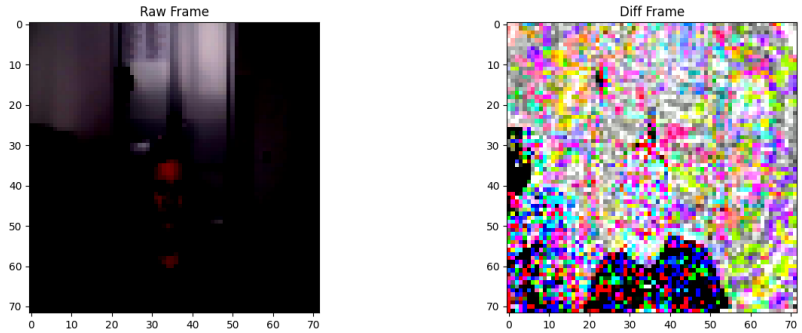


Figure 4.1.: DeepPhys and TS-CAN input frames

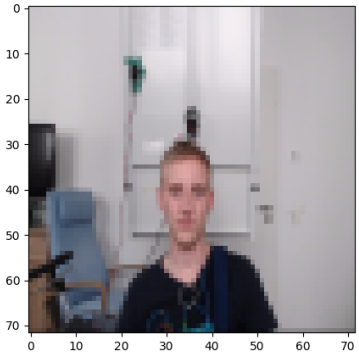

Figure 4.2.: EfficientPhys input frame

to the difference frame used in DeepPhys and TS-CAN (as shown in Figure 4.3). The difference here is that BigSmall uses a smaller resolution for the time-difference frame to focus on macrolevel changes observed over time.

The MTTS-CAN model leverages a multitask approach, where the input consists of two frames: A raw resized frame of 36x36 pixels and a normalized difference frame of the same resolution (36x36 pixels). The raw frame captures the overall appearance, while the normalized difference frame focuses on the temporal changes between consecutive frames (as shown in Figure 4.4). This two-input mechanism allows the model to extract features related to both spatial content and temporal variations in respiration. Color normalization and resizing ensure that these inputs are processed consistently across the model's layers.

For optical flow models, including FlowNet2 and RAFT, no specialized downscaling or temporal difference frame generation is required. The optical flow networks work directly on the raw 640x480 frames without any additional pre-processing.

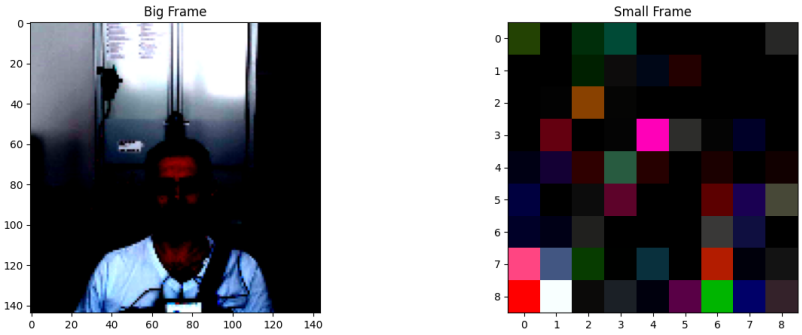


Figure 4.3.: BigSmall input frame

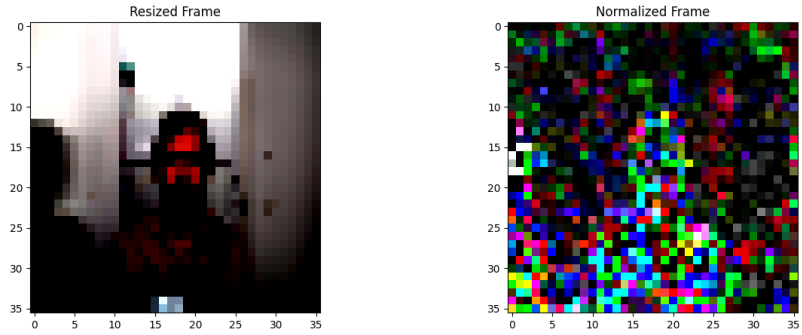


Figure 4.4.: MTTS-CAN input frame

### 4.1.4. Filtering and Normalization

The raw signals extracted from the models undergo a critical filtering and normalization process to improve the quality and reliability of respiratory information. This process serves multiple essential functions in signal processing and analysis.

Firstly, filtering is applied to eliminate non-respiratory frequencies and smooth the frequency response. This step is crucial because it helps isolate the respiratory component from other physiological and environmental signals that may be present in the raw data. Using Butterworth bandpass filters with carefully selected cutoff frequencies, we can focus on the frequency range typically associated with human respiration, which is generally between 0.1 Hz and 0.5 Hz (6 to 30 breaths per minute) [46]. This filtering technique effectively removes high-frequency noise and low-frequency drift, resulting in a cleaner respiratory signal.

In the case of photoplethysmography (PPG) signals, an important aspect of the filtering process is the removal of heart rate components. The PPG waveform

contains cardiac and respiratory information, with the cardiac component typically being more prominent. By applying appropriate filters, we can separate the respiratory modulation from the stronger cardiac pulsations, allowing for more accurate respiratory rate estimation [47].

Furthermore, the filtering process plays a crucial role in noise reduction. Various sources of noise can contaminate the raw signals, including motion artifacts, electromagnetic interference, and baseline wandering. By implementing well-designed filters, we can significantly attenuate these unwanted components, thereby improving the signal-to-noise ratio and enhancing the overall quality of the respiratory signal [24].

Normalization of filtered signals is another vital step in the processing pipeline. This procedure helps to standardize the amplitude of respiratory signals between different subjects and recording conditions. Normalization techniques, such as scaling the signal to a fixed range or applying adaptive normalization methods, ensure that respiratory information is comparable across various datasets and reduce the impact of inter-subject variability [48].

By employing these filtering and normalization techniques, we can obtain cleaner, more reliable respiratory signals that are better suited for subsequent analysis and interpretation. This processed data allows for a more accurate estimation of respiratory rates and enables the extraction of other relevant respiratory parameters, ultimately contributing to improved respiratory monitoring in both clinical and research settings.

The effects of filtering and normalization on the raw signals can be clearly observed in Figure 4.5, which illustrates the transformation of the signal at each stage of the preprocessing workflow.

### 4.1.5. Frequency Extraction

To accurately analyze and compare the performance of various models in predicting respiratory rates, we employ a comprehensive approach to frequency extraction. This subsection details the steps involved in extracting the dominant frequencies from the predicted respiratory signals and the ground truth data.

To ensure statistical robustness and the validity of the p-values, a sliding window approach is used. This method allows maintaining a sufficient sample size for analysis and provides a more detailed understanding of the performance of the model over different periods. The parameters for the sliding window are as follows:

- **Window Sizes:** [30s, 40s, 50s, 60s, 70s, 80s, 90s, 100s, 110s] seconds. These varying window sizes help capture different temporal dynamics within the respiratory signals.

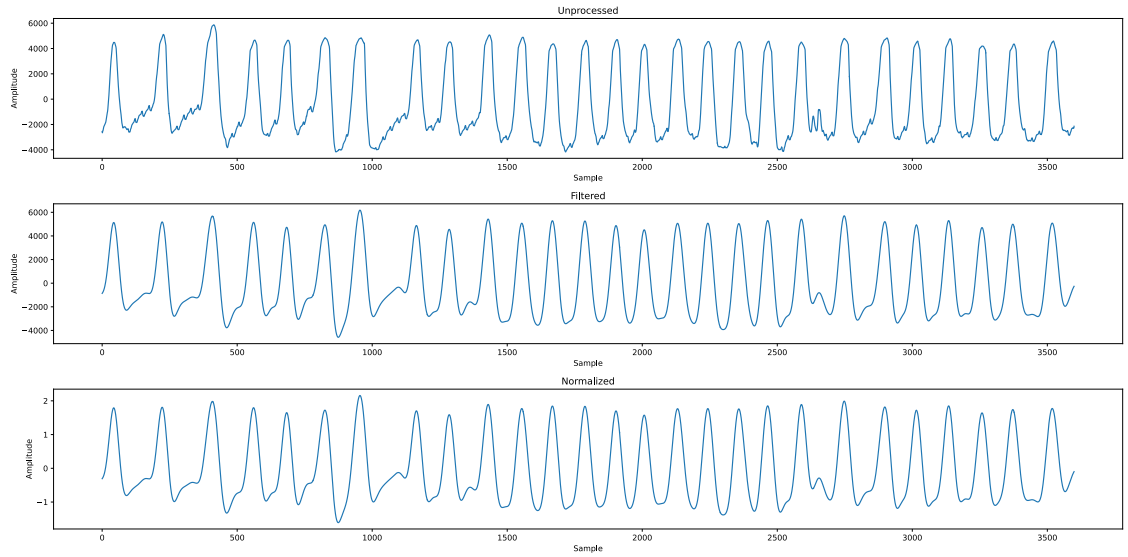


Figure 4.5.: Effects of normalization and filtering on the raw signals.

- **Stride:** 1 second. The stride defines how much the window moves forward at each step, ensuring an overlap and allowing for a more granular analysis of the signal.

The sliding window is applied to the predicted respiratory signals and the ground-truth respiration signals. For each window size and position, the subsequent steps are followed:

1. **Signal Segmentation:** The predicted and ground truth signals are segmented according to the window size and stride. This process generates multiple overlapping segments of signals, allowing us to analyze the respiratory rates over various durations.

2. **Frequency Calculation:** For each segmented signal, the Power Spectral Density (PSD), Peak Count (PC), Crossing Point (CP) and Negative Feedback Crossing Point (NFCP) methods are used to calculate the frequencies.

3. **Dominant Frequency Extraction:** From each PSD, we extract the dominant frequency within the range typically associated with normal breathing (e.g., 0.1 Hz to 0.5 Hz). This dominant frequency is considered the estimated respiratory rate for that particular segment.

After aggregating all frequencies from all segments, for both the predicted and ground truth signals, we calculate the following error metrics:

- **Mean Absolute Error (MAE):** This metric quantifies the average absolute difference between the predicted and actual respiratory rates, providing an overall measure of prediction accuracy.
- **Pearson Correlation Coefficient (PCC):** PCC evaluates the linear correlation between predicted and actual respiratory rates. A high PCC indicates that the model predictions closely follow the trend of the ground truth.

By aggregating the results from all window sizes and strides, we obtain a comprehensive set of performance metrics for each model. This multi-window approach ensures that our analysis accounts for various temporal spans and provides a robust evaluation of the models' predictive accuracy. The resulting metrics (MAE and PCC) are used to assess and compare the performance of each model, aiding in identifying the most effective approaches for non-contact respiratory rate monitoring.

## 4.2. Extraction respiration with RAFT

In this section, we outline the detailed processing steps involved in extracting respiration signals using the RAFT model.

### 4.2.1. Chest Detection

Initially, the face of the subject is identified using the Haar cascades algorithm provided by the OpenCV Python library. This classical computer vision technique is robust and efficient, making it well-suited for real-time applications. The detected face provides a bounding box which serves as the reference point for subsequent steps.

From the face bounding box, the coordinates of the subject's chest area are estimated. This is a crucial step because the chest movement is directly correlated with the respiration process. An example of the detected chest region can be observed in Figure 4.6, which visually demarcates the areas of interest for signal extraction.

### 4.2.2. Calculating Motion Vectors

The next step involves calculating the motion vectors for consecutive frames. This is achieved by using the RAFT from the torchvision library. These motion vectors capture the fine-grained movements between frames, which are essential for deriving the respiration signal. The visualization of these motion vectors is presented in Figure 4.7, where the left image shows the vectors superimposed on the frame, and the right image conveys the magnitude of these vectors using color coding.

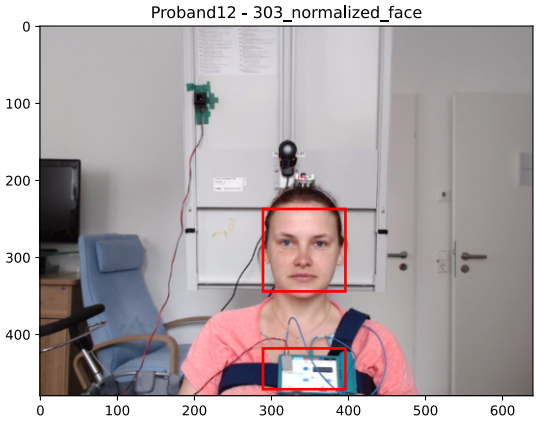


Figure 4.6.: Face and chest regions of interests

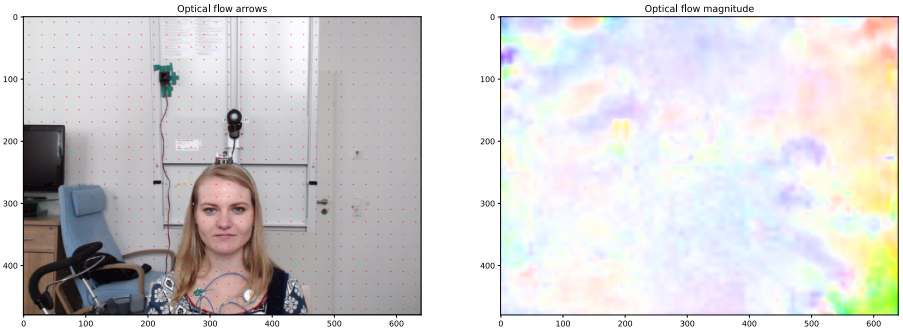


Figure 4.7.: Visualised optical flow between two frames

### 4.2.3. Extracting Vertical Motion Components

Once the motion vectors are computed, the focus is shifted to the vertical component of these vectors. Research has shown that vertical motion is highly correlated with the breathing signal due to the nature of chest movements during respiration [2]. Therefore, the vertical amplitudes of the motion vectors are extracted and averaged over each frame within the detected chest region. Figure 4.8 illustrates the averaged vertical motion vectors along with their standard deviation, providing a preliminary view of the unprocessed respiration signal.

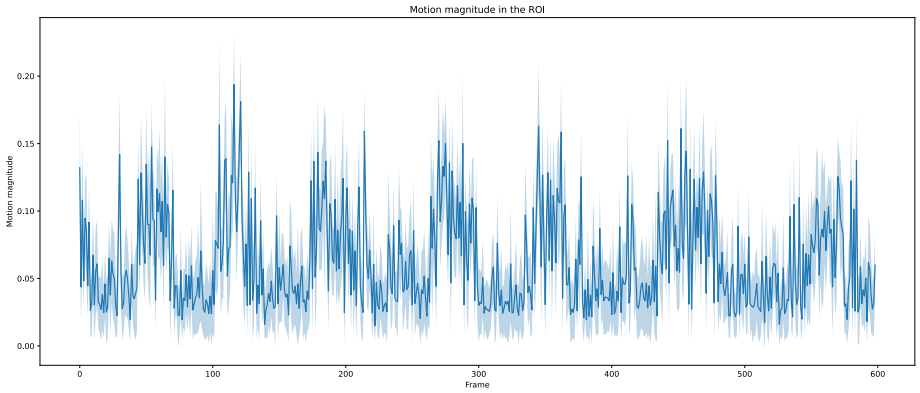


Figure 4.8.: Mean motion vectors in the chest area for each frame

### 4.2.4. Filtering and Normalization

To enhance the quality and reliability of the extracted signal, filtering and normalization processes are applied next. Filtering removes non-respiratory components and noise, while normalization ensures that the signal is standardized across different subjects and recording conditions. The effects of this pre-processing stage are demonstrated in Figure 4.9, which contrasts the processed signal with the raw data and includes the ground truth for direct comparison. This processed signal represents the final output, ready for further analysis and interpretation.

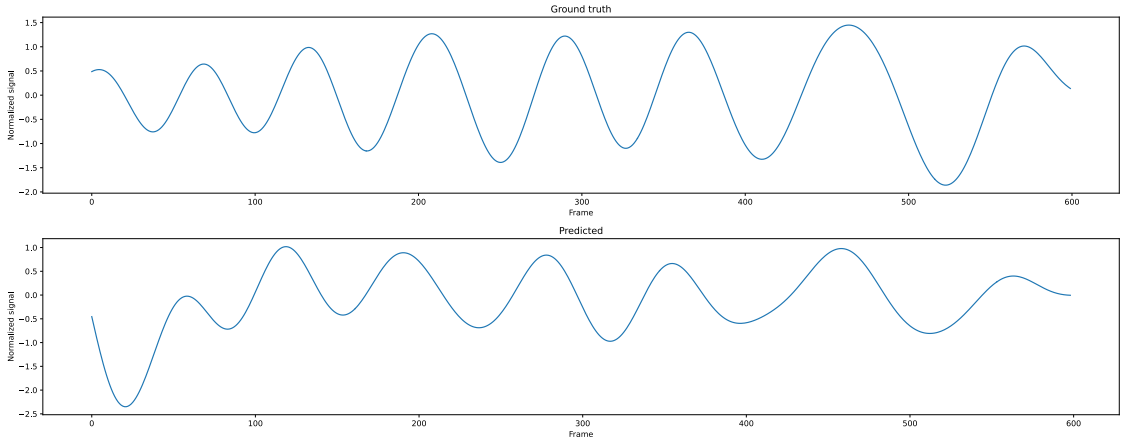


Figure 4.9.: Comparison between the ground truth and the predicted breathing signal

These meticulous steps ensure the extraction of high-quality respiration signals, aligning well with the goals of accurate and reliable respiratory monitoring in clinical and research environments.

### 4.2.5. Frequency Extraction

To evaluate the performance of the model in predicting respiratory rates, we used four frequency extraction methods: Power Spectral Density (PSD), Peak Counting (PK), Crossing Points (CP), and Negative Feedback Crossing Points (NFCP). These techniques help identify the dominant frequency that corresponds to the respiration rate.

The results are visualized in Figure 4.10, which shows the relationship between predicted and actual respiratory rates in beats per minute (BPM). Each plot uses a heatmap to display the density of predictions, with brighter colors indicating a higher density. The red diagonal line represents perfect predictions, where the predicted rates match the ground truth.

- **PSD**: This method reveals how power varies with frequency, highlighting dominant respiration frequencies.
- **PK**: Counts the peaks in the signal, linking them to the respiration rate.
- **CP and NFCP**: Identify where the signal crosses a reference level, estimating respiratory cycles.

These methods collectively ensure robust assessment and comparison of predicted respiratory rates with actual values.

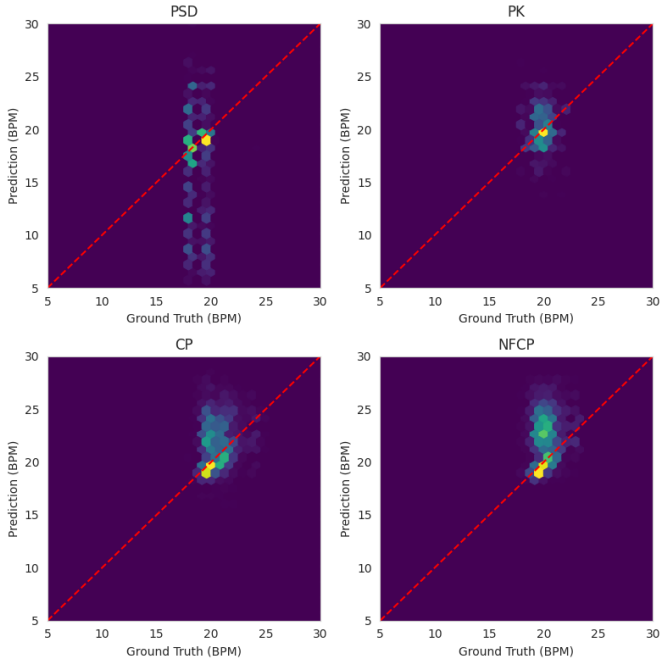


Figure 4.10.: Comparison of frequency extraction methods: PSD, PK, CP, and NFCP. Brighter colors indicate higher density of predictions

## 4.3. Respiration Transformer

The Respiration-Transformer section provides a comprehensive exploration of transformer-based models used to predict respiratory signals from video data. It is divided into several key subsections: Models, Scenarios, Hybrid Loss Function, and Training. The Models subsection introduces two primary architectures, SimpleViT and RhythmFormer, explaining their contrasting methodologies for respiratory signal extraction. The Scenarios subsection discusses the two distinct recording environments used to train the models, the natural lighting scenario, and the normalized face scenario. Next, the Hybrid Loss Function subsection explains the novel combination of multiple loss components. Lastly, the Training section details the split of the data set, the training process and specific training parameters such as batch size, which influence the frequency resolution and overall performance of the model. This structured approach to model development allows for a detailed evaluation of the strengths of transformer-based architectures in respiratory signal extraction.

### 4.3.1. Model Architectures

The Models subsection focuses on two primary architectures employed in predicting respiratory signals: SimpleViT [49] and RhythmFormer [44]. Each of these

models brings a distinctive methodology to the problem, using different aspects of transformer-based architectures.

The SimpleViT model represents an advancement of the original Vision Transformer (ViT) model. It was obtained from the vit-pytorch[6] repository on GitHub. SimpleViT processes single frames to predict respiration at specific points in time. It is designed to capture intricate details from individual frames, focusing on spatial features extracted from each snapshot. This model benefits from the established ViT framework, which emphasizes the importance of attention mechanisms to model relationships within the data effectively. Using a single frame for each prediction, SimpleViT aims to provide precise frame-specific respiratory signals without relying on temporal context from adjacent frames.

On the other hand, the RhythmFormer model takes a different approach by incorporating multiple frames to predict respiration. This model includes time-shifting modules, enabling it to leverage temporal information from multiple frames to make predictions. RhythmFormer is designed to capture dynamic changes and temporal dependencies within the data, which are crucial to accurately modeling respiratory patterns. Using a sequence of frames, this model can better understand the evolution of respiratory signals over time, potentially leading to more robust predictions that account for variations in the input data.

Both models reflect different strategies for dealing with the task of predicting respiratory signals. SimpleViT's approach of using single frames highlights its ability to focus on specific spatial details and provides a cleaner, more straightforward prediction mechanism. In contrast, RhythmFormer's use of multiple frames and the integration of time-shifting modules underscores its ability to capture temporal dynamics, offering a richer representation of input data over time.

These models are integral to our overall methodology, providing complementary perspectives on how to extract and predict respiratory signals from visual data. By exploring both single-frame and multi-frame approaches, we aim to identify the most effective techniques for various scenarios within the task of respiratory signal prediction.

### 4.3.2. Scenarios

The models were trained using different scenarios to comprehensively evaluate their performance in diverse environments. In this context, a scenario refers to a subject being recorded in a specific setting. The most basic scenario in the VitalCam dataset is the natural lighting setting (Figure 4.11), where subjects sit still on a chair without any interference, such as changing light conditions or head rotations. This setting

[6] `https://github.com/lucidrains/vit-pytorch`

provided a controlled environment to focus on the models' ability to handle standard conditions without additional complexities.

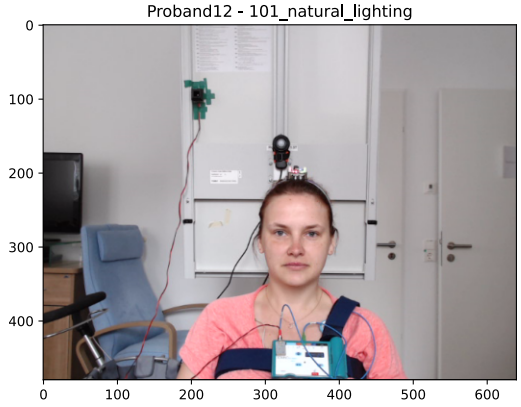


Figure 4.11.: Natural Lighting Scenario

To further advance the research, a new scenario was created called the normalized face scenario (Figure 4.12). This new scenario aims to provide a consistent coordinate framework to mitigate variations caused by head movements and facial expressions, ensuring uniformity across the dataset. One of the primary reasons for introducing the normalized face scenario is that the photoplethysmography (PPG) signal is predominantly observable in the face. By normalizing the face region, we can exclude irrelevant background areas, thereby reducing noise and improving the accuracy of respiratory signal extraction.

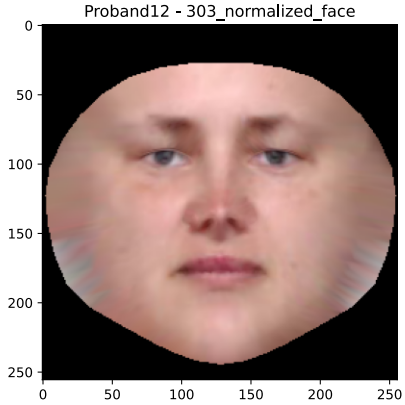


Figure 4.12.: Normalized Face Scenario

Normalization also ensures uniform processing, which is critical for reliable feature extraction and pattern recognition. Standardizing the face region across all frames

allows the models to focus on relevant features without being distracted by inconsistencies in the background or varying facial orientations. This consistent coordinate framework is vital for enhancing model performance and ensuring that the extracted features are statistically significant and reliable.

The process of creating the normalized face scenario involves texture warping. In this approach, the pixel intensities from the input face images are directly mapped onto a reference coordinate system. A 3D Morphable model (3DMM) is used to project the face onto a cylindrical surface, which facilitates triangulation and effective texture mapping [6]. This method ensures that all face images conform to a standard geometry, enabling the models to process the data more effectively and accurately extract respiratory signals.

The normalized face scenario was derived from the natural lighting scenario to ensure a fair comparison between the two settings. By basing the normalized face scenario on an existing controlled environment, we can better evaluate the benefits of normalization and its impact on the performance of the respiratory signal extraction models.

### 4.3.3. Hybrid Loss Function

In this study, a novel Hybrid Loss Function is proposed and tested to effectively train neuronal networks for respiratory signal prediction. This hybrid approach integrates several spatial and temporal loss metrics, each focused on different aspects of prediction quality. The design of this Hybrid Loss Function draws heavily from the methodology presented in the RhythmFormer paper [44], which emphasizes the importance of combining multiple loss components to capture various dimensions of prediction accuracy comprehensively. By doing so, this loss function optimizes the training process of the neuronal networks, facilitating their ability to learn complex patterns from the video data and enhancing the robustness and precision of respiratory signal predictions.

One of the components is the Kullback-Leibler (KL) divergence loss, which measures the divergence between two normal distributions derived from predicted and ground-truth power spectral densities (PSDs). KL divergence quantifies how much a given distribution diverges from a reference distribution, thereby aligning the spectral characteristics of predicted PSDs with those of the ground-truth PSD [44].

Pearson's correlation loss evaluates the linear correlation between the predicted and ground-truth respiratory signals. The Pearson correlation coefficient (PCC) measures the strength and direction of the linear relationship between two variables. This component ensures that the predicted signal closely follows the trends and variations of the ground-truth signal. The loss is defined as $1 - \text{PCC}$, promoting maximization of the correlation during training.

$$L_{\text{Pearson}} = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Here, $x_i$ and $y_i$ are the predicted and ground truth values, respectively, while $\bar{x}$ and $\bar{y}$ are their means.

The frequency loss is derived from the cross-entropy loss between predicted and ground-truth PSDs. By treating PSDs as probability distributions, the cross-entropy loss measures the dissimilarity between these distributions. This alignment ensures that the frequency content of the predicted signal matches that of the ground truth, which is crucial to accurately capture the periodic components of respiratory signals.

$$L_{\text{Frequency}} = \text{Cross-Entropy}(\text{maxIdx}(\text{PSD}(\text{pred\_signal})), \text{PSD}(\text{gt\_signal}))$$

The mean squared error (MSE) is a widely used metric in regression tasks and measures the average of the squared differences between predicted and actual values. In the context of respiratory signal prediction, MSE loss penalizes large deviations between predicted and ground-truth signals, promoting accuracy and smoothness.

$$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The loss of spectral convergence[7] evaluates the difference between the predicted and ground-truth PSDs on the basis of their spectral properties. Defined as the Frobenius norm of the difference between the predicted and ground-truth PSDs, normalized by the Frobenius norm of the ground-truth PSD, this loss ensures the predicted spectrum closely approximates the ground-truth spectrum.

$$L_{\text{Spectral Convergence}} = \frac{\|\text{pred\_psd} - \text{gt\_psd}\|_F}{\|\text{gt\_psd}\|_F}$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm.

The loss in spectral magnitude[8] assesses the discrepancy between the magnitudes of the predicted spectra and the ground-truth. Depending on the chosen norm (L1 or L2), this loss measures either the absolute or squared differences between the PSD magnitudes. This component aims to align the amplitude characteristics of the predicted signals with those of the ground truth, preserving the signal intensity variations.

[7] https://github.com/rishikksh20/UnivNet-pytorch/blob/master/stft_loss.py
[8] https://github.com/rishikksh20/UnivNet-pytorch/blob/master/stft_loss.py

$$L_{\text{L1}} = \sum_{i=1}^{n} |\text{pred\_psd}_i - \text{gt\_psd}_i|$$

$$L_{\text{L2}} = \sqrt{\sum_{i=1}^{n} (\text{pred\_psd}_i - \text{gt\_psd}_i)^2}$$

The overall Hybrid Loss Function is the weighted sum of all the individual loss components. Each component can be weighted according to its relative importance, allowing for a flexible and comprehensive evaluation during training.

$$L_{\text{Hybrid}} = \alpha * L_{\text{KL}} + \beta * L_{\text{Pearson}} + \gamma * L_{\text{Frequency}} + \delta * L_{\text{MSE}} + \epsilon * L_{\text{Spectral Convergence}} + \zeta * L_{\text{Spectral Magnitude}}$$

The coefficients $\alpha, \beta, \gamma, \delta, \epsilon$, and $\zeta$ are hyperparameters that adjust the contribution of each loss component to the overall hybrid loss. The Hybrid Loss Function can be adapted to different training scenarios and model requirements. This approach ensures a balanced optimization that captures both temporal and spectral nuances of the respiratory signals.

### 4.3.4. Training

For model training, a 80% / 20% split was used to ensure a robust evaluation process. This means that 80% of the data is reserved for training, while the remaining 20% is held out for testing and validation. This split ensures that the models are tested on data that they were not exposed to during training, providing a reliable measure of their generalization capabilities. Specifically, subjects 1 to 20 from the VitalCam dataset were used for training, while subjects 21 to 26 were used for testing. The models that were not trained on the VitalCam dataset use 100% of the videos for testing.

Each model was trained for 22 epochs, and the epoch that yielded the best performance on the test dataset was selected. The selection criteria involved monitoring the models' performance metrics during training and picking the weights that minimized the validation loss.

The input frames are resized to 128x128 pixels to balance the trade-off between computational efficiency and the need for sufficient spatial resolution to capture relevant features. No additional data augmentation techniques were applied, as the primary objective was to evaluate the models under consistent conditions reflective of the actual recording setup.

The batch size is a critical parameter that significantly influences the frequency resolution of the Power Spectral Density (PSD) used in the loss function (see Section

4.3.3). The batch size corresponds to the number of frames processed simultaneously by the GPU.

A larger batch size improves frequency resolution but also increases memory consumption. Therefore, an optimal batch size balances frequency resolution against available GPU memory.

The effects of the batch size on the PSD can be calculated using the following formulas:

$$\text{frequency\_step} = \frac{\text{sampling rate}}{\text{Batch Size}}$$

$$\text{frequency\_resolution} = \lceil \frac{\text{max\_freq} - \text{min\_freq}}{\text{frequency\_step}} \rceil$$

In this study, each video in the VitalCam dataset contains 3600 frames, recorded at a sampling rate of 30 Hz. Depending on the batch size chosen, the PSD frequency spectrum can be adjusted as follows:

- Split of 5: 3600 frames are divided by 5, resulting in 720 frames per batch. The frequency step is $\frac{30}{720} = 0.04$ Hz. Thus, the PSD spectrum in the respiration rate range of 0.1-0.5 Hz has approximately 10 frequencies.
- Split of 10: 3600 frames are divided by 10, resulting in 360 frames per batch. The frequency step is $\frac{30}{360} = 0.08$ Hz. Therefore, the PSD spectrum in the respiration rate range of 0.1-0.5 Hz has approximately 5 frequencies.

# 5. Analysis and Results

This chapter provides a comprehensive evaluation of various models used for non-contact respiration extraction. It assesses the performance of different architectures, comparing metrics like Pearson's correlation coefficient and mean absolute error across several categories such as optical flow models, transformer-based approaches, and pre-trained respiration models. The chapter further analyzes statistical significance using methods such as the t-test, investigates the impact of hybrid loss function components on model accuracy, and evaluates scenarios such as natural lighting versus normalized face settings. In addition, potential threats to the validity of the study are discussed, acknowledging limitations and suggesting areas for further exploration.

## 5.1. Direct Approach Comparison

This section compares the best-performing models from each architecture. The models evaluated here represent the best performers across various approaches, with the complete results for all models detailed in the Appendix A. For the analysis, a random model was added as a baseline to show the pre-processing effects.

The best-performing models from each category are:

- **rPPG**: The rPPG models RhythmFormer, EfficientPhys, TS-CAN, DeepPhys are trained on the datasets MMPD, PURE, UBFC, BP4D, and SCAMPS. An overview of all model scores can be seen in the Appendix in Section A.1.4. The best performing models are the MMPD_intra_RhythmFormer, UBFC-rPPG_TSCAN, UBFC-rPPG_EfficientPhys and BP4D_PseudoLabel_DeepPhys.
- **RAFT**: The RAFT small performs slightly better than the RAFT large model (See Section A.1.1).
- **Pixel Intensity**: The RGB version of the Pixel-Intensity algorithm performances slightly better than the grey variant (See Section A.1.1).
- **FlowNet**: The FlowNet2CS is the best performing model in the FlowNet2 model family (See Section A.1.1).

- **Respiration-RhythmFormer**: During this work many Respiration-RhythmFormer were trained. All models are analyised in detail in Section 5.4. In this model family the best model was RF_20240902_210159.
- **SimpleViT**: During this work many Respiration-RhythmFormer were trained. All models are analyised in detail in Section 5.4. In this model family the best model was RF_20240902_210159.
- **Other**: The lucas_kanade, MTTS-CAN and BigSmall models don't come in different variants.

### 5.1.1. Correlation

The correlation plots (Figure 5.1) provide a visual comparison between the predicted respiration rates (in breaths per minute, bpm) and the ground truth values. These plots use a heatmap-based visualization, where darker colors (purple) indicate a lower density of point pairs, and brighter colors (yellow/white) indicate regions with higher density.

Each plot includes three metrics in the top-left corner:

- **PCC:** Pearson Correlation Coefficient
- **P:** p-value of the correlation
- **MAE:** Mean Absolute Error
- **Red Diagonal:** Trend Line

Some key insights from the correlation plots:

- Lucas-Kanade achieved the highest correlation with the ground truth (PCC = 0.78) and the lowest MAE of 1.6 BPM.
- FlowNet2CS also performed well with a high PCC of 0.674 and an MAE of 1.9 BPM.
- raft_small, SimpleViT_20240906_131118, and RF_20240902_210159 showed similar performances, with PCCs in the range of 0.57 to 0.63 and MAEs between 2.3 and 2.4 BPM.
- The rPPG models generally exhibited moderate to low correlations (ranging from 0.198 to 0.252) and high MAEs (4.1 to 4.8 BPM).
- BigSmall and MTTS-CAN show comparable results to the rPPG models, exhibiting the lowest correlations and highest MAEs.

In general, the Lucas-Kanade and FlowNet2CS models led the pack in terms of correlation and lower errors, while the rPPG models struggled with both tasks.

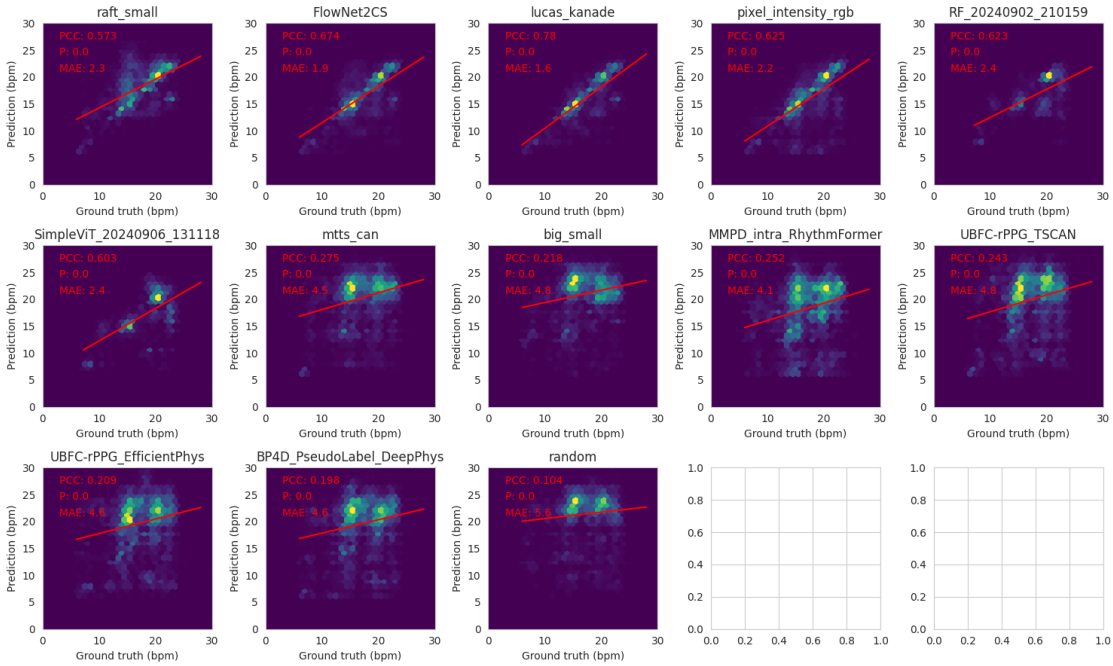


Figure 5.1.: Correlation plots of the best performing models

## 5.1.2. Bland-Altman plots

The Bland-Altman plots (Figure 5.2) show the difference between the predicted and actual respiratory rates versus the mean of both measurements. The heatmap visualization highlights the density of data points (again, darker is lower density, brighter is higher). The red line shows the mean of the differences, while the green lines represent the 95% confidence intervals (CI).

Key findings include:

- FlowNet2CS and Lucas-Kanade performed the best, with smaller mean differences (-0.9 and -1.3, respectively) and tighter confidence intervals (±3.3 and ±2.4, respectively), indicating more consistent performance.
- raft_small, RF_20240902_210159, and pixel_intensity_rgb also had small mean differences but with slightly larger confidence intervals.
- MTTS-CAN, BigSmall, and rPPG models had larger mean differences and the widest confidence intervals, revealing less consistent agreement between predicted and true values.
- The random model, unsurprisingly fared the worst, with a mean error of 3.9 and a wide confidence interval of ±5.3.

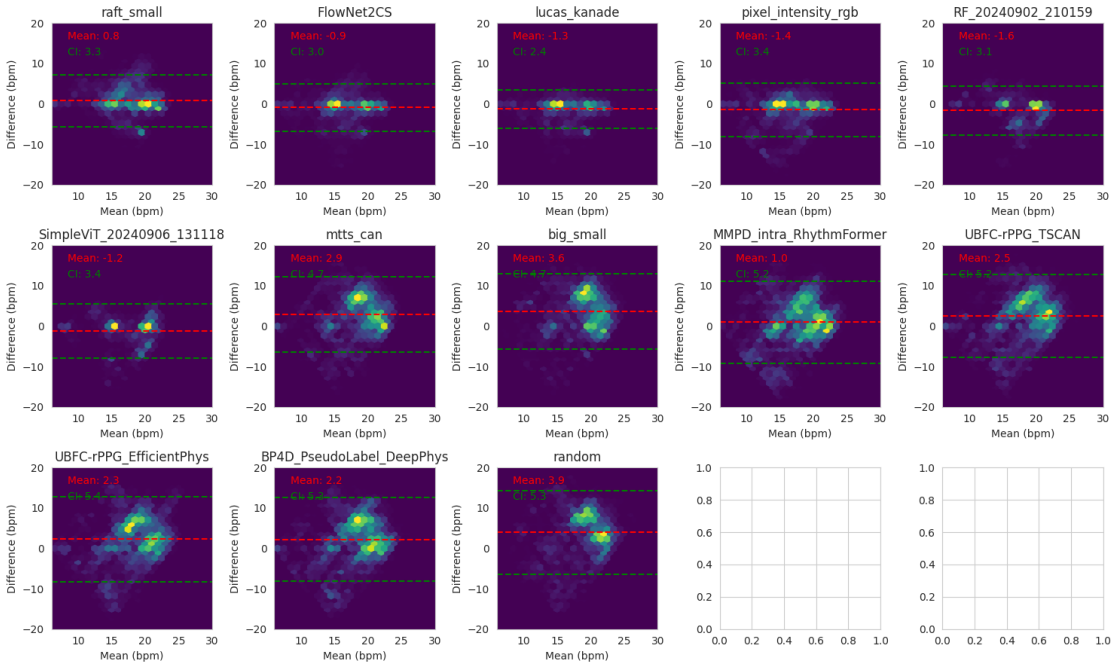


Figure 5.2.: Bland-Altman plots of the best performing models

Overall, the Lucas-Kanade model continues to demonstrate strong performance with lower variance and better agreement between predictions and ground truth.

### 5.1.3. Visualisation

Figure 5.4 presents a comprehensive visual comparison of the performance of different models, evaluated in terms of both mean absolute error (MAE) and Pearson's correlation coefficient (PCC). The illustration highlights the distinctions between the models and provides insight into their precision in predicting respiratory rates.

A closer examination of the visualized data reveals that the Lucas-Kanade model emerges as a top performer, demonstrating the highest PCC (0.78) and achieving the lowest MAE (1.64 breaths per minute). This result is closely followed by the FlowNet2CS model, which also shows a strong correlation with the ground truth (PCC = 0.67) and maintains a low MAE of 1.92 bpm. Both models emphasize the efficacy of optical flow methods in capturing respiratory movements, making them highly reliable for estimating respiratory rates.

The RAFT small and SimpleViT models also demonstrate commendable performance, with their PCC values hovering between 0.57 and 0.60, and their MAE scores just over 2.3 bpm. Similarly, the transformer-based Respiration-RhythmFormer model, encoded as RF_20240902_210159 in the results, achieved a comparable PCC of

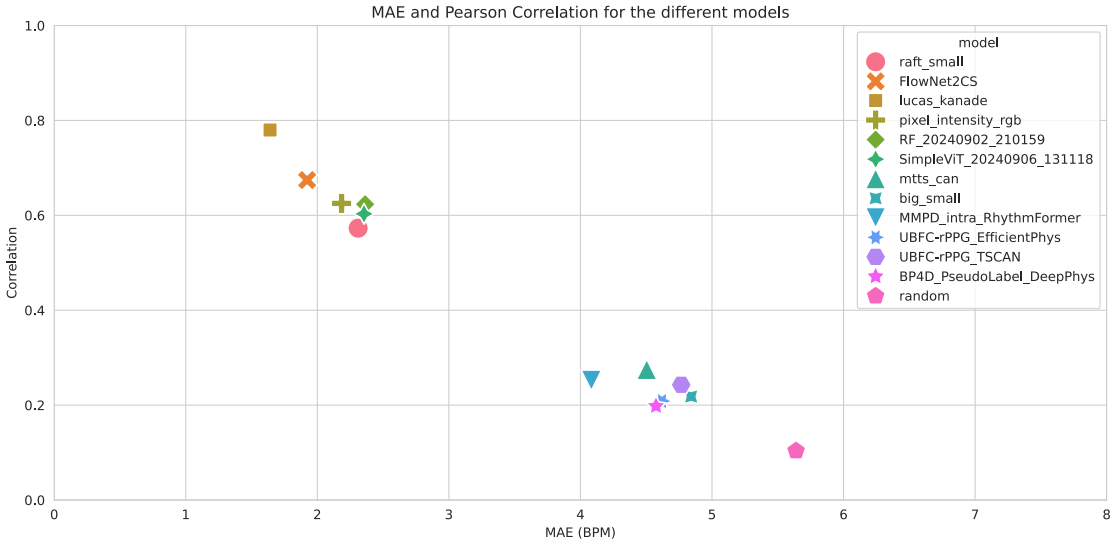


Figure 5.3.: MAE and Pearson correlation of the best performing models

0.62 and an MAE of 2.36 bpm. Notably, these performances place both RAFT and transformer models in a competitive range with optical flow methods, highlighting their potential in non-contact respiration monitoring.

However, models such as MTTS-CAN, BigSmall, and several rPPG pre-trained models, including RhythmFormer, EfficientPhys, TS-CAN, and DeepPhys, fall short in terms of predicting respiratory rates with high accuracy. These models exhibit much higher MAE values (ranging from 4.08 to 4.84 bpm) and achieve relatively low Pearson correlations (ranging from 0.20 to 0.27). Although rPPG techniques are traditionally focused on heart rate estimation, the results suggest they may struggle to transfer that success to respiration rate monitoring in the specific context of the VitalCam dataset.

Finally, the Random baseline model, used as a control, exhibits by far the poorest performance, with a PCC of 0.10 and an MAE of 5.64 bpm, further underscoring the added value provided by the more sophisticated models evaluated in this study.

In conclusion, the visual comparison highlights the dominance of optical flow-based models such as Lucas-Kanade and FlowNet2CS, while transformer-based models such as SimpleViT and Respiration-RhythmFormer also demonstrate strong promise. Conversely, the pre-trained rPPG models show substantial room for improvement in estimating respiratory rates from video data in this setting.

The numerical results of the performance metrics (MAE and PCC) for each model are summarized in Table 5.1 for further reference.

| Model | MAE | PCC | p-value |
|---|---|---|---|
| raft_small | 2.31 | 0.573 | 0.0 |
| FlowNet2CS | 1.923 | 0.674 | 0.0 |
| lucas_kanade | 1.641 | 0.78 | 0.0 |
| pixel_intensity_rgb | 2.184 | 0.625 | 0.0 |
| RF_20240902_210159 | 2.364 | 0.623 | 0.0 |
| SimpleViT_20240906_131118 | 2.355 | 0.603 | 0.0 |
| mtts_can | 4.504 | 0.275 | 0.0 |
| big_small | 4.842 | 0.218 | 0.0 |
| MMPD_intra_RhythmFormer | 4.083 | 0.252 | 0.0 |
| UBFC-rPPG_EfficientPhys | 4.62 | 0.209 | 0.0 |
| UBFC-rPPG_TSCAN | 4.766 | 0.243 | 0.0 |
| BP4D_PseudoLabel_DeepPhys | 4.574 | 0.198 | 0.0 |
| random | 5.639 | 0.104 | 0.0 |

Table 5.1.: MEA, PCC and p-values of the best performing models

## 5.2. Group Comparison

The various models evaluated for the estimation of respiratory rates can be categorized into the following groups: Optical flow, Respiratory-RhythmFormer (R-RhythmFormer), SimpleViT, Pretrained Respiration Models, Pretrained rPPG Models, and a Random model. Each group consists of models employing similar architectural principles or methodologies, and their performance was compared to determine which approach is most effective in predicting respiration rates. Figure 5.4 shows the mean absolute error (MAE) and Pearson's correlation coefficient for the different models in each group.

Using an independent (two-sample) t-test, the groups were compared against each other to statistically assess their performance differences. The t-test is a standard method used to compare the means of two independent groups to determine whether there is statistical evidence that the mean of the associated population is significantly different.

The results of these comparisons are shown in the matrix shown in Figure 5.5. In the matrix:

- A **higher t-value** indicates a more significant divergence in performance between the two groups.
- A **positive t-value** indicates that the group listed on the bottom axis performs better than the group on the left.
- A **negative t-value** shows that the group on the left outperforms the group on the bottom.

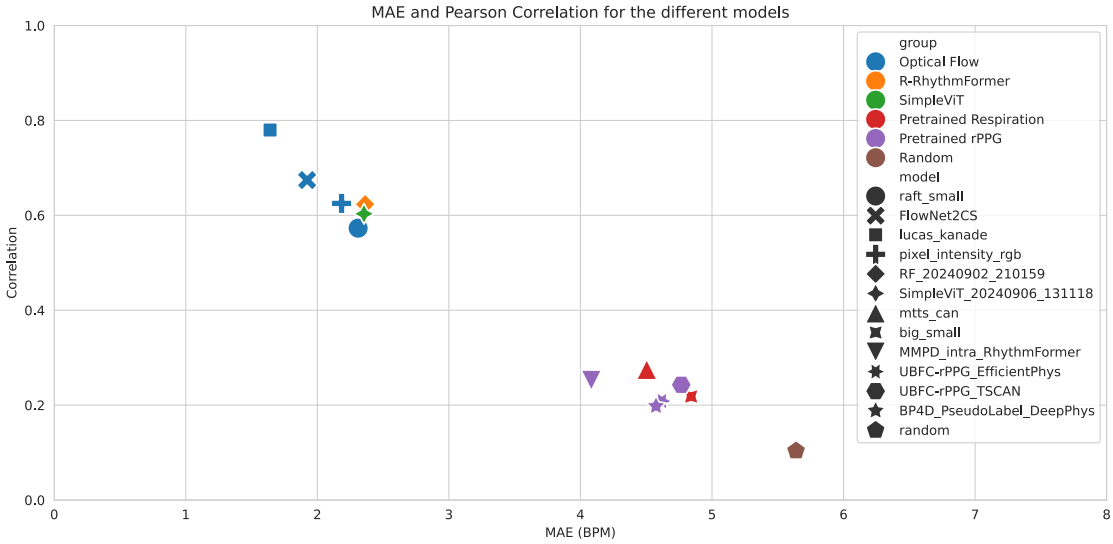


Figure 5.4.: MAE and Pearson correlation of the models

- A **t-value of zero** implies that there are no significant differences between the performances of the groups.

The Key Findings are the following:

- **Optical flow models** outperform all other groups. As seen in Figure 5.5, Optical Flow models consistently perform better across all comparisons, which is attributed to their ability to capture accurate motion-based features related to respiratory movements.

- **The SimpleViT and R-RhythmFormer models** perform similarly and both groups are close in performance, slightly trailing the Optical Flow models. These transformer-based approaches manage to capture spatial and temporal features well for respiration signal prediction, outperforming the pretrained rPPG and respiration models.

- **Pretrained respiration models** perform poorly. They exhibit inferior performance relative to almost all other groups, except for the random model. These models were initially trained on other datasets and were not fine-tuned on the VitalCam dataset, which may explain their subpar results.

- **Pretrained rPPG models** outperform the pretrained respiration models but still show mediocre performance overall. They fare better than the random model and the pretrained respiration group but lag behind the models explicitly designed for respiratory rate estimation.

- **The random model**, as expected, performs the worst, acting as a baseline to show the minimal level of performance. Its outputs are random guesses, and it significantly underperforms all other groups, as shown by its position at the bottom-right corner of the T-statistics matrix.

In conclusion, the Optical Flow group emerges as the most effective approach for non-contact respiratory rate estimation. The SimpleViT and R-RhythmFormer models also demonstrate promising results, slightly trailing behind the Optical Flow group. Meanwhile, the Pretrained Respiration and rPPG models noticeably underperform, possibly due to suboptimal adaptation to the specific dataset and task at hand.

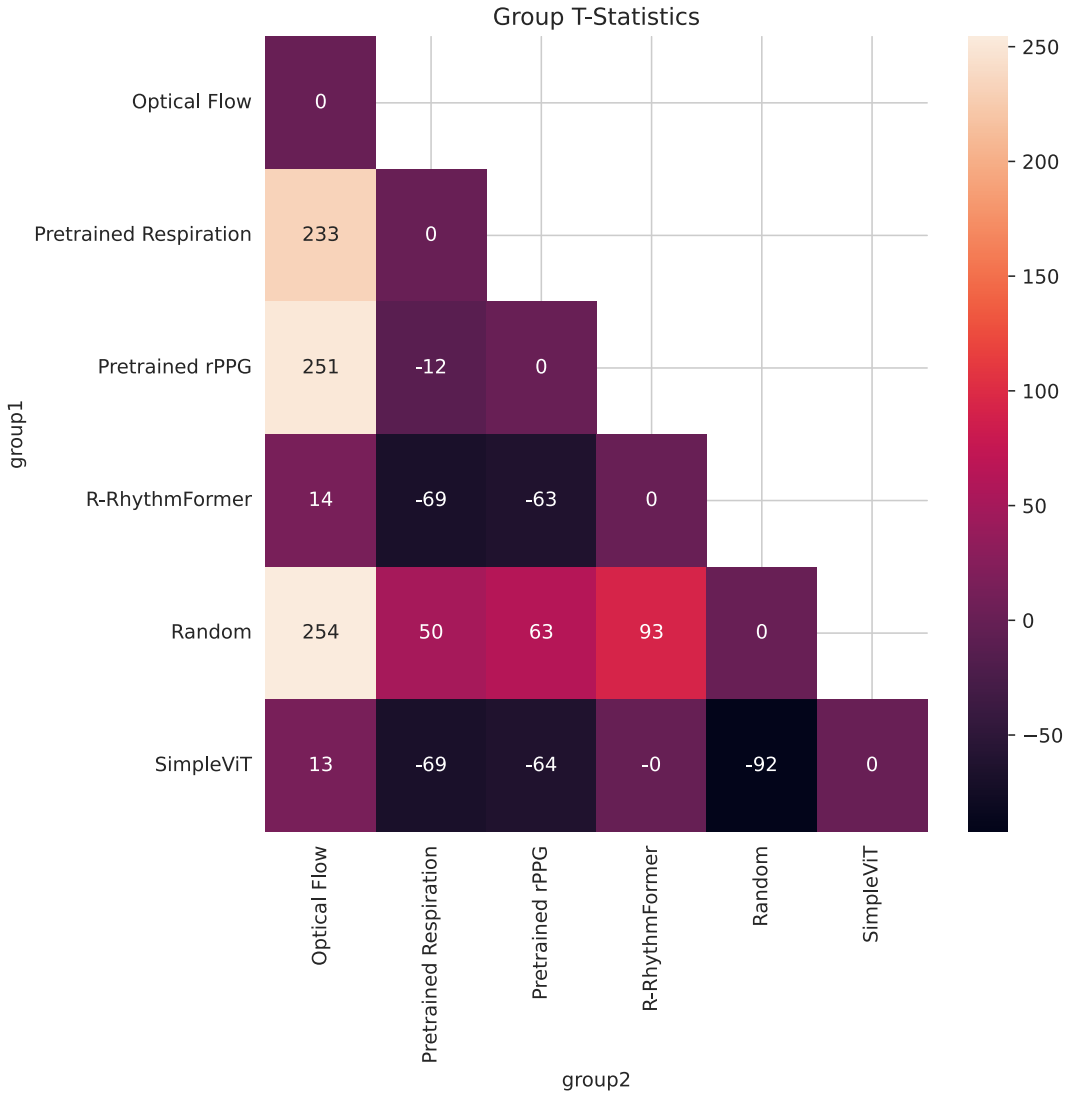


Figure 5.5.: T-Stats of the groups

## 5.3. SimpleViT vs RhythmFormer

As illustrated in section 5.2, both the SimpleViT and RhythmFormer models demonstrate comparable performance when trained on the same data and optimized with the same hybrid loss function. These two transformer-based models, while architecturally distinct, each exhibit strong capabilities in predicting respiratory signals from video data. However, the differences in their internal mechanics lead to varying trade-offs between performance, resource efficiency, and model complexity.

The key distinction between these two models lies in the way they process video frames and extract respiratory information. The RhythmFormer model employs a sophisticated approach utilizing multiple frames to make each prediction. It uses its Time-Shifting Module (TSM), which explicitly conveys information between consecutive frames, allowing the model to capture temporal dependencies and dynamic changes within the data. This temporal information enables RhythmFormer to capture subtle, time-dependent variations in respiratory patterns that may span multiple frames. The added complexity in processing comes at the cost of increased computational and memory overheads, making RhythmFormer more demanding in terms of resources.

In contrast, the SimpleViT model, a more streamlined variant of the original Vision Transformer (ViT), processes each frame independently. It does not employ any form of time-shifting or temporal modeling like RhythmFormer. Instead, it uses a more straightforward approach with just one frame per prediction, focusing solely on the spatial features present in that single frame. Although this limits the SimpleViT model from capturing temporal dependencies, its simpler architecture translates into significant gains in computational efficiency. SimpleViT is smaller in size, requiring fewer computational and memory resources, making it more resource-efficient than the RhythmFormer model, particularly when deployed in environments with constrained hardware capabilities.

Despite architectural differences and contrasting strategies for processing video inputs, the comparative performance results show that both models achieve similar levels of accuracy in terms of predicting respiratory signals. This is particularly noteworthy for SimpleViT, as it manages to keep pace with RhythmFormer despite using a single frame for each prediction, highlighting the efficiency of its design.

In conclusion, SimpleViT offers a lightweight, resource-efficient alternative to RhythmFormer, making it well suited for applications where computational power or memory are limited. However, RhythmFormer, with its ability to analyze temporal dependencies across multiple frames, may be more advantageous in scenarios where dynamic temporal variations in respiratory signals are more pronounced or critical. Ultimately, the choice between these models depends on the specific requirements of the deployment environment, balancing performance and resource constraints.

## 5.4. Influence of Hybrid Loss Function components

This section discusses the influence of different components of the loss function on the overall performance of models trained with the RhythmFormer architecture. All models were evaluated using the scenario "101_natural_lighting" from the VitalCam dataset. The hybrid loss consists of several components:

- **frequency**: Cross-entropy loss applied to the Power Spectral Density (PSD) frequency spectrum.
- **spectral_convergence**: Evaluates the difference between predicted and ground-truth PSDs using the Frobenius norm, normalized by the ground-truth PSD norm.
- **mse**: Standard Mean Squared Error, which penalizes large deviations between predicted and ground-truth signals to enhance prediction accuracy and smoothness.
- **pearson**: Pearson correlation loss that evaluates the linear relationship between the predicted and the ground truth respiratory signal.
- **spectral_magnitude**: Measures the discrepancy in magnitudes between predicted and ground-truth spectra, either as absolute or squared differences.
- **norm**: Measures the Kullback-Leibler (KL) divergence between predicted and ground-truth power spectral densities normal distributions.

These components can be enabled or disabled during model training. To study their individual and joint influences, we conducted multiple experiments, allowing different combinations of loss components, listed in Table 5.2, and then analyzed the performance of the resulting models. Due to an exponential blow-up in the number of possible component combinations (since each component can either be enabled or disabled), it was impractical to test all combinations comprehensively. Therefore, only a subset of combinations could be tested, focusing on specific combinations of interest based on existing literature and prior intuitions. An overview of the tested loss components combinations can be seen in Table 5.2.

To investigate how much each component contributes to the performance of the model, we divided the experiments into two groups for each component. Group 0 consists of models where the component is disabled and group 1 consists of models where the component is enabled. We then calculated the Mean Absolute Error (MAE) for each group and applied an Independent (Two-Sample) T-Test to assess whether the differences in MAE are statistically significant.

Tables 5.3 show the results. The t-value and the p-value from the T-Test are used to determine the influence of each component. A positive t-value indicates that the component improves model performance (lower MAE when enabled), while a

| Model | freq | mse | norm | pearson | spec_conv | spec_mag | mae | pcc |
|---|---|---|---|---|---|---|---|---|
| RF_20240902_094648 | 0.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 2.983 | 0.415 |
| RF_20240902_123124 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 2.668 | 0.406 |
| RF_20240902_152753 | 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 3.085 | 0.285 |
| RF_20240902_181443 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 3.235 | 0.229 |
| RF_20240902_210159 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.364 | 0.623 |
| RF_20240902_234749 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 5.085 | 0.227 |
| RF_20240903_023334 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.026 | 0.068 |
| RF_20240903_051739 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 2.452 | 0.592 |
| RF_20240903_080307 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.997 | 0.38 |
| RF_20240903_104800 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.916 | 0.353 |
| RF_20240904_231209 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 3.825 | 0.252 |
| RF_20240905_094727 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.643 | 0.559 |
| RF_20240905_134835 | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 3.416 | 0.159 |
| RF_20240905_173644 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 1.0 | 3.956 | 0.145 |
| RF_20240905_233757 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 3.367 | 0.34 |
| RF_20240906_213402 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 3.981 | 0.163 |
| RF_20240907_190942 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 3.147 | 0.416 |

Table 5.2.: Results of the Respiration-RhythmFormer model, with different combinations of loss function components

negative t-value indicates a detrimental effect (higher MAE when enabled). The columns MAE_0 and MAE_1 show the respective MAE values when the component is disabled or enabled. The final results highlight how much the performance varies with each individual loss component.

| Component | t_value | p_value | mae_0 | mae_1 | models_0 | models_1 | points_0 | points_1 |
|---|---|---|---|---|---|---|---|---|
| frequency | 26.243 | 0.0 | 3.5 | 3.1 | 7 | 10 | 75600 | 108000 |
| spectral_convergence | 13.824 | 0.0 | 3.4 | 3.2 | 9 | 8 | 97200 | 86400 |
| mse | 12.481 | 0.0 | 3.4 | 3.2 | 11 | 6 | 118800 | 64800 |
| pearson | 7.486 | 0.0 | 3.4 | 3.3 | 8 | 9 | 86400 | 97200 |
| spectral_magnitude | 6.22 | 0.0 | 3.3 | 3.2 | 10 | 7 | 108000 | 75600 |
| norm | -20.977 | 0.0 | 3.2 | 3.5 | 12 | 5 | 129600 | 54000 |

Table 5.3.: T-scores for each loss component. A positive t-value indicates a positive effect on model performance, while a negative t-value indicates a negative influence.

The key finding includes the following.

- **Frequency loss** has the strongest positive influence on model performance. The model that scores the best in terms of Mean Absolute Error (MAE) (RF_20240902_210159) uses only the frequency loss component. The corresponding t-value is the highest among the components ($t = 26.24$), with models employing this component achieving an average MAE of 3.1, compared to 3.5 for those that do not.

- **Norm loss** is the only component that shows a strongly negative influence on model performance. As indicated by its highly negative t-value ($t = -20.98$),

models that use this component perform worse. The model that was trained using only the norm loss component (RF_20240902_234749) exhibited the highest MAE (5.08), demonstrating its detrimental effect.

- **Spectral Convergence loss** performs well both individually and in combination with other components. The model trained with only the spectral convergence loss (RF_20240903_051739) performs slightly worse than the best model but still shows good performance, indicating its utility when combined with other loss components.
- **MSE loss** surprisingly performs poorly on its own. The model trained solely with MSE (RF_20240903_023334) is among the lower-performing models, with a MAE of 4.03. This result is unexpected, given that MSE is commonly used in time-series regression problems and in many photoplethysmography extraction models (see Section 3.1.2).
- **Pearson and spectral_magnitude losses** show mediocre performance. When used individually, they yield average performance compared to other loss components (e.g., RF_20240902_181443 and RF_20240903_080307). However, they provide slight improvements when used in combination with other loss functions.

In conclusion, the results demonstrate that different loss components have varying levels of influence on the performance of respiration signal extraction models. The frequency loss from the PSD emerges as the most crucial component, while the norm loss negatively affects the model accuracy. Spectral convergence and MSE show useful but not game-changing performance, and Pearson and spectral magnitude seem to have a limited but additive impact on improving the MAE.

The detailed evaluation of these results is visualized in Figure 5.7, while Table 5.3 statistically quantifies the influence of each loss component.

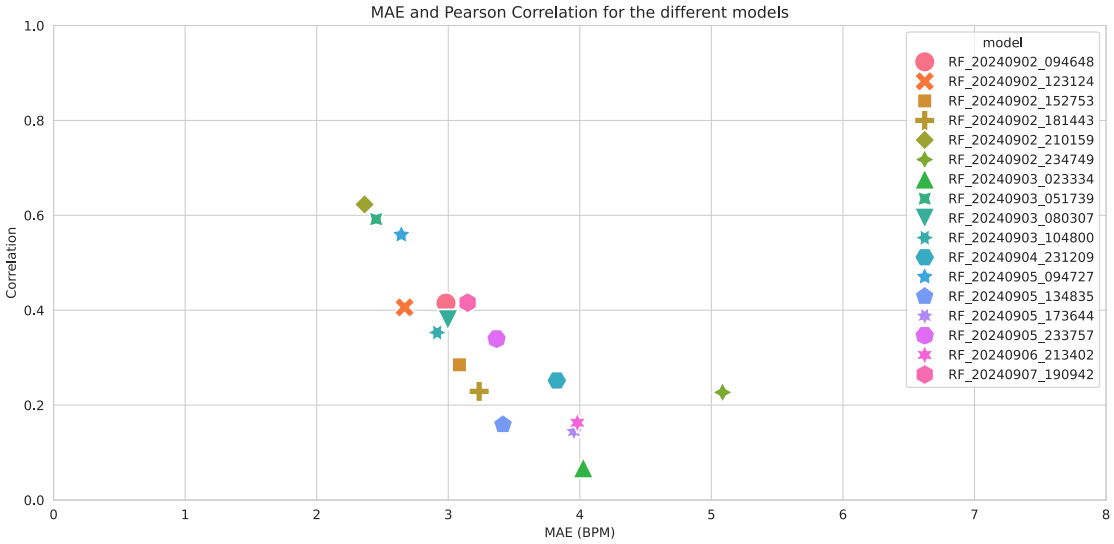


Figure 5.6.: Respiration-RhythmFormer MAE and Pearson scores across different loss functions

## 5.5. Setting Influence

This section compares the effect of the "101_natural_lighting" setting versus the normalized face setting described in Section 4.3.2. The "101_natural_lighting" setting includes the subject's head, chest area, arms, and background, while the normalized face setting only includes the face projected on a plane (see Section 4.3.2).

The hypothesis was that the normalized face setting contains less noise compared to the natural lighting setting, resulting in better results in estimating the respiration rate.

### 5.5.1. Direct Comparsion

As described in Section 5.4, the best performing loss function is the cross-entropy Power Spectral Density (PSD) loss. Models were trained using this loss function for both settings to identify differences in performance.

The model "RF_20240902_210159" was trained in the "101_natural_lighting" setting. The model "RF_20240908_194112" was trained on the normalized face setting. Performance metrics are as follows.

- **RF_20240902_210159**:

  - Mean Absolute Error (MAE): 2.364
  - Pearson Correlation: 0.623
- **RF_20240908_194112**:
  - Mean Absolute Error (MAE): 4.79
  - Pearson Correlation: -0.109

Contrary to the hypothesis, the model trained in the normalized face setting performed significantly worse in terms of MAE and correlation.

### 5.5.2. Grouped Comparison

The models shown in Figure 5.7 and Table 5.4 were trained using various combinations of the components of the hybrid loss function in both the settings "101_natural_lighting" and "303_normalized_face". The objective was to determine whether any specific loss function combination could work better in the normalized face setting.

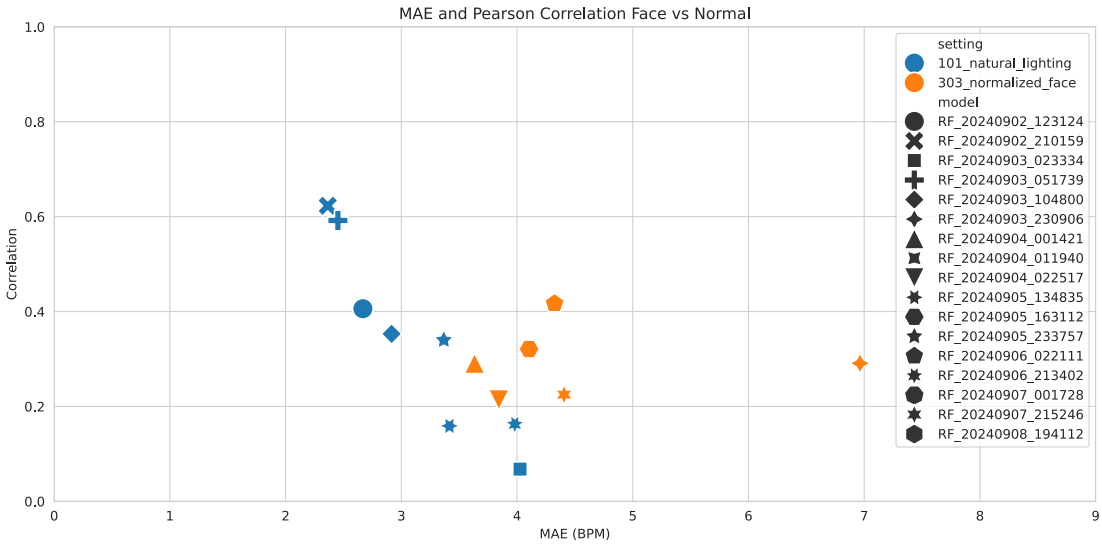


Figure 5.7.: Respiration-RhythmFormer MAE and Pearson correlation grouped by setting

The results indicate that the normalized face setting consistently underperforms compared to the natural lighting setting. When comparing both groups collectively:

- **Normalized Face Group (Face Group)**:
  - Mean Absolute Error (MAE): 4.622

| model | setting | freq | mse | norm | pearson | spec_con | spec_mag | mae | pcc | p |
|---|---|---|---|---|---|---|---|---|---|---|
| RF_20240902_123124 | 101_natural_lighting | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 2.668 | 0.406 | 0.0 |
| RF_20240902_210159 | 101_natural_lighting | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.364 | 0.623 | 0.0 |
| RF_20240903_023334 | 101_natural_lighting | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.026 | 0.068 | 0.0 |
| RF_20240903_051739 | 101_natural_lighting | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 2.452 | 0.592 | 0.0 |
| RF_20240903_104800 | 101_natural_lighting | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.916 | 0.353 | 0.0 |
| RF_20240903_230906 | 303_normalized_face | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 6.964 | 0.291 | 0.0 |
| RF_20240904_001421 | 303_normalized_face | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 3.633 | 0.291 | 0.0 |
| RF_20240904_011940 | 303_normalized_face | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.851 | -0.025 | 0.01 |
| RF_20240904_022517 | 303_normalized_face | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.843 | 0.214 | 0.0 |
| RF_20240905_134835 | 101_natural_lighting | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 3.416 | 0.159 | 0.0 |
| RF_20240905_163112 | 303_normalized_face | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 4.104 | 0.321 | 0.0 |
| RF_20240905_233757 | 101_natural_lighting | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 3.367 | 0.34 | 0.0 |
| RF_20240906_022111 | 303_normalized_face | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 4.324 | 0.417 | 0.0 |
| RF_20240906_213402 | 101_natural_lighting | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 3.981 | 0.163 | 0.0 |
| RF_20240907_001728 | 303_normalized_face | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 4.683 | -0.082 | 0.0 |
| RF_20240907_215246 | 303_normalized_face | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 4.407 | 0.225 | 0.0 |
| RF_20240908_194112 | 303_normalized_face | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.79 | -0.109 | 0.0 |

Table 5.4.: Natural light vs normalized faces: Loss configuration and results

- **Natural Lighting Group (Normal Group)**:
  - Mean Absolute Error (MAE): 3.149

The independent (two-sample) T-Test yields a t-value of 87.362 and a p-value of 0.0, indicating a statistically significant difference between the groups. These results suggest that the natural lighting setting works much better beyond a reasonable doubt.

### 5.5.3. Conclusion

The hypothesis that the normalized face setting would yield better results as a result of reduced noise was not supported by the findings. Both direct and grouped comparisons show that models trained in the natural lighting setting perform significantly better than those trained on the normalized face setting. Future work should investigate the reasons behind this disparity and explore other potential improvements or adjustments to the normalized face setting to improve its effectiveness.

## 5.6. Threats to Validity

The conclusions drawn from this study are subject to several limitations and potential threats to validity, which must be acknowledged to strengthen the robustness and generalizability of our findings. One of the primary limitations is the limited scope of the dataset and the scenarios used. The study relies solely on the VitalCam dataset, which consists of only 26 subjects, each recorded for 120 seconds. This relatively small dataset may constrain the generalizability of the results. Furthermore, the models that were trained on the dataset have even less evaluation data since only

the last six subjects were used for testing. The use of a single specific scenario from the dataset for evaluation further limits the applicability of the findings to diverse settings or conditions. Future research should aim to include additional datasets and scenarios to comprehensively validate the models.

Another significant threat to validity pertains to the recording conditions and their applicability to real-world scenarios. The recording setup depicted in Figure 4.6 includes a device mounted on the chest with clear cables that emerge from it. Such visual cues are not typically present in real-world applications. The device and cables provide distinct edges that optical flow algorithms can easily track, potentially leading to overly optimistic performance metrics. In practical applications, individuals often wear plain clothing without distinctive patterns, making motion tracking more challenging. The effectiveness of the optical flow algorithms tested in this study may vary significantly under such conditions. Therefore, future studies should explore more realistic and varied recording environments to accurately assess model performance.

Additionally, only the RhythmFormer and SimpleViT models were trained using the VitalCam dataset. The remaining models, while designed for tasks related to photoplethysmography (PPG), were evaluated using pre-trained versions not specifically optimized for the dataset at hand. It is plausible that other models, such as those focused on PPG, could achieve performance levels similar to those of the RhythmFormer or SimpleViT models if they were explicitly trained on the VitalCam dataset for the purpose of extracting respiratory signals. Further research should involve training these models on the VitalCam dataset to provide a fair and comprehensive comparison.

The unique characteristics of the VitalCam dataset, including the specific recording setup and scenarios, may also limit the ability to generalize the results to broader populations and different contexts. To strengthen external validity, it is essential to evaluate models against other diverse datasets and incorporate more varied recording conditions and subject demographics.

Recognizing these threats to validity is vital for interpreting the results of this study and guiding future research directions. Addressing these limitations can provide a more thorough and reliable evaluation of non-contact respiration extraction methods, ultimately leading to improved models that are applicable across various real-world conditions and diverse populations. Future work should focus on expanding the diversity of the dataset, improving training methodologies, and exploring more realistic recording environments to strengthen the generalizability and robustness of the findings.

# 6. Conclusion

The Conclusion section is structured into two parts: a summary of the key findings and insights from the research, followed by suggestions for future work to further advance the field of non-contact respiration extraction technologies.

## 6.1. Summery

This work presents a comprehensive framework for testing and evaluating non-contact respiration models, contributing to the advancement of vital sign monitoring from video-based data. By leveraging the VitalCam dataset, a wide range of respiration extraction models were tested, such as traditional optical flow algorithms, modern neural network architectures, and combinations of different machine learning techniques. The findings provide valuable insights into model performance and uncover practical implications for improving non-invasive respiratory monitoring technologies.

The study demonstrated that optical flow-based models (specifically FlowNet2 and Lucas-Kanade) consistently outperform other approaches in predicting respiratory rates. These optical flow models achieved high Pearson Correlation Coefficients (PCC) and low Mean Absolute Error (MAE) scores, indicating their superior ability to capture subtle respiratory motions.

Another key finding of this research is that modern neural networks do not always outperform simpler solutions. For instance, the SimpleViT model, a less sophisticated Vision Transformer, performed on par with the more complex RhythmFormer, illustrating that more advanced architectures do not necessarily guarantee superior results. This was also evident in the optical flow-based models, where the older Lucas-Kanade consistently delivered better results than the newer RAFT method.

The study also explored the utility of combining multiple loss components into a Hybrid Loss Function. Contrary to expectations, this combination did not result in significant performance improvements. The most effective loss function turned out to be the cross-entropy Power Spectral Density (PSD) loss, while models trained with the full hybrid loss struggled to outperform simpler setups. This finding demonstrates that adding complexity to optimization criteria may not always

enhance model accuracy, and simpler, more focused loss functions may suffice in some applications.

An unexpected result resulted from the comparison of different recording settings. The normalized face scenario, designed to isolate photoplethysmography (PPG) signals and reduce noise, performed worse than the general natural lighting scenario. Although PPG signals should theoretically be better captured in the normalized face scenario, the performance gap suggests that the better performing models likely rely on observing chest motion rather than facial skin variations caused by respiration. This implies that while facial PPG signals may contain valuable data, they might not be as effective for respiration estimation as previously thought, particularly when chest movement can serve as a reliable proxy.

## 6.2. Future Work

The field of non-contact respiration extraction from video is rapidly evolving, and several promising directions for future research can be identified to enhance the accuracy, applicability, and efficiency of current methodologies. This section outlines four key areas for future work: Spiking Neural Networks, expanding the dataset, integrating CNN-based extractors, and exploring mobile deployment.

### 6.2.1. Spiking Neural Networks

Spiking Neural Networks (SNNs) [50] represent a cutting-edge paradigm in artificial neural networks that more closely mimic the way biological neurons operate. Unlike traditional artificial neural networks, SNNs utilize discrete spikes to process and transmit information, enabling them to potentially offer significant energy efficiency and processing speed advantages. Future research could explore the application of SNNs for non-contact respiration extraction. By leveraging the event-driven nature of SNNs, it is conceivable that these networks could more effectively handle the temporal dynamics inherent in video data, thereby improving both the accuracy and efficiency of respiration signal extraction. The development and training of SNN-based models could lead to novel architectures optimized for real-time monitoring and low-power mobile applications.

### 6.2.2. More Datasets

One of the critical limitations identified in the current study is the reliance on a single dataset, VitalCam. Future work should focus on evaluating and training models on a broader range of datasets to enhance generalizability and robustness. For example, incorporating data sets that include a diverse range of subjects,

varying environmental conditions, and different recording setups would be beneficial. Extending the evaluation to datasets such as COHFACE [51], PURE [52], or newly acquired datasets with varied scenarios could reveal the performance of the models in different contexts and populations. Moreover, cross-dataset validation studies would provide a more comprehensive understanding of the models' generalization capabilities, thereby ensuring their applicability in various real-world scenarios.

### 6.2.3. Training PPG Models on VitalCam

As discussed in the "Threats to Validity" section (see Section 5.6), the pre-trained photoplethysmography (PPG) models were not specifically optimized for the VitalCam dataset. Future work should involve training PPG models directly on the VitalCam dataset for respiratory rate prediction, which includes fine-tuning existing models. Furthermore, a comprehensive comparison of the PPG models trained on the VitalCam dataset against other models should be performed to evaluate performance improvements and generalizability. Additionally, cross-dataset evaluations are essential to test trained models on external datasets, evaluating their ability to generalize beyond the VitalCam dataset. This approach would provide a fair comparison and potentially enhance the accuracy and robustness of PPG models for non-contact respiration monitoring.

### 6.2.4. Mobile Deployment

The practical deployment of non-contact respiration monitoring systems in real-world applications, such as remote health monitoring and mobile healthcare, requires models that are not only accurate but also resource-efficient. Future research should focus on optimizing these models for mobile deployment. Techniques such as model pruning, quantization, and knowledge distillation could be used to reduce the computational and memory footprints of the models. In addition, developing lightweight architectures that can run efficiently on edge devices is crucial. Exploring frameworks such as TensorFlow Lite [1] or PyTorch Mobile [2] to deploy these optimized models can enable real-time monitoring of respiration on smartphones, tablets, and other portable devices. This would significantly enhance the accessibility and convenience of non-contact respiratory monitoring, especially in remote and resource-limited settings.

[1] https://www.tensorflow.org/lite
[2] https://pytorch.org/mobile/home/

# A. Appendix

## A.1. Model Metrics

This section shows the correlation and Bland-Altman plots for each tested model in this work.

### A.1.1. Optical Flow

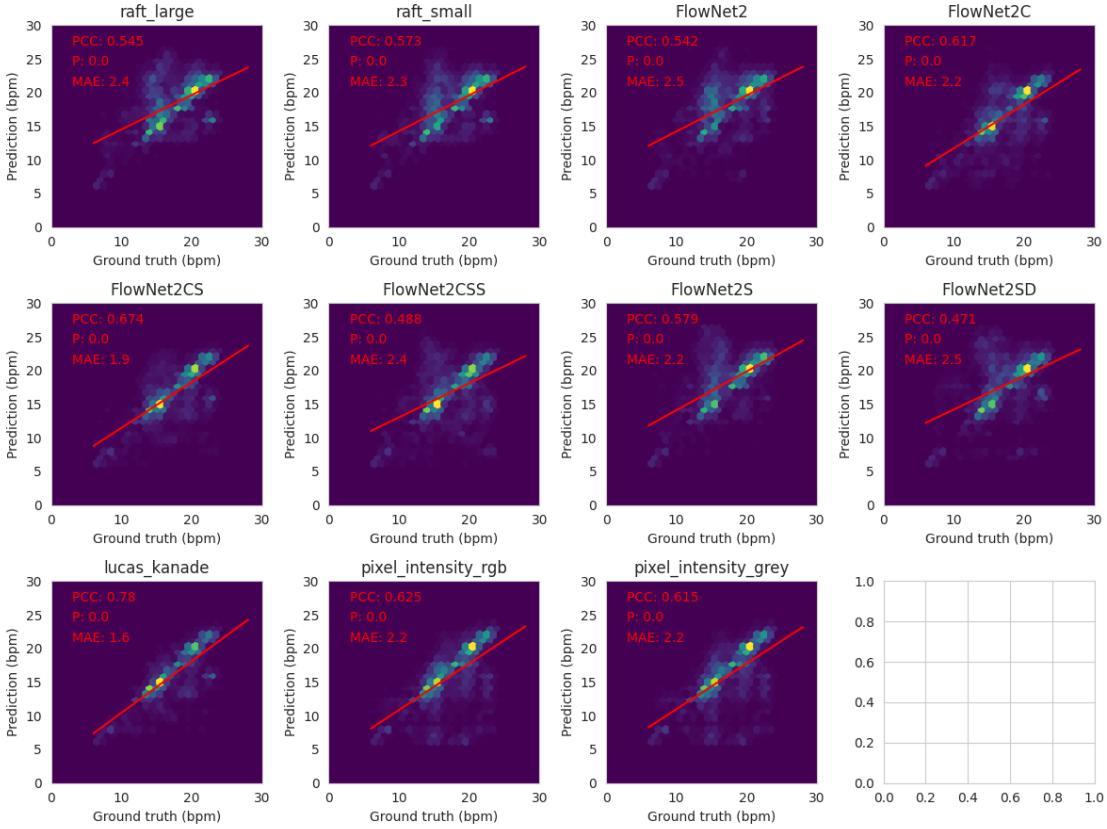


Figure A.1.: Correlation of optical flow methods

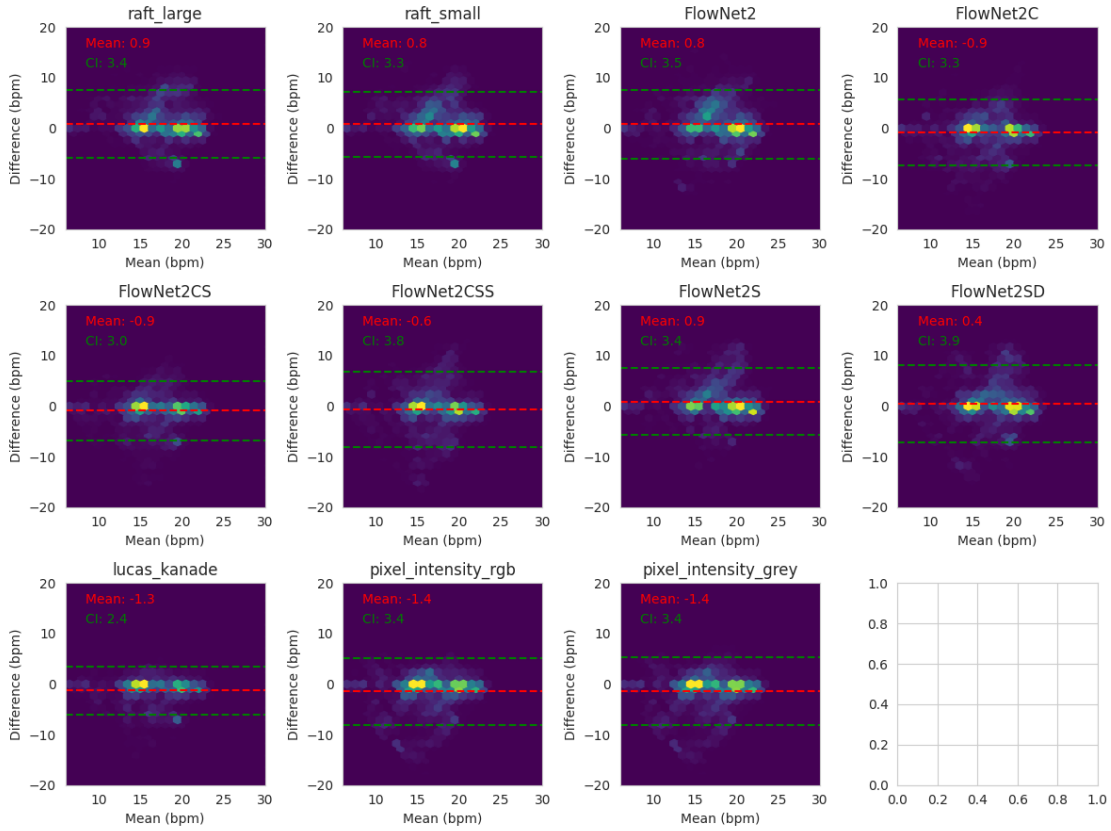


Figure A.2.: Bland-Altman plots of optical flow methods

### A.1.2. Respiration-RhythmFormer

### A.1.3. SimpleViT

### A.1.4. Pretrained rPPG Models

### A.1.5. Pretrained Respiration Models

### A.1.6. Random

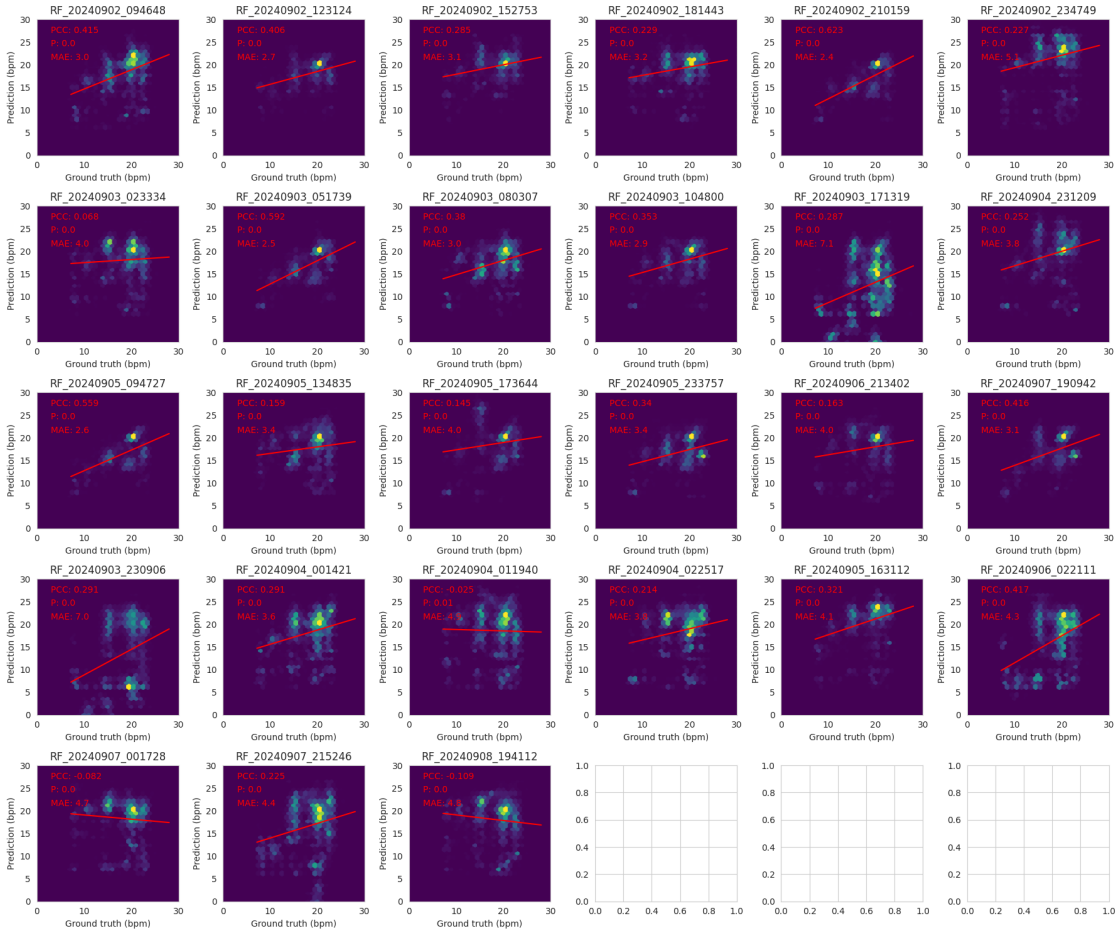


Figure A.3.: Correlation of Respiration-RhythmFormer models

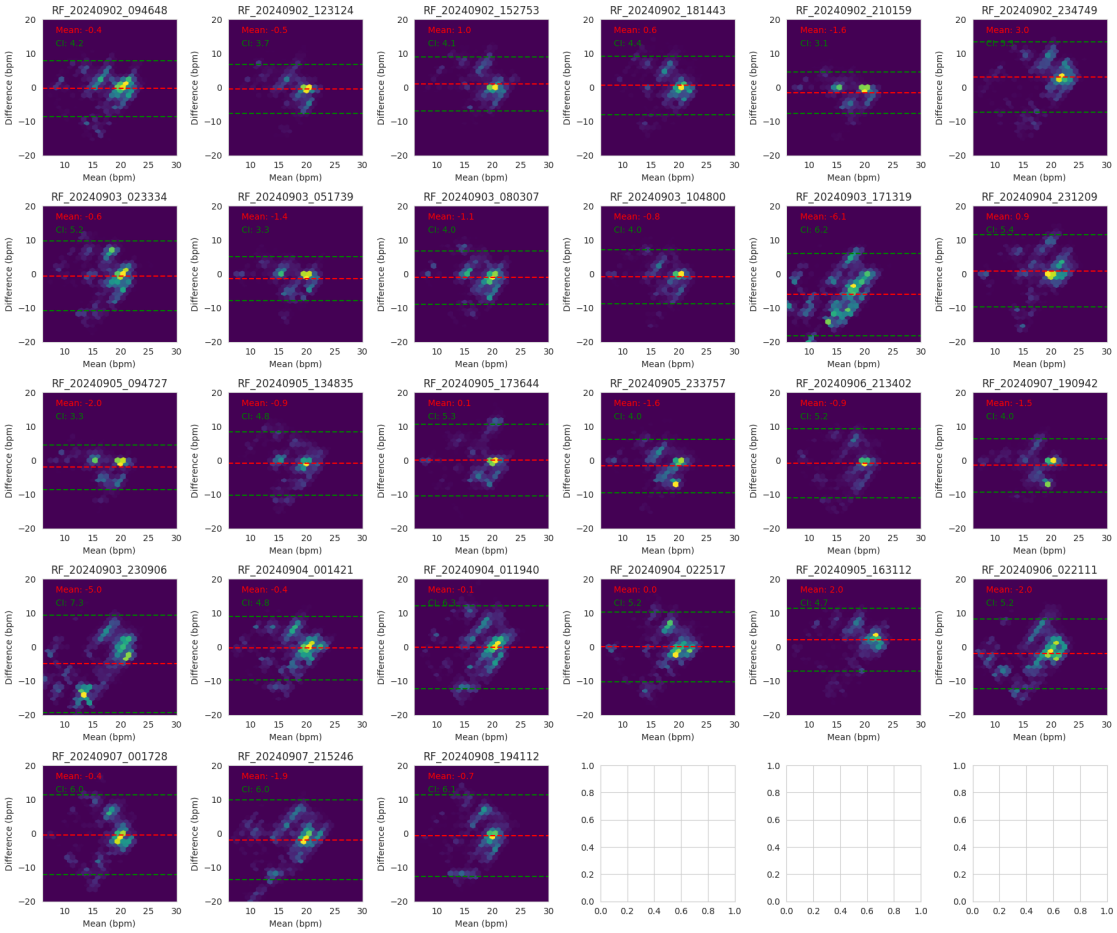


Figure A.4.: Bland-Altman plots of Respiration-RhythmFormer models

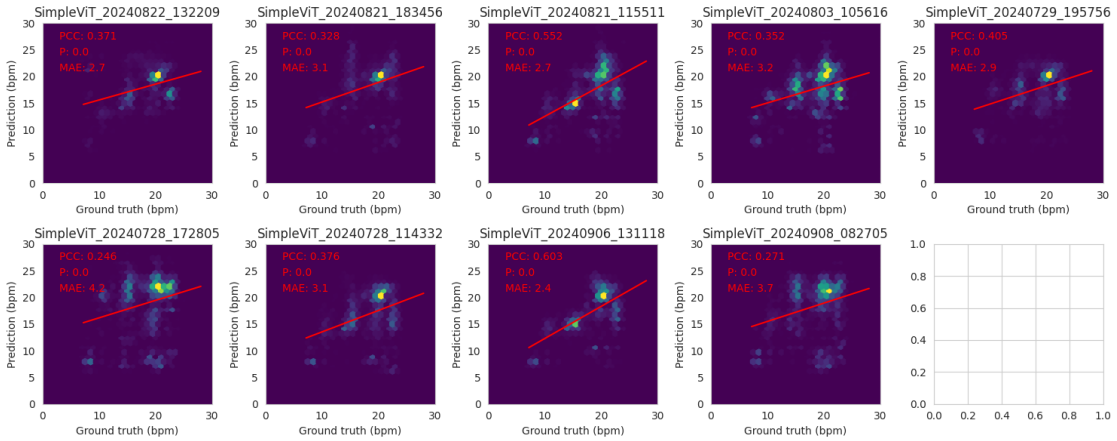


Figure A.5.: Correlation of SimpleViT models

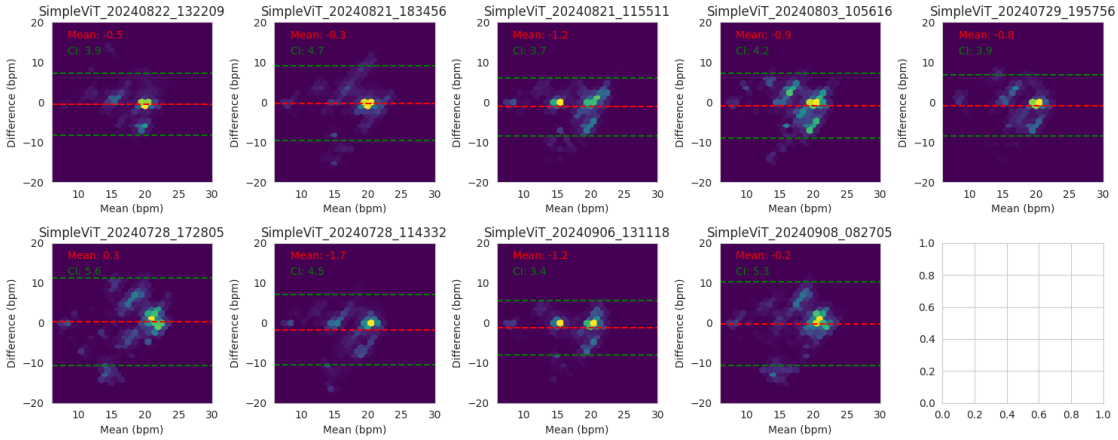


Figure A.6.: Bland-Altman plots of SimpleViT models

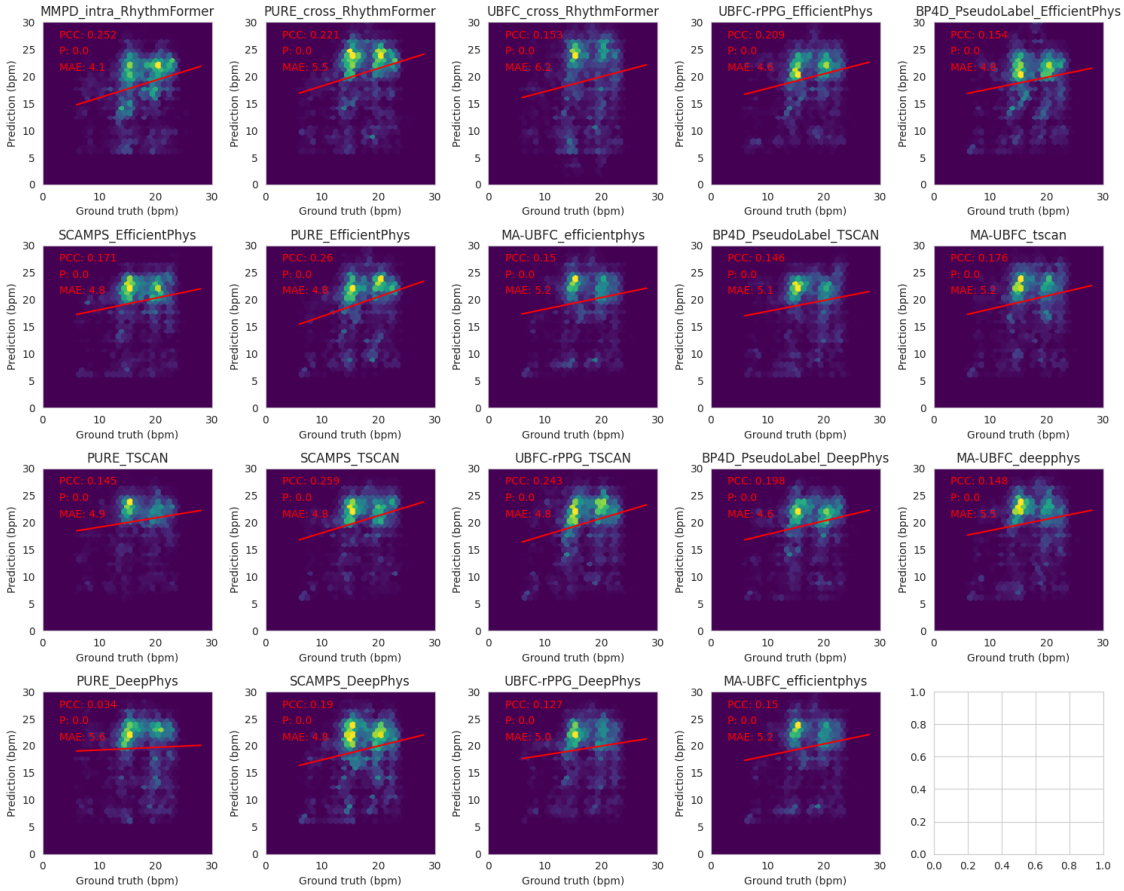


Figure A.7.: Correlation of rPPG models

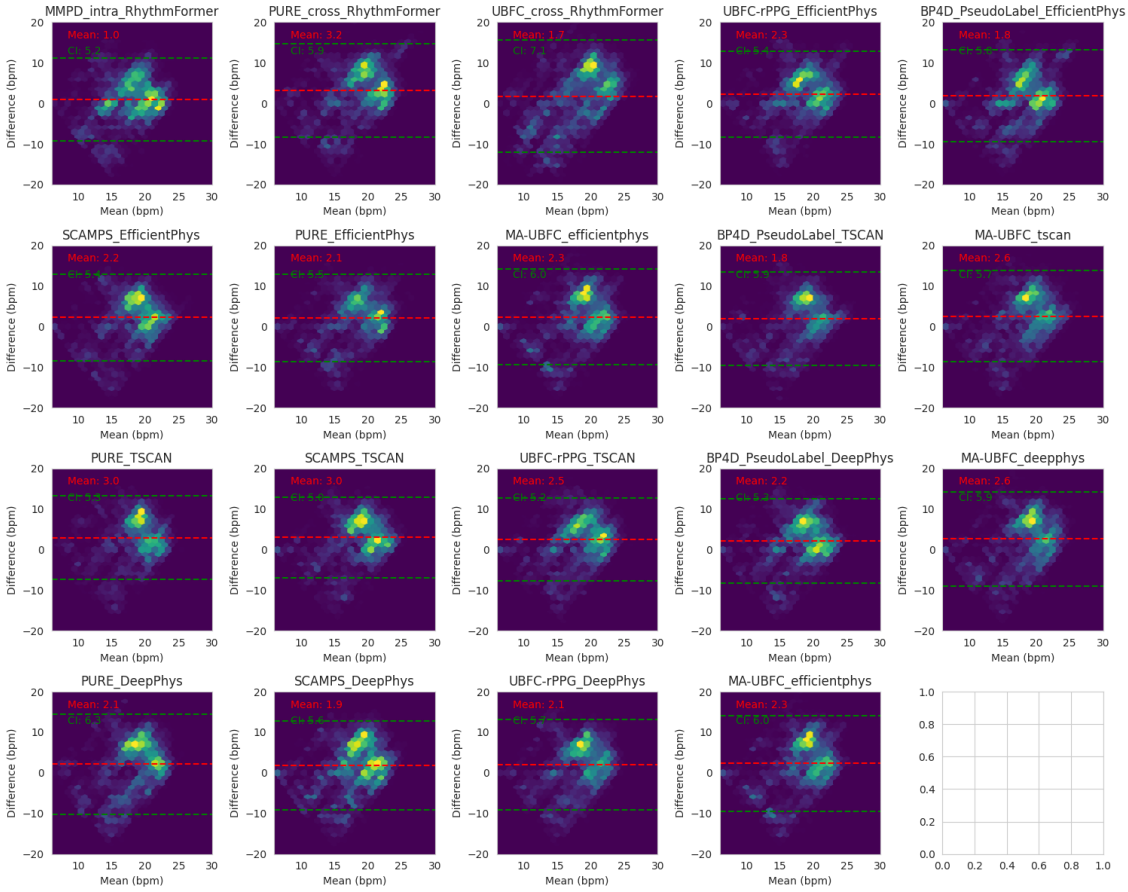


Figure A.8.: Bland-Altman plots of rPPG models

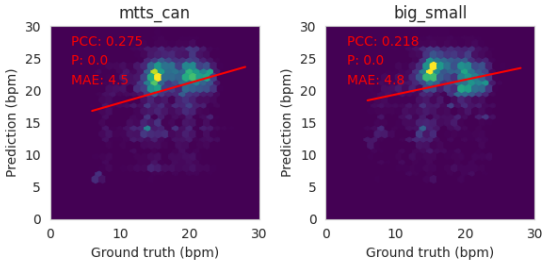


Figure A.9.: Correlation of the pre-trained respiration models

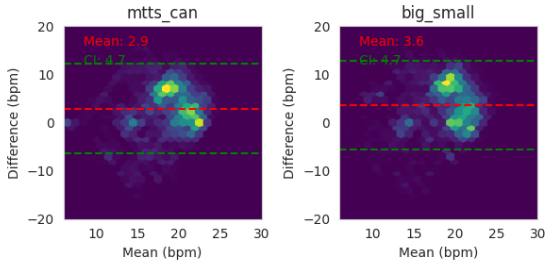


Figure A.10.: Comparison of pre-trained respiration model analyses

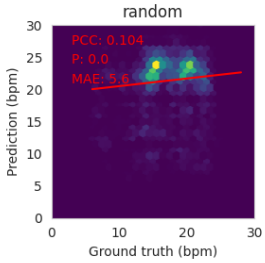


Figure A.11.: Correlation of the random reference model

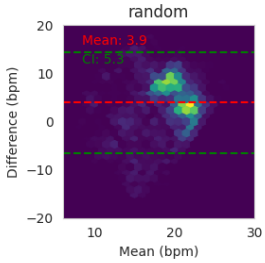


Figure A.12.: Bland-Altman plots of the random reference model

# Bibliography


[1] Fabian Braun et al. “Contactless Respiration Monitoring in Real-Time via a Video Camera”. In: *EMBEC & NBC 2017*. Ed. by Hannu Eskola et al. Singapore: Springer Singapore, 2018, pp. 567–570. ISBN: 978-981-10-5122-7.

[2] Rik Janssen et al. “Video-based respiration monitoring with automatic region of interest detection”. In: *Physiological measurement* 37.1 (2015), p. 100.

[3] Carlo Massaroni et al. “Contactless monitoring of breathing patterns and respiratory rate at the pit of the neck: A single camera approach”. In: *Journal of Sensors* 2018 (2018).

[4] Jorge Brieva, Hiram Ponce, and Ernesto Moya-Albor. “Non-Contact Breathing Rate Estimation Using Machine Learning with an Optimized Architecture”. In: *Mathematics* (2023). URL: https://api.semanticscholar.org/CorpusID:256438207.

[5] Hyeonsang Hwang and Eui Chul Lee. “Non-Contact Respiration Measurement Method Based on RGB Camera Using 1D Convolutional Neural Networks”. In: *Sensors (Basel, Switzerland)* 21 (2021). URL: https://api.semanticscholar.org/CorpusID:235229132.

[6] Kai Zhou. “Stress and Emotion Recognition based on Remote Photoplethysmography”. PhD thesis. Karlsruher Institut für Technologie (KIT), 2024. 194 pp. DOI: 10.5445/IR/1000171282.

[7] Petra Barthel et al. “Respiratory rate predicts outcome after acute myocardial infarction: a prospective cohort study”. In: *European Heart Journal* 34.22 (Dec. 2012), pp. 1644–1650. ISSN: 0195-668X. DOI: 10.1093/eurheartj/ehs420. eprint: https://academic.oup.com/eurheartj/article-pdf/34/22/1644/17895058/ehs420.pdf. URL: https://doi.org/10.1093/eurheartj/ehs420.

[8] Daniel McDuff. “Camera Measurement of Physiological Vital Signs”. In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: 10.1145/3558518. URL: https://doi.org/10.1145/3558518.

[9] Farah Q AL-Khalidi et al. “Respiration rate monitoring methods: A review”. In: *Pediatric pulmonology* 46.6 (2011), pp. 523–529.

[10] Carlo Massaroni et al. “Non-contact monitoring of breathing pattern and respiratory rate via RGB signal measurement”. In: *Sensors* 19.12 (2019), p. 2758.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org/. MIT Press, 2016.

[12] Maithra Raghu et al. *Do Vision Transformers See Like Convolutional Neural Networks?* 2022. arXiv: 2108.08810 [http://cs.cv/].

[13] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [http://cs.cv/].

[14] Berthold KP Horn and Brian G Schunck. "Determining optical flow". In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[15] Zachary Teed and Jia Deng. *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. 2020. arXiv: 2003.12039 [http://cs.cv/].

[16] Philipp Fischer et al. *FlowNet: Learning Optical Flow with Convolutional Networks*. 2015. arXiv: 1504.06852 [cs.CV]. URL: https://arxiv.org/abs/1504.06852.

[17] Eddy Ilg et al. *FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks*. 2016. arXiv: 1612.01925 [http://cs.cv/].

[18] Timon Blöcher et al. "VitalCamSet - a dataset for Photoplethysmography Imaging". In: *2019 IEEE Sensors Applications Symposium (SAS)*. 2019, pp. 1–6. DOI: 10.1109/SAS.2019.8705999.

[19] Jacob Benesty et al. "Pearson Correlation Coefficient". In: *Noise Reduction in Speech Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4. ISBN: 978-3-642-00296-0. DOI: 10.1007/978-3-642-00296-0_5. URL: https://doi.org/10.1007/978-3-642-00296-0_5.

[20] MIT Critical Data. *Secondary analysis of electronic health records*. Springer Nature, 2016.

[21] Jacob T VanderPlas. "Understanding the lomb–scargle periodogram". In: *The Astrophysical Journal Supplement Series* 236.1 (2018), p. 16.

[22] Kedar Khare, Mansi Butola, and Sunaina Rajora. *Fourier optics and computational imaging*. Springer, 2015.

[23] I.W. Selesnick and C.S. Burrus. "Generalized digital Butterworth filter design". In: *IEEE Transactions on Signal Processing* 46.6 (1998), pp. 1688–1694. DOI: 10.1109/78.678493.

[24] Marco AF Pimentel et al. "Toward a robust estimation of respiratory rate from pulse oximeters". In: *IEEE Transactions on Biomedical Engineering* 64.8 (2016), pp. 1914–1923.

[25] Axel Schäfer and Karl W Kratky. "Estimation of breathing rate from respiratory sinus arrhythmia: comparison of various methods". In: *Annals of Biomedical Engineering* 36 (2008), pp. 476–485.

[26] Robyn Maxwell et al. “Non-Contact Breathing Rate Detection Using Optical Flow”. In: (2023). DOI: 10.5281/ZENODO.8238518. URL: https://zenodo.org/record/8238518.

[27] Xudong Tan et al. “Lightweight Video-Based Respiration Rate Detection Algorithm: An Application Case on Intensive Care”. In: *IEEE Transactions on Multimedia* 26 (2024), pp. 1761–1775. DOI: 10.1109/TMM.2023.3286994.

[28] Xudong Tan et al. “Unobtrusive Respiratory Monitoring System for Intensive Care”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095831.

[29] Tomáš Lukáč, Jozef Púčik, and Lukáš Chrenko. “Contactless recognition of respiration phases using web camera”. In: *2014 24th International Conference Radioelektronika*. 2014, pp. 1–4. DOI: 10.1109/Radioelek.2014.6828427.

[30] Avishek Chatterjee, AP Prathosh, and Pragathi Praveena. “Real-time respiration rate measurement from thoracoabdominal movement with a consumer grade camera”. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2016, pp. 2708–2711.

[31] Wenjin Wang and Albertus C den Brinker. “Algorithmic insights of camera-based respiratory motion extraction”. In: *Physiological Measurement* 43.7 (2022), p. 075004.

[32] Peter H Charlton et al. “An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram”. In: *Physiological measurement* 37.4 (2016), p. 610.

[33] Mark Van Gastel, Sander Stuijk, and Gerard De Haan. “Robust respiration detection from remote photoplethysmography”. In: *Biomedical optics express* 7.12 (2016), pp. 4941–4957.

[34] Ferdous Karim Lucy et al. “Video based non-contact monitoring of respiratory rate and chest indrawing in children with pneumonia”. In: *Physiological Measurement* 42.10 (2021), p. 105017.

[35] Bruce Lucas and Takeo Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)”. In: vol. 81. Apr. 1981.

[36] Zimeng Liu et al. “Contactless Respiratory Rate Monitoring For ICU Patients Based On Unsupervised Learning”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2023), pp. 6005–6014. URL: https://api.semanticscholar.org/CorpusID:260908731.

[37] Christian Wiede et al. “Remote respiration rate determination in video data-vital parameter extraction based on optical flow and principal component analysis”. In: *International Conference on Computer Vision Theory and Applications*. Vol. 5. SCITEPRESS. 2017, pp. 326–333.

[38] Michael H Li, Azadeh Yadollahi, and Babak Taati. "A non-contact vision-based system for respiratory rate estimation". In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2014, pp. 2119–2122.

[39] Tianqi Guo, Qian Lin, and Jan Allebach. "Remote estimation of respiration rate by optical flow using convolutional neural networks". In: *Electronic Imaging* 33.8 (2021), pp. 267-1–267-1. DOI: 10.2352/ISSN.2470-1173.2021.8.IMAWM-267. URL: https://library.imaging.org/ei/articles/33/8/art00004.

[40] Weixuan Chen and Daniel McDuff. *DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks*. 2018. arXiv: 1805.07888 [http://cs.cv/].

[41] Xin Liu et al. *EfficientPhys: Enabling Simple, Fast and Accurate Camera-Based Vitals Measurement*. 2022. arXiv: 2110.04447 [http://cs.cv/].

[42] Xin Liu et al. *Multi-Task Temporal Shift Attention Networks for On-Device Contactless Vitals Measurement*. 2021. arXiv: 2006.03790 [eess.SP].

[43] Girish Narayanswamy et al. *BigSmall: Efficient Multi-Task Learning for Disparate Spatial and Temporal Physiological Measurements*. 2023. arXiv: 2303.11573 [cs.CV].

[44] Bochao Zou et al. *RhythmFormer: Extracting rPPG Signals Based on Hierarchical Temporal Periodic Transformer*. 2024. arXiv: 2402.12788 [http://cs.cv/]. URL: https://arxiv.org/abs/2402.12788.

[45] Xin Liu et al. *rPPG-Toolbox: Deep Remote PPG Toolbox*. 2023. arXiv: 2210.00716 [http://cs.cv/].

[46] Marc-André Fiedler, Micha Rapczyński, and Ayoub Al-Hamadi. "Fusion-based approach for respiratory rate recognition from facial video images". In: *IEEE Access* 8 (2020), pp. 130036–130047.

[47] Leila Mirmohamadsadeghi et al. "Real-time respiratory rate estimation using imaging photoplethysmography inter-beat intervals". In: *2016 Computing in Cardiology Conference (CinC)*. 2016, pp. 861–864.

[48] M. Mateu-Mateus et al. "A non-contact camera-based method for respiratory rhythm extraction". In: *Biomedical Signal Processing and Control* 66 (2021), p. 102443. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.102443. URL: https://www.sciencedirect.com/science/article/pii/S1746809421000409.

[49] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. *Better plain ViT baselines for ImageNet-1k*. 2022. arXiv: 2205.01580 [http://cs.cv/]. URL: https://arxiv.org/abs/2205.01580.

[50] Amirhossein Tavanaei et al. "Deep Learning in Spiking Neural Networks". In: *CoRR* abs/1804.08150 (2018). arXiv: 1804.08150. URL: http://arxiv.org/abs/1804.08150.

[51] Guillaume Heusch, André Anjos, and Sébastien Marcel. *A Reproducible Study on Remote Heart Rate Measurement*. 2017. arXiv: 1709.00962 [cs.CV].

[52] Ronny Stricker, Steffen Müller, and Horst-Michael Gross. "Non-contact video-based pulse rate measurement on a mobile service robot". In: *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. 2014, pp. 1056–1062. DOI: 10.1109/ROMAN.2014.6926392.

# List of Tables

3.1. Taxonomy of Models for Respiration Signal Extraction . . . . . 31

5.1. MEA, PCC and p-values of the best performing models . . . . . 60
5.2. Results of the Respiration-RhythmFormer model, with different combinations of loss function components . . . . . 66
5.3. T-scores for each loss component. A positive t-value indicates a positive effect on model performance, while a negative t-value indicates a negative influence. . . . . 66
5.4. Natural light vs normalized faces: Loss configuration and results . . 70

# List of Figures


2.1. Frequency spectrum of the preprocessed signal. . . . . 16
2.2. Counted peaks of the preprocessed signal. . . . . 17
2.3. Crossing points and the cross-curve signal of the preprocessed signal. 18
2.4. NFCP points and the cross-curve signal of the preprocessed signal. . 19

3.1. Feature points of the Lucas-Kanade algorithm in the chest area . . 24

4.1. DeepPhys and TS-CAN input frames . . . . . 39
4.2. EfficientPhys input frame . . . . . 39
4.3. BigSmall input frame . . . . . 40
4.4. MTTS-CAN input frame . . . . . 40
4.5. Effects of normalization and filtering on the raw signals. . . . . 42
4.6. Face and chest regions of interests . . . . . 44
4.7. Visualised optical flow between two frames . . . . . 44
4.8. Mean motion vectors in the chest area for each frame . . . . . 45
4.9. Comparison between the ground truth and the predicted breathing signal . . . . . 45
4.10. Comparison of frequency extraction methods: PSD, PK, CP, and NFCP. Brighter colors indicate higher density of predictions . . . . 47
4.11. Natural Lighting Scenario . . . . . 49
4.12. Normalized Face Scenario . . . . . 49

5.1. Correlation plots of the best performing models . . . . . 57
5.2. Bland-Altman plots of the best performing models . . . . . 58
5.3. MAE and Pearson correlation of the best performing models . . . . 59
5.4. MAE and Pearson correlation of the models . . . . . 61
5.5. T-Stats of the groups . . . . . 63
5.6. Respiration-RhythmFormer MAE and Pearson scores across different loss functions . . . . . 68
5.7. Respiration-RhythmFormer MAE and Pearson correlation grouped by setting . . . . . 69

A.1. Correlation of optical flow methods . . . . . 77
A.2. Bland-Altman plots of optical flow methods . . . . . 78
A.3. Correlation of Respiration-RhythmFormer models . . . . . 79
A.4. Bland-Altman plots of Respiration-RhythmFormer models . . . . . 80
A.5. Correlation of SimpleViT models . . . . . 80

A.6. Bland-Altman plots of SimpleViT models . . . . . 81
A.7. Correlation of rPPG models . . . . . 81
A.8. Bland-Altman plots of rPPG models . . . . . 82
A.9. Correlation of the pre-trained respiration models . . . . . 82
A.10.Comparison of pre-trained respiration model analyses . . . . . 83
A.11.Correlation of the random reference model . . . . . 83
A.12.Bland-Altman plots of the random reference model . . . . . 83