NrPPG-NNET: an End to End Deep Learning Approach to Remote Heart Rate Estimation Through Spatial Temporal Representation

Master Thesis Research Project

November 2021

Author:
Filippo M. Libardi

University Supervisors:
A. Telea, R. Poppe
Company Supervisors:
V. Tufano, F. De Nola, A. Letizia

Contents

Abstract

This research paper describes the development of a deep learning approach for remote non-contact heart rate estimation. The field of non-contact HR detection is vast, therefore all previously proposed methods are investigated and analysed. In addition, a previous work is selected to be extended and improved. The deep learning approach hereby presented is called NrPPG-NNET and was developed by applying the knowledge of a previously trained Convolutional Neural Network to the task of predicting HR from video footage alone. NrPPG-NNET achieves competitive results in terms of bpm Mean Absolute Error but does not reach the state of the art. However, it is significantly lighter than most of the previously proposed methods and can be run in real-time on mid-range laptops (even without GPU). Thanks to its speed and ease of use, NrPPG-NNET could potentially find application in the field of human-computer interaction. In addition, clues emerged during development that are important for future work and provide a solid foundation for further research.

Chapter 1

Introduction

1.1 Project Description and Motivation

Heart Rate (HR) assessment has a wide variety of uses, ranging from tracking fitness behaviours to detecting acute heart accidents. HR is usually measured by palpating the pulse or using specialised instruments such as pulse oximeters or electrocardiographs. Because of their close proximity to the subject, such touch sensors can accurately identify the underlying vital signs (e.g. Respiration Rate), and thus have applications in sensitive areas such as patient monitoring. However, the ability to remotely obtain such measurements through a camera/webcam - though with less precision - cannot only allow applications outside of the medical domain (for example, effective computing, human-computer interaction) but it can also increase accessibility to these measurements to individuals with sensitive skin or dermatitis.
All in all, vital signs monitoring is ubiquitous in clinical environments and emerging in home-based healthcare applications. Since modern techniques necessitate the use of uncomfortable sensors, a focus should be put on systems able to remotely detect clinical parameters.

This work goal is to study the algorithms already implemented in the field of remote clinical parameters estimation, and modify them in order to fit heterogeneous setups and test their performance.

The system proposed takes care of Human and Heart Rate detection. Through the implementation of a computer vision software, it is possible to identify a human subject within a given image (or frame). Additionally, an image segmentation process is performed and only a respiratory region of interest (RoI) is cropped out. From the latter, hearth (HR) is extrapolated in the form of BPM (beats per minute).

From the above, a question stems out:
Research Question: *Does video footage of a human convey enough graphical information to accurately estimate clinical parameters such as Heart Rate?*

This paper tries to answer this question by building an end to end system which receives images (or frames) as an input and it outputs a predicted BPM. We call this *NrPPG-NNET*. NrPPG-NNET does not only try to estimate the average BPM over a one minute window, but is able to predict the BPM every 10 seconds instead. In several areas, such as health monitoring and emotion detection, a more detailed HR prediction is much needed, and this is what motivates this research.

1.2 Terminology and Taxonomy

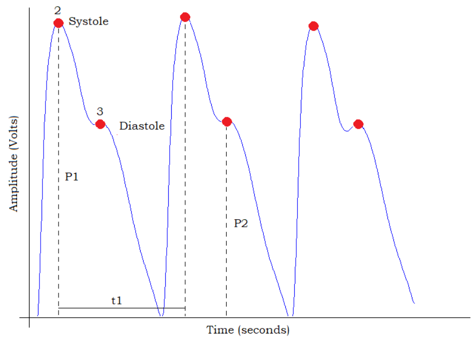


Figure 1.1: Example of an PPG signal from Moraes et al. 2018.

This paper will occasionally present some terms the reader may not be aware of - hence the coming section aims at giving an overview of these terminologies and their meaning. By observing small variations in skin tone, visual HR prediction techniques calculate the heart rate (HR). These variations are thought to be caused by blood circulation in the peripheral arteries, hence the examined signal being the "blood volume pulse" (BVP) sensed through plethysmographic data.

Plethysmography has traditionally been used to study peripheral circulation. From the plethysmographic signal (shown in Figure 1.1), besides BVP, is possible to derive Heart Rate, Respiratory Rate, SpO2 (measurement of blood oxygen) and other clinical parameters.

An Electrocardiographic (ECG) signal represents the voltage of the electrical activity of the body. In order to acquire such signal, electrodes placed on the skin are used to measure the voltage versus duration of the heart's electrical operation. During each cardiac cycle, these electrodes sense the minor electrical changes that occur as a result of cardiac muscle depolarization and repolarization (i.e. a heartbeat).

Several heart irregularities, such as cardiac rhythm disruptions (including atrial fibrillation and ventricular tachycardia), insufficient coronary artery blood supply and electrolyte disturbances, induce variations in the normal ECG sequence. An example of a segmented ECG signal can be seen in Figure 1.2. As shown in the figure, different waves appear in the signal. R complex is the actual beat, whereas the other peaks represent variations for the pre and post beat, and are labelled accordingly. This work will often refer to these parts of the signal. In reading literature on Remote Photoplethysmography, it becomes evident that confusion is made between terms used to define this concept and other affiliating ones. The taxonomy proposed by Špetlík 2018 is adopted, as it is based on a clear distinction between the approaches.

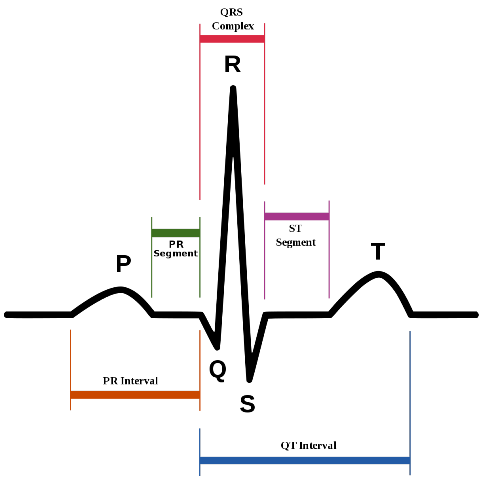


Figure 1.2: Example of an ECG signal (source).

Photoplethysmography (PPG) is an elementary optical procedure used to recognize volumetric changes in the blood at a peripheral level. It is a minimal-effort and non-obtrusive strategy that makes estimations based merely on the surface of the skin.
The estimation can be done remotely (the sensor is not touching the skin) or in contact with the subject. *Non contact Photoplethysmography* will be called *NrPPG*. *rPPG* alone refers to *Reflectance Photoplethysmography*, whereas *tPPG* refers to methods making use of *Transmittance Photoplethysmography*.

The distinction between these two deals with the way light changes when the skin is captured.

More formally, in PPG, a volume of peripheral tissues is lit by optical radiation that goes through numerous occasions of dissipation and assimilation as it navigates through various tissue layers. Lastly, it is transmitted through (transmittance) or reflected from (reflectance) the tissue volume - Barun and Ivanov 2009.

Both rPPG and tPPG can be remote or contact based, as this research only focuses on remotely operated techniques, it suffices to define not-contact reflective PPG as NrPPG.
A PPG strategy may aim for a blood volume imaging or a blood volume signal (BVS) recreation. Both the blood volume imaging and BVS allude to the assessed quality, i.e. the volume of blood going through the tissue.
Heart Rate (HR) might be assessed from the BVS, for example by including the number of peaks in a given time-spanned signal. HR captured from the BVS is at times called the "blood volume pulse". Visual HR assessment is an NrPPG technique performing blood volume pulse estimation - and it is what this research is about.

1.3 Project Embedding

This project has been developed as an internship work within a company, namely Teoresi Group. Quoting from the company's website, *Teoresi is high profile engineering. It offers an innovative approach in close synergy with the Research & Development departments of the main industrial players. Teoresi provides turn-key solutions accelerating the customer's time-to-market.* This paper has been developed while in constant contact with Teoresi's research department. The work is not towards the realisation of a specific (market ready) product, but is rather mostly focused on innovative methodologies.

Possibly, the end product is based on a previously built method and tries to enhance it by modifying it in order to yield better results. Given that the product won't need to be marketed, there are a few requirements that need to be matched - and these are listed in Section 2.3.
Generally, the field of NrPPG is still at a very early stage with very few companies claiming to have a market/clinic ready product. Given that the aim is to detect clinical parameters, the error margin for which the method can be allowed needs to be extremely small as there is no space for unconfident decisions.
Interestingly, it has been noted that methods producing a *fairly good* pulse detection can still be used in the field of HCI (Human-Computer Interaction). Here these systems can detect (trough the user webcam) the subject behaviour on a clinical level (e.g. heartbeat) to infer the software or visualisation quality based on the subject physiological reaction.

1.3.1 Contacts

This project is supervised by three company employees:
Francesco De Nola, Technology Leader
Vincenza Tufano, Junior Engineer
Annalisa Letizia, Junior Data Scientist

1.4 Thesis structure

The paper is developed in 6 chapters - each with related sections and subsections. For clarity sake, a detailed explanation is listed below.

Section 1.2 clarifies what terms are adopted and their definitions, explaining the notation consistently used throughout this work. Section 1.3 provides an overview of the company this work is being developed with.

Chapter 2 describes both the work previously done in the field (with the necessary conclusions) and also deals with how these methods have been evaluated. More specifically, Section 2.2 delves into the description of methods thus far developed in the field and tries to list a preliminary set of requirements needed towards its realisation. These helped shaping the chosen method as a core idea for the project itself. Section 2.1 expands on things such as metrics usually proposed in the NrPPG field.

Furthermore, Chapter 3 analyses the data at hand in more details as well as evaluating appropriate methods for some steps of the pipeline (e.g. which face detector). The information provided in this chapter is propaedeutic to a better and smoother development as well as a clearer understanding of final results.

Chapter 4 details the approach taken for developing this research project. This means both objectively describing the procedures employed as well as critically assessing every possible downsides. This chapter also explores the architecture and the inner functioning of the system itself, addressing both how the input has been pre-processed and how the model interprets and learns from it in order to give an accurate prediction.

Chapter 5 delves into analysing the results obtained as well as benchmarking the method with other proposed approaches and some different versions of the method itself. Lastly, in Chapter 6 results are discussed and possible improvements assessed accordingly.

In Section 7.1, a visual representation of the timeline for the project is proposed. The real-time development is then discussed accordingly.

Finally in Section 7.2, a list of all the abbreviations used in this research is given along with their meanings.

Chapter 2

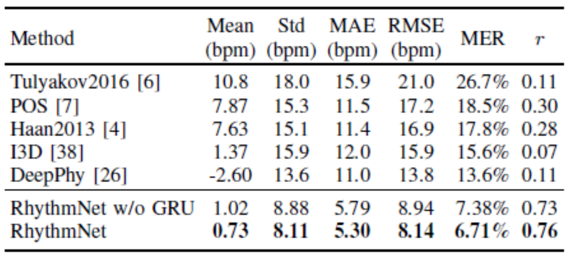
Related Work

The aim of this chapter is manifold. Firstly, Section 2.1 will explain the most common metrics used in the NrPPG field to evaluate methods. Secondly, Section 2.2 will give an overview of the method so far developed in the field of NrPPG detection. Rather than being just a list of papers studied, each paper explanation will highlight the pros and cons - and even issues - faced when trying an implementation of the approach proposed by the authors. Thirdly, Section 2.3 will list the requirements that should be fulfilled by the method chosen as a baseline for this research project. Lastly, Section 2.4 will rank the methods accordingly, so to find the best candidate and also explain the reason leading to such conclusion.

2.1 Evaluation Methodologies

This will describe what metrics have been developed to serve the purpose of comparing and evaluating methods in the NrPPG field.

Since Verkruysse, Svaasand, and Nelson 2008 introduced the very first method that made it possible to estimate HR merely using light reflectance, a whole field has been developed around this notion. At first, authors suffered from great discrepancies in how they presented their work. These could be found in how methods were evaluated in terms of surrounding conditions, subject motion, subject skin type and general experiment setup.



Method	Mean (bpm)	Std (bpm)	MAE (bpm)	RMSE (bpm)	MER	r
Tulyakov2016 [6]	10.8	18.0	15.9	21.0	26.7%	0.11
POS [7]	7.87	15.3	11.5	17.2	18.5%	0.30
Haan2013 [4]	7.63	15.1	11.4	16.9	17.8%	0.28
I3D [38]	1.37	15.9	12.0	15.9	15.6%	0.07
DeepPhy [26]	-2.60	13.6	11.0	13.8	13.6%	0.11
RhythmNet w/o GRU	1.02	8.88	5.79	8.94	7.38%	0.73
RhythmNet	**0.73**	**8.11**	**5.30**	**8.14**	**6.71%**	**0.76**

Figure 2.1: Results table for RhythmNet, borrowed from Niu, Shan, et al. 2020 as a showcase of a typical results depiction in the NrPPG field.

It was only in between 2005 and 2010 that researchers started to align their procedures to mutual standards. Mostly, these standards concerned the metric with which methods have been evaluated. This definitely aided the comparison between different methods as it unified the results obtained by different authors. These metrics are also the ones used in this research, so to conform to the standards proposed. Whereas during training, all the mentioned metrics have been studied, the final results will only be presented in terms of MAE. A classic results table for NrPPG methods can be seen in Figure 2.1. Besides the main metrics, some authors introduced more of their own, but always as an addition to the main ones listed below.

1. **MAE**: Mean Absolute Error (MAE) is often also known as L1 Loss, and mathematically represents the mean of distance between the predicted and actual values. As it is a distance, there is no negative value, and it follows the absolute value operator. Alternative formulations may include relative frequencies as weighting factors. The scale for the mean absolute error needs to be the same as for the calculated results. This is referred to as a scale-dependent precision metric, meaning that it can't be used to compare series of different sizes. Mean absolute error is a common measure of prediction error in time series analysis. Referring to the aforementioned notation the MAE can be expressed as:

$$\mathrm{MAE} = \frac{1}{N}\sum_t \mid \hat{h}(t) - h(t) \mid \tag{2.1}$$

   where $\hat{h}(t)$ and $h(t)$ represent the predicted and the actual values.

2. **RMSE**: The Root-Mean-Square Error (RMSE) represents a disparity between two variables as the square root of the average of squared differences. RMSE represents the sample standard deviation of the absolute difference between reference and measurement, i.e. smaller RMSE suggests more accurate extraction. Deviations here are squared to prevent positive and negative values from cancelling each other out. With this measure, larger value errors are also amplified - a feature that can facilitate the elimination of methods with the most significant errors. It is formally noted as below:

$$\mathrm{RMSE} = \frac{1}{N}\sqrt{\sum_t (\hat{h}(t) - h(t))^2} \tag{2.2}$$

3. **PCC**: Pearson Correlation Coefficient (PCC) represents the correlation between the estimate $\hat{h}(t)$ and the ground truth $h(t)$. The value of this (and others) correlation coefficient varies between -1 and +1, where both extreme values represent perfect relationships between the data, while 0 represents the absence of relation. This is true as long as we consider linear relationships. A positive relationship means that data points that obtain high values in one variable tend to obtain high values in the second variable. And this works the other way around too, where those who have low values on one variable tend to have low values on the second variable. Conversely, a negative relationship means that low scores on one variable correspond to high scores on the other variable. Formally PCC is expressed as below and is often abbreviated to r:

$$\mathrm{PCC} = \frac{\sum_t (\hat{h}(t) - \hat{\mu})(h(t) - \mu)}{\sqrt{\sum_t (\hat{h}(t) - \hat{\mu})^2}\sqrt{\sum_t (\hat{h}(t) - \mu)^2}} \tag{2.3}$$

   Where $\hat{\mu}$ and μ denote the means of the respective signals.

The above metrics are the ones commonly used when evaluating NrPPG methods and among have been chosen to create a mutual ground for comparison.

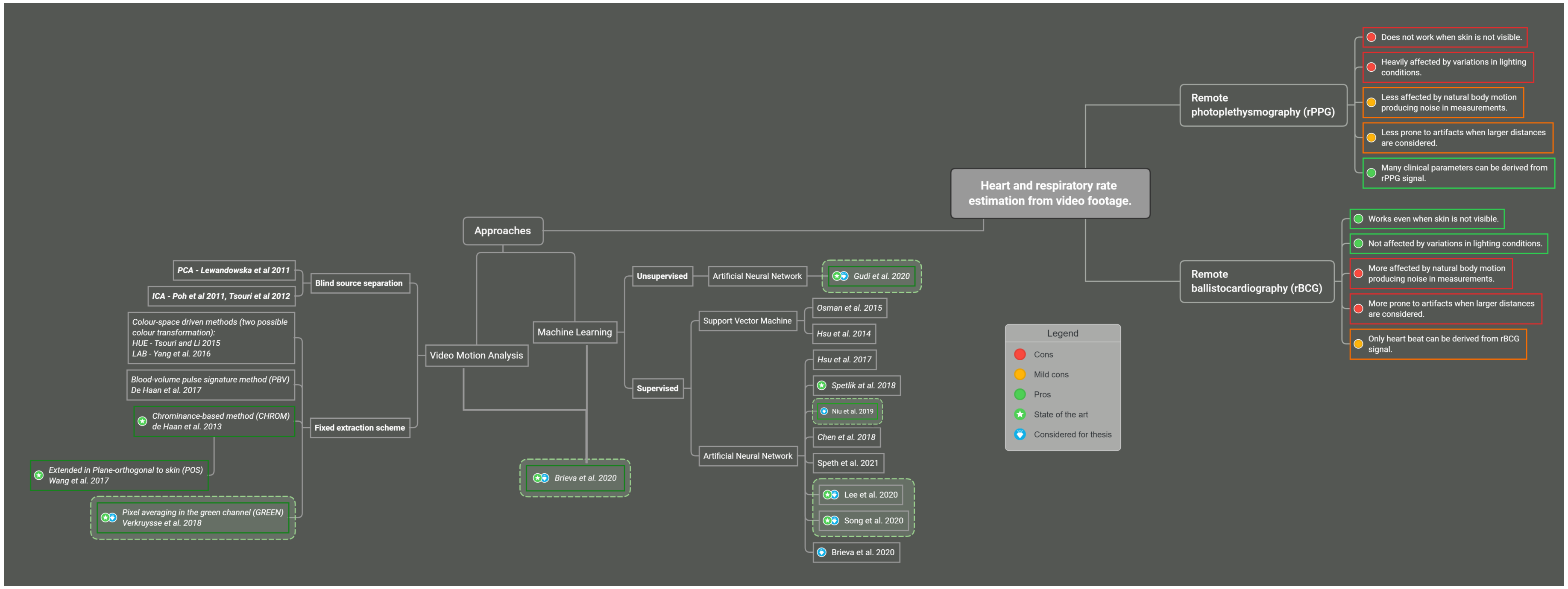

Heart and respiratory rate estimation from video footage.
Remote photoplethysmography (rPPG)
Does not work when skin is not visible.
Heavily affected by variations in lighting conditions.
Less affected by natural body motion producing noise in measurements.
Less prone to artifacts when larger distances are considered.
Many clinical parameters can be derived from rPPG signal.
Remote ballistocardiography (rBCG)
Works even when skin is not visible.
Not affected by variations in lighting conditions.
More affected by natural body motion producing noise in measurements.
More prone to artifacts when larger distances are considered.
Only heart beat can be derived from rBCG signal.
Legend
Cons
Mild cons
Pros
State of the art
Considered for thesis
Approaches
Video Motion Analysis
Blind source separation
PCA - Lewandowska et al 2011
ICA - Poh et al 2011, Tsouri et al 2012
Fixed extraction scheme
Colour-space driven methods (two possible colour transformation):
HUE - Tsouri and Li 2015
LAB - Yang et al. 2016
Blood-volume pulse signature method (PBV)
De Haan et al. 2017
Chrominance-based method (CHROM)
de Haan et al. 2013
Extended in Plane-orthogonal to skin (POS)
Wang et al. 2017
Pixel averaging in the green channel (GREEN)
Verkruysse et al. 2018
Machine Learning
Brieva et al. 2020
Unsupervised
Artificial Neural Network
Gudi et al. 2020
Supervised
Support Vector Machine
Osman et al. 2015
Hsu et al. 2014
Artificial Neural Network
Hsu et al. 2017
Spetlik at al. 2018
Niu et al. 2019
Chen et al. 2018
Speth et al. 2021
Lee et al. 2020
Song et al. 2020
Brieva et al. 2020

Figure 2.2: An overview of the papers reviewed so far. Some of them are marked as *considered for thesis*, these are the ones I am investigating in further details.

2.2 Methods

This section provides an overview of the work previously done in the NrPPG field. Many different approaches have been taken, ranging from employing computer vision algorithms to the use of (supervised ans unsupervised) deep learning techniques. A summary of this comprehensive literature review is hereby proposed, where the most interesting methods taken in consideration for this research have been marked accordingly.

First of all, the main distinction can be made on the approaches taken so far. In fact, researchers either took a "deterministic approach" (meaning they employed algorithms that, when given an input, will always return the same output) or a Machine Learning (ML) approach. Some might argue that neural networks can be considered deterministic in some sense, and this would be partially correct.

The reason being that once a Neural Network's weights and structure are fixed (say, after it has been trained), it does become a deterministic function. The training technique, on the other hand, is not always deterministic: it typically employs some form of a stochastic gradient descent method; it almost certainly employs some random weight initialization; it may exhibit some chaotic behavior due to nonlinearities, and so on. As a result, it is possible that the same training technique used for the same dataset creates two distinct neural networks - despite the fact that they're intended to function similarly in terms of training criterion. Each of the two networks would be a deterministic function in this scenario, but the training method would be stochastic.

While more deterministic methods are reported in Figure 2.2, this project employs a supervised ML approach.

In the next paragraphs, I will elaborate on some of the considered Machine Learning methods and illustrate the reasons why some of the approaches are not suitable for this specific case - and why some others might be.

2.2.1 Janssen et al. 2016, 'Video-based respiration monitoring with an automatic region of interest detection.

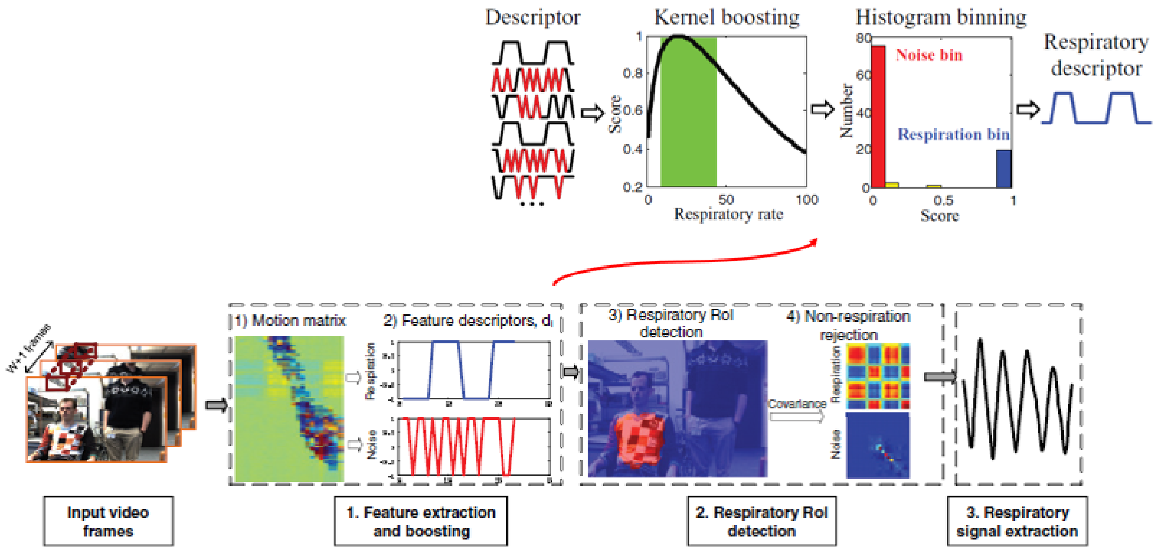


Figure 2.3: The pipeline of the work presented in Janssen et al. 2016, with a focus on the feature extraction bit of the process.

The method (illustrated in Figure 2.3) is outdated and the results are not comparable with more recent methods. That said, the focus put on the relevance of the ROI (Region Of Interest) is still worth mentioning. In fact, the method contribution is twofold. Firstly, it proposes a robust feature representation for respiratory signal based on motion features, which exploits the intrinsic properties of respiration. Secondly, it enables the automatic respiratory ROI detection to enhance the monitoring performance. Once again, many methods make use of automatic ROI detection. The ROI detection topic has first been investigated by the authors of this work. It is very important to determine what region of interest the model should be looking at or, in other words, which region contains the most PPG signal information. Having said this, the method proves to be effective only on the dataset created ad-hoc for this research, and it shows results in metrics that are not the suggested ones for this field. Additionally, many methods outperformed the results here reported since the publication. Overall, it suggested that certain face regions - such as forehead and cheeks - are the most relevant to perform a convolution (i.e. these areas are the ones that convey the most information in terms of clinical parameter estimation).

2.2.2 Špetlík 2018, 'Visual heart rate estimation with convolutional neural network'.

This method has been researched to the point of trying to replicate the work. A convolutional neural network (denoted as HR-CNN) is proposed to approximate a subject's heart rate (HR) in a video series. The network's input is a time-lapse series of frames of a subject's face. A single scalar – projected HR – is the output. The network is comprised of two parts: the extractor and the HR estimator. The extractor transforms an image into a single digit. A sequence of scalar outputs (i.e. the PPG signal) is produced by running the extractor over a series of images. The HR estimator receives the PPG signal and emits the HR. The two components are trained separately. First, the extractor is trained to increase the signal-to-noise ratio (SNR) when given the real heart rate. The estimator is then trained to minimize the mean absolute error between the approximate and real HR. The real innovation introduced by this method is in the loss function of the first network.

Let $\mathcal{T} = \left\{ \left(\mathbf{x}_1^j, \ldots, \mathbf{x}_N^j, f^j \right) \in \mathcal{X}^N \times \mathcal{F} \mid j = 1, \ldots, l \right\}$ be our training set comprised of l sequences of N facial RGB image frames $x \in X$ and their corresponding HR labels $f \in F$. The symbol X denotes a set of all input images and F is a set of all sequence labels, i.e. the true HR frequencies measured in Hertz. Additionally, the output of the first CNN for the nth image is noted as $h\left(\mathbf{x}_n; \mathbf{\Phi}\right)$, where ϕ is a concatenation of all convolutional filter parameters.

$$\mathrm{PSD}(\hat{f}, \mathbf{X}; \mathbf{\Phi}) = \left(\sum_{n=0}^{N-1} h\left(\mathbf{x}_n; \mathbf{\Phi}\right) \cdot \cos\left(2\pi \hat{f} \frac{n}{f_s}\right) \right)^2 + \left(\sum_{n=0}^{N-1} h\left(\mathbf{x}_n; \mathbf{\Phi}\right) \cdot \sin\left(2\pi \hat{f} \frac{n}{f_s}\right) \right)^2 \tag{2.4}$$

$$\mathrm{SNR}(f, \mathbf{X}; \mathbf{\Phi}) = 10 \cdot \log_{10} \left(\sum_{\hat{f} \in \mathcal{F}^+} \mathrm{PSD}(\hat{f}, \mathbf{X}; \mathbf{\Phi}) / \sum_{\hat{f} \in \mathcal{F} \setminus \mathcal{F}^+} \mathrm{PSD}(\hat{f}, \mathbf{X}; \mathbf{\Phi}) \right) \tag{2.5}$$

$$\ell(\mathcal{T}; \mathbf{\Phi}) = -\frac{1}{l} \sum_{j=1}^{l} \mathrm{SNR}\left(f^j, \mathbf{X}^j; \mathbf{\Phi}\right) \tag{2.6}$$

A signal to noise ratio function is used between the power spectral densities of the signal to assess the efficiency of the derived signal. More intuitively in Equation (2.5) the numerator captures the strength of the true HR signal frequency, whereas the denominator represents the energy of the background noise, excluding the tolerance interval. The latter accounts for the true HR uncertainty (e.g. due to HR non-stationarity within the sequence). The resulting loss function (i.e. Equation (2.6)) seemed to make a lot of sense, so this method was cherry-picked as one of the most valid ones.

Though when further researching, some incongruities started to raise - especially between the full PhD thesis (from which the paper derives) and the paper itself. In fact, not only the loss function was completely changed for no apparent reason whatsoever, but many pitfalls were also discovered in development (requires more than 12GB of VRAM, the input for the second model is not as explained in the paper and more). The one great contribution the authors indeed took towards the research of this field is the creation of a massive and uncompressed dataset (tens of TB). The latter one was accessible upon request, and it stands out in the field as subject motion and light changes are extremely emphasized - to the point where experiments are performed during fitness training procedures.
In conclusion, the method has been outperformed by more recent ones. Given the many discrepancies between what has been written and done, it is plausible that researchers have not reported the method thoroughly - hence making it un-reproducible. The methodology reported still carried great addition to the thought process of the end system.

2.2.3 Yu, Peng, et al. 2019, 'Remote Heart Rate Measurement from Highly Compressed Facial Videos: an End-to-end Deep Learning Solution with Video Enhancement'.

During the exploration of this field, it has become clear that the main issue with obtaining an accurate prediction of PPG signal from a video is the video quality itself - for quality is not just the stability and definition of the camera, but rather the compression techniques used in the process. This issue is well discussed in both Mironenko et al. 2020 and McDuff, Blackford, and Estepp 2017. Video compression resulted to be a determinant factor towards the accuracy of the prediction. Firstly, let's assess compression quickly. Data compression is a way of minimizing data size without losing information. Data compression methods are divided into two categories: lossy compression and lossless compression. The key distinction between the two compression methods is that lossy compression does not recover data in its original state after decompression, while lossless compression does. The aim of lossy compression is to reduce the number of bits absorbed by an image while ensuring that the gaps between the initial (uncompressed) image and the restored image are not perceptible to the human eye — or at least not objectionable. In practice and for our specific case, models trained and/or validated on uncompressed videos yielded way better performance.

From the above, it follows the author intuition of preprocessing the video to enhance the quality of this. The approach taken is to employ an initial Spatial-Temporal Video Enhancement Networks (STVEN) that parses video frames and enhances their quality. Secondly, the video is used as input for rPPGNet, composed by a spatial-temporal convolutional network (a skin-based attention module and a partition constraint module). As previously mentioned, separate skin areas have differing degrees of blood vessel density as well as biophysical parameter maps (melanin and haemoglobin), thus contribute to different levels for rPPG signal measurement. The Skin Segmentation and Attention mechanism determine which skin regions are the most important to examine and could help the most in generating a clean PPG signal. Skin-based attention aids in the adaptive selection of skin areas. Whereas via the partition constraint, the output signal is taken and processed in order for the model to focus more on the rPPG signals instead of interference. Additionally, a major role is played by loss functions of the spatial-temporal NN (reported on the rightmost side of Figure 2.4).

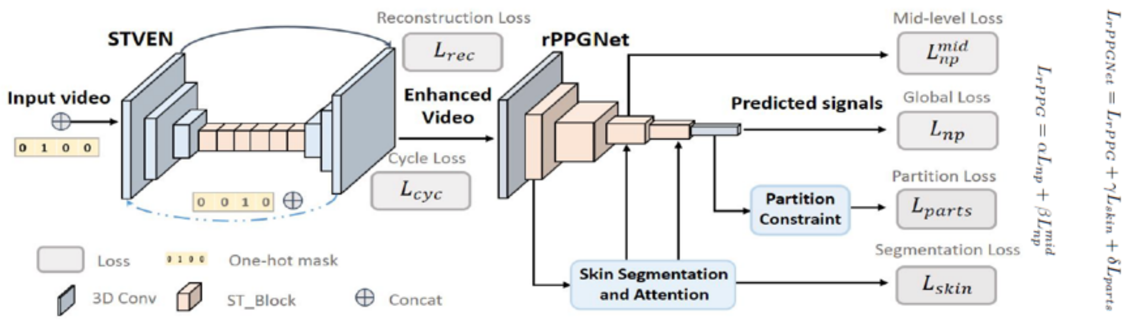


Figure 2.4: Models architecture, overall pipeline and loss formulae from Yu, Peng, et al. 2019.

There are two main takeaways: the stress the authors put on the quality of video compression, and the engineering behind the loss functions. Additionally, it is worth to notice that the authors are among the first ones to have used Pearson correlation as a loss function. This is to minimise the linear similarity error instead of the point-wise intensity error.

Overall, this method presents little if no issues. Nevertheless, it is worth mentioning that the model itself is quite *deep* and might prove to be resource hungry. The two main reasons this was not taken in consideration are: the final model will ideally be ran on an embedded device, and the lack of a computational power for training the model in the first place.

2.2.4 Niu, Shan, et al. 2020, 'RhythmNet: End-to-end Heart Rate Estimation from Face via Spatial-temporal Representation'

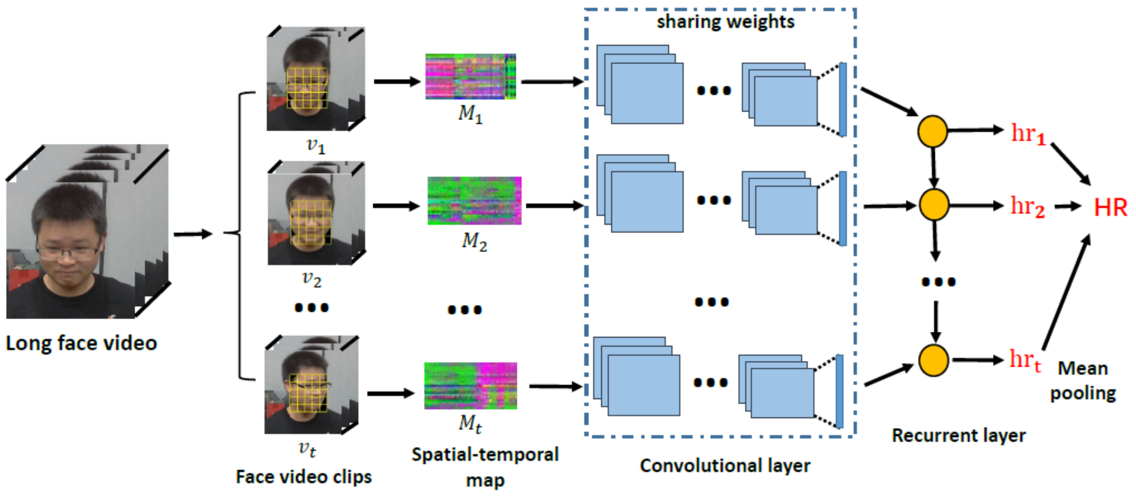


Figure 2.5: Overall method pipeline from Niu, Shan, et al. 2020

RhythmNet is a very promising method that focuses more on the processing of the input rather than the complexity of the model itself. RyhthmNet uses a spatial-temporal representation as to its input, decoding HR signals from different ROI volumes. The spatial-temporal representations are fed into a convolutional network for HR estimation. The Gated Recurrent Unit (GRU) accounts for the relationship of neighbouring HR measurements from a video series, resulting in effective HR calculation. The overall pipeline of the method can be seen in Figure 2.5. There are two things that need to be explained in order to understand why this method stands out. First, let's take a look at the pre-processing pipeline applied to an input video sequence, which is turned into a so-called spatial-temporal feature map.

Given an input of frames depicting a person, the subjects' faces are first aligned with each frame in respect of the viewpoint - where the alignment is based on face landmarks. The output is projected in YUV coordinates (an alternative colour space). The Y'UV model defines a colour space using one luminance component (Y') and two chrominance components (UV). The face (transformed and aligned) is then divided into N blocks of interest, the average colour value is taken for each YUV colour channel for each block. The same averages in the same block but in different frames are concatenated to form a sequence. The various sequences are allocated one on top of the other to form a space-time map ($T \times n \times c$, i.e. n sequences of T frames for c colours). The pipeline described above is best pictured in Figure 2.6.

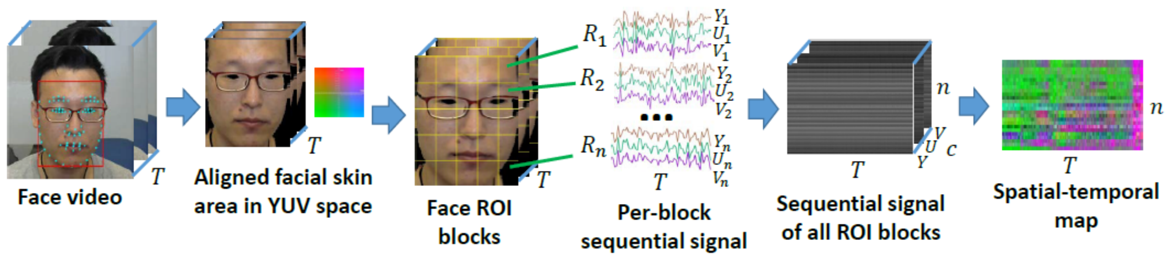


Figure 2.6: Zooming in onto the spatial-temporal maps creation, from Niu, Shan, et al. 2020

It is interesting to note that this method is one of the few using some kind of Recurrent Neural Network (RNN). These are a class of artificial neural network that includes neurons connected together in a loop. Usually, the output values of a higher layer are used as an input of a lower layer in these kind of architectures. This interconnection between layers allows the use of one of the layers as a state memory and to model a temporal dynamic behaviour dependent on the information received at previous time instants, by providing a temporal sequence of values as an input. Generally, the peculiarity of Recurrent Neural Network is the ability to work on sequences of arbitrary length, overcoming the limitations imposed by other structures, such as Convolutional Neural Network (CNN) that require an input of fixed length.

As shown in Figure 2.5 , the map created from video clips is fed as an input to a convolutional network (in this case ResNet18, from He et al. 2015). Here convolution is applied to each map conveying all the information pertinent to a given temporal window. The sequence of convoluted maps is then fed to a Gated Recurrent Unit to better model the time dimension of the input. At last, a mean HR within the T continuous measurements (output of the recurrent unit) is computed.

Overall, this method presents: an innovative formulation of the input; an efficient modelling of the temporal dimension; it achieves promising results on both the public-domain and the created VIPL-HR estimation databases; and it does work in real-time (given that the computational cost is quite low). Drawbacks could be pinned down in terms of the output and code availability. The former concerning the possibility of outputting more HR values rather than one minute mean HR, could be solved by trying to remove the last averaging layer and reducing the time of each time window (e.g. making it less than 10 seconds). Technically one could think of removing the final averaging layer so to obtain a discrete value for each time window. In terms of code availability, this issue is shared among all the methods presented, as none really gives out a replication of the method in terms of a packaged application - though the implementation does seem feasible for what described in the paper.

2.2.5 Brieva, Ponce, and Moya-Albor 2020, 'A contactless respiratory rate estimation method using a Hermite magnification technique and convolutional neural networks'.

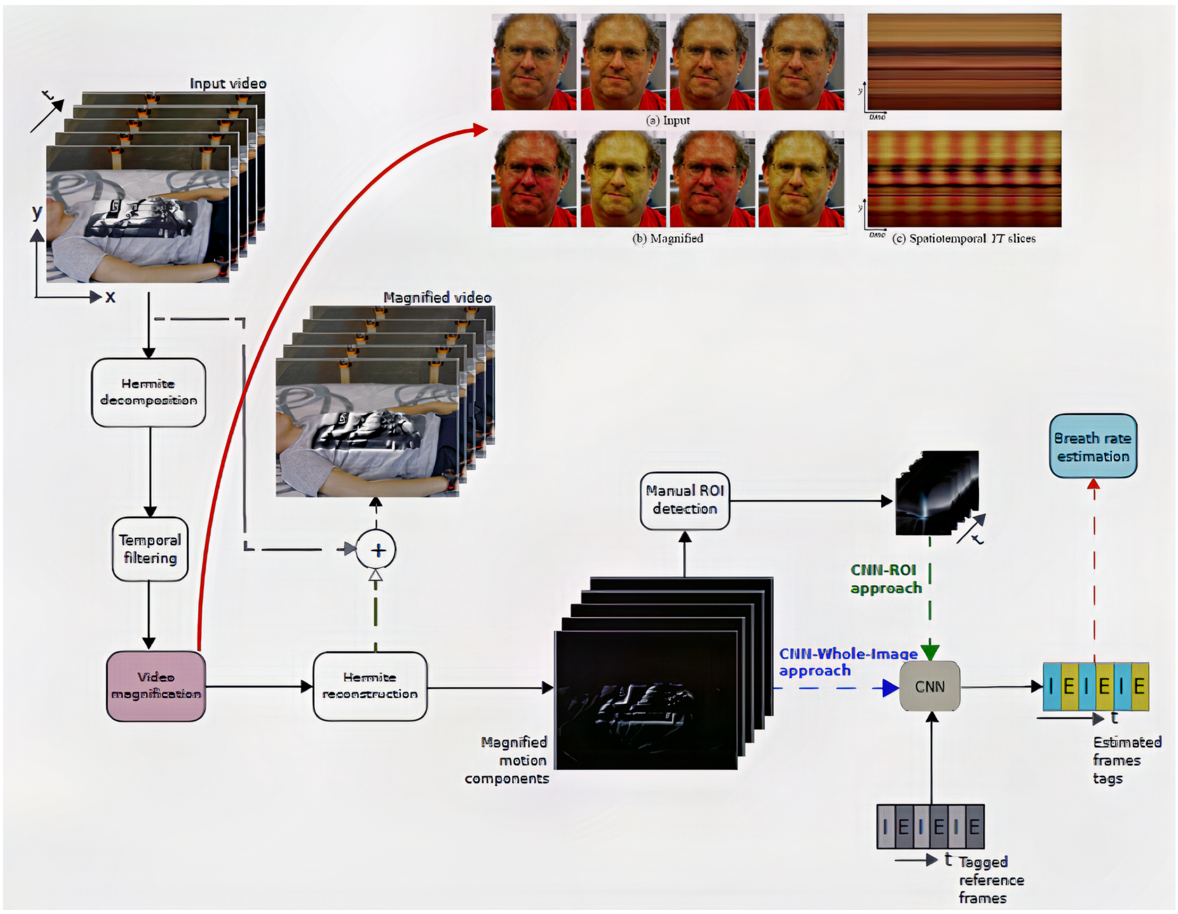


Figure 2.7: The pipeline for method explained in Brieva, Ponce, and Moya-Albor 2020, additionally an example of what the Hermite video magnification techniques is illustrated in the top right.

This method is very innovative. It is among the first ones that tried to borrow from previously developed deterministic methods and it uses the conclusions drawn from these to improve the training of simple Convolutional Network. More specifically, it exploits the inferred knowledge that colour channels of the pixels where the skin is depicted retain most of the information useful towards the detection of HR. An instance of the above-mentioned knowledge is the green channel of the RGB spectrum reflecting light changes in a machine perceptible way.
The pipeline followed by the researchers (depicted in Figure 2.7) is as follows:

- Carrying out a spatial decomposition of the image sequence using Hermite transform (first introduced in Martens 1990).
- Performing temporal filtering of the spatial decomposition to retain the motion components.

- Amplifying the different spatial frequency bands by a given factor α.
- Reconstructing the magnified motion components through an inverse spatial decomposition process.
- Adding the reconstructed magnified motion components to the original image sequence.
- Outputs may be cropped depending on ROI and fed to a NN which binarily classifies each frame.

The only issue one could see with this method is that the out prediction of the Neural Network is a binary classification of each frame where Convolution is performed. The classification is between E(xhaling) and I(nhaling). This surely helps to detect the respiratory rate, but much more work should be added when trying to reconstruct a BPM signal. Eventually, the end system aimed to be built should be able to output HR values every 10 - or so - seconds (and not just per minute averaged HR and RR values).
In conclusion, the method is valid and the magnification technique employed (i.e. Hermite) might still be relevant if coupled with another methodology.

2.2.6 Song et al. 2020, 'PulseGAN: Learning to generate realistic pulse waveforms in remote photoplethysmography'.

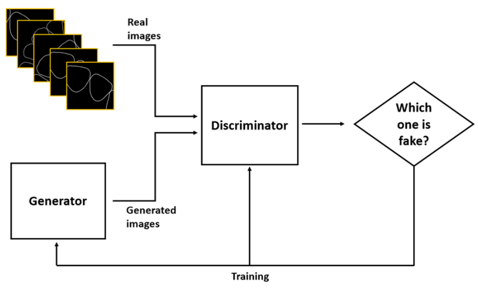


Figure 2.8: Example of a GAN architecture.

This method acts as a supplement to a pipeline that already determines the plethysmographic signal and exploits the discriminative power of a generative adversarial network (GAN).

GANs are a clever way to train a generative model, framing the problem as a supervised learning problem with two sub-models: the generator model that is trained to generate new examples, and the discriminator model that tries to classify examples as real (from the domain) or false (generated) - a classical GAN architecture can be seen in Figure 2.8.
Both networks try to play their competing roles. When the discriminator detects false data, it returns the data. In this case, the generative network is not yet effective enough and therefore must continue to learn. At the same time, however, the discriminator has also learned. The generator tries to create data that looks so real that it can be categorized as such by the discriminator. The discriminator, on the other hand, tries to analyze and understand the real examples so precisely that the falsified data have no chance of being identified as real.
In the specific case of PulseGAN, the model takes a plethysmographic signal as an input generated by any method and tries to improve it by making it as close to ground truth as possible. In the method described by Song et al., the authors use a deterministic algorithm (CHROM) based on the light reflectance given by an analysis of the skin colour space and its change over time (the overall architecture is depicted in Figure 2.9). The authors point out that the method that provides the GAN network input is replaceable by a Deep Learning method, and so a possible approach could be to use RhythmNet as the first step and use the output signal from RhythmNet as input to PulseGAN.

As this method works as an add-on to another method, it can be considered as a second phase of the work, if time allows.

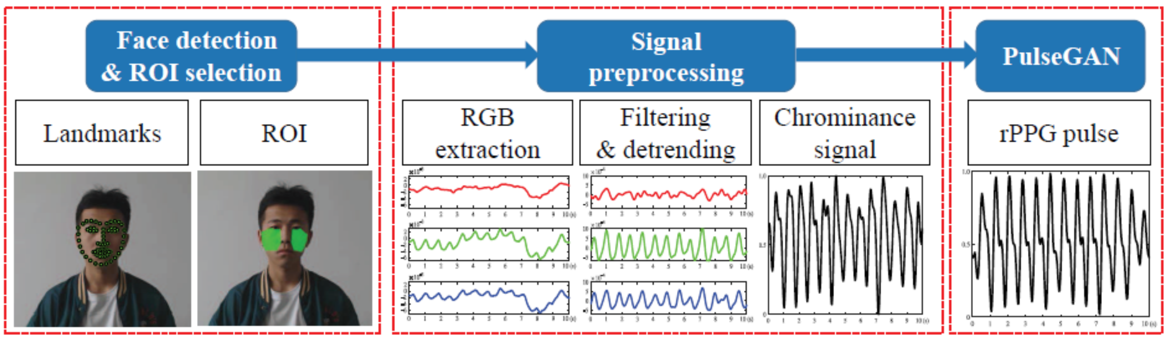


Figure 2.9: The framework of the proposed PulseGAN method, from Song et al. 2020

2.3 Requirements

Given the long list of methods taken into consideration, it has been difficult to just opt for one. Luckily, a list of requirements was provided by Teoresi - making it a simpler and more direct choice. The requirements are listed below and the conclusion is drawn accordingly in the next section. A simple ordinal scale has been adopted: "Critical" (is fundamental for the method to have satisfy this requirement), "Major" (is important to have this requirement matched, though the method doesn't strictly require it in order to be selected), "Minor" (it would be nice for the method to implement this feature, but it can function without).

1. **Method reliability - Major**: First and foremost, it is necessary to understand that the system will neither be marketed nor is it a strict aim to employ the end product into the medical industry right away. This implies that it does not need to match industry device accuracy (depending on the type of device ECG signal can be estimated with a 90%-99% accuracy, which translates to a mean absolute error not larger than 3 bpm - Anh, Krishnan, and Bogun 2006 and Rui Zou et al. 2016). This project goal is to reach an error margin that is as close as possible to the state of the art. Considering the methods seen so far and their evaluation benchmarks, we can empirically set the end Mean Absolute Error to be in the 5 - 10 (bpm) range during inference, when evaluated on the COHFACE dataset (by Heusch, Anjos, and Marcel 2017). Given its large usage in the field - as explained in more details in Section 2.1 - the MAE metric has been picked.

2. **Speed - Critical**: Although the system might not be used for medical purpose, it will most likely be employed in HCI studies and surveys where system speed is required. It is necessary for the system to be able to perform in real-time. To be more specific, if the system can only support one video frame at a time and the video is coming in at 30 frames per second, the system must process each frame in 1/15 of a second or less to satisfy real-time constraints. The reason is twofold: firstly, it would be good to give results on the spot for this kind of frameworks monitoring, while the subject is taking actions and performing experiments; and secondly, with the future possibility to employ this in health care, it would be necessary for screening clinical parameters on the fly rather than on a second stage.

3. **Robustness to light and movements - Minor**: In the field, very few methods have dealt with realistic body motion (only the one by Špetlík), and only more recent methods propose datasets head movements are considered (i.e. speaking, turning the head). On the other hand, light is stressed in every method quite a lot, where most of the dataset varies in terms of

light conditions and light type (halogen, natural or LED). Accordingly, the method hereby proposed aims at resisting light but being potentially sensitive to changes in movement.

4. **Lightweight hardware - Major**: The system not only needs to perform in real-time but also needs to do so on middle-end laptops and, if possible, it should not necessarily require a GPU to reach real-time performance. Additionally, given that the system might be installed on an embedded device, the randomly allocated memory consumed cannot be very high. RAM is used to store both the outputs of intermediate layers (forward pass) and parameters at inference time. Models that have a number of parameters in the million or less order of magnitude should be able to run on any device (whether embedded or not).

5. **Replicability - Minor**: Ideally, one would have a code repository to get an idea of the possible implementations, the frameworks used and other technical details. Unfortunately, none of the methods presented (or more in general analysed in the literary review) has an open-sourced code to start with. As a consequence, it becomes hard to make procedure explanations detailed and flawless in order to make it replicable.

2.4 Selection of Optimal Method

After carefully matching each method with how well it scores in the given requirements, the one by Niu, Shan, et al. 2020 turned out to be the most approachable - and the full score evaluation can be seen in Table 2.1. In fact, this is not only real-time, but it also has a very detailed and intuitive description of the method employed. Additionally, the light conditions scenario have been extensively studied and they concluded that *even under dim light condition, the HR estimation error by RhythmNet is lower than 8 bpm, which is an encouraging result for practical application usage (Niu, Shan, et al. 2020).*
Overall, the method proved to be robust, feasible to implement and to yield very good result against the state of the art methods. Therefore, this is the method selected for inspiring this work.

Besides the above methods, there are many others that have been researched. Some of them were very outdated and results could not hold against newer systems, whereas more recent ones were simply too complicated or too big for this paper time frame.. Overall, researching the field has allowed for the understanding of not only facts pertinent to computer vision field (i.e. video magnification, colour spaces and working with video data processing) but also information borrowed from the medical environment. Strengthened by this knowledge, a better understanding of the differences between the various methods was achieved and allowed to classify and rank them accordingly.

Method	Reliability	Speed	Computational Complexity	Code Availability	Robustness to light and body movements	Total
Janssen et al. (2015)	0	1	1	0	0	2
Spetlik et al. (2018)	0	1	0	2	2	5
Yu et al. (2019)	1	1	0	0	1	3
Niu et al. (2019)	1	2	2	0	1	**6**
Brieva et al. (2020)	2	0	1	1	0	4
Song et al. (2020)	2	1	1	0	1	5

Table 2.1: Method comparison based on a 0 to 2 score (0 - poor, 1 - neutral, 2 - good), 0 means the method does not match the proposed requirement at all (a 0 scored method on speed means it is too slow to function for the scope of this project). 1 entails that the method has a possibility to improve in the given requirement but it doesn't fully match it yet (a 1 score in speed means the method is almost fast enough and a possibility to improve its speed is foreseeable). lastly, a 2 score means the method matches and works well enough with the requirement (such a score in speed means the method *as is* already matched the minimum requirement).

Neural networks can sometimes result hard to interpret and reproduce due to their high number of hyperparameters that need to be tuned. In the method proposed by Niu, Shan, et al., a very classic Neural Net (ResNet18) is proposed as the main block of the pipeline, given the fame of this network, fine-tuned, pre-trained versions can be downloaded from the web. If needed, there is the possibility to make it more suitable for the NrPPG case through transfer learning. RhythmNet does put most of the work in the preprocessing of the input. This intuitively makes more sense than building a deep enough network that will eventually learn the signal construction. It has been proven by more deterministic methods and years of research in this field that insightful information is conveyed by a certain region (the face) and certain features of the image (colours and reflectance). The method proposed exploits these points of interest by both projecting the usual RGB space into the YUV space (where reflectance has greater weight) and by more accurately selecting the ROI by using face landmarks rather than just face detection techniques.

Chapter 3

Components Investigation

Although a method has been selected already, each component takes care of a part of the pipeline overall. In fact, the ANN plays a minor role in the construction of the system. Tasks such as face detection, cropping and input pre-processing, are needed in other parts of the project. In return, these parts can be built with different approaches. For example, a wide range of face detectors is present in the literature, and choosing one over another must be done carefully. Accordingly, each method taken in consideration is compared to other available ones, and performances are evaluated with the end task in mind.

Therefore, an overview of the face detectors considered for the project - and their evaluation - is presented and discussed in Section 3.1.

Section 3.2 provides a detailed overview what the pool of possible datasets looks like. Only one of the datasets will be selected from this list (i.e. COHFACE, presented by Heusch, Anjos, and Marcel 2017) and explanations for such choice can be found in Section 3.2.4.

Finally, the dataset chosen in the previous section is studied in terms of lighting condition variations and the subject's body movements, so to understand the model limits and strengths that will train on this data. Consequently, the dataset selected in Section 3.3 is qualitatively assessed in the analysis and in its interpretation, in terms of how it reflects on the data itself. Accordingly, Section 3.4 briefly explores how any of the previous conclusions can possibly affect the model performance.

3.1 Quantitative evaluation of face detectors

Face detection is that process where a human face is automatically recognised in a picture or a video frame, and it can be done in several different ways. In recent times, Deep Learning methods have taken over thanks to their speed and reliability. Yet a variety of approaches and models have been released in this field too. Some of these methods are hereby considered, the best ones are selected and then compared through a quantitative analysis.

A list of all models considered in the evaluation first step is here to follow.

1. **Haar cascade face detector**:
   It is an Object Detection Algorithm applied on face detection from video or image data, it is very well known and still used in many devices as of now. The algorithm firstly proposed in Viola and Jones 2001, leverages the use of Haar Features to detect pixel patterns that recall the ones associated with a face. In a nutshell, the algorithm makes use of a given number of Haar

filters (two valued filters) to determine which face features are most valid. The final classifier makes the face decision (face or not face) based on the output of all the features. Each feature cannot categorize the picture alone, instead it becomes powerful when combined with others. Most of the work boils down to which features the classifiers are trained on. These features are extracted by sliding a filter (or kernel) over a set of face images and on an equally large set of faceless images. This kernel is split into regions or parts. To compute a large number of features, each kernel potential sizes and positions are employed. The sum of each pixel value for each region of each kernel is evaluated. Because this procedure can become very expansive very fast, the authors used the integral picture to solve the problem. An integral image is a data structure for quickly calculating the sum of values in a rectangular subset of a grid. This lowers the calculations for a particular pixel to a four-pixel process, no matter how big the image is. Because there are still a lot of regions left, an optimizer (namely Adaboost) is employed. This finds the best threshold to classify the faces as present or not present.
AdaBoost stands for "Adaptive Boosting" and was originally presented in Freund and Schapire 1997. It was the first highly successful boosting algorithm developed for binary classification. Adaboost represents a popular boosting technique for combining multiple "weak classifiers" into a single "strong classifier". By putting many such models together, AdaBoost is able to generate a model that overall is better than the individual weak classifiers taken individually. Adaboost makes use of many decision trees with some level of depth. At each iteration, a new weak classifier is introduced sequentially and aims to compensate for the "shortcomings" of previous models to create a strong classifier. The overall goal of this is to consecutively fit new models to provide more accurate estimates of the response variable. In fact, AdaBoost does not only accept decision trees as weak learners: any machine learning algorithm can be used as a base classifier if it accepts weights on the training set.

Advantages

- Functions almost in real time on the CPU.
- Simple architecture.

Disadvantages

- Many false predictions (Rosebrock 2021).
- Does not perform well under occlusion.
- Scale of Haar Filter needs to match scale of face.

2. **MultiBox single shot (SSD) detector**:
The single shot detector is a simple solution to the problem of face detection, but it has shown to be quite successful thus far. As clearly explained on the ArcGis developers website, an SSD model (first introduced in Liu et al. 2016) is made of two parts: a backbone convolutional model and the SSD head. The Backbone CNN is a feature extractor-trained image classification network. In most models, the fully connected classification layer is omitted. The outputs of the SSD Head are interpreted as the bounding boxes and object classes in the spatial position of the final layer activations. Instead of using sliding windows, SSD divides the image into grid cells. Each grid cell is in charge of recognizing items in its own area of the picture. When there are multiple elements in a single grid cell, the anchor box and receptive field approaches

are applied. Each grid cell in SSD can have several anchor/prior boxes. These anchor boxes are pre-defined, and each one controls the size and shape of a grid cell. During the training, SSD employs a matching step to match the proper anchor box to the bounding boxes of each ground truth. The basic assumption of the SSD design is the receptive field. This allows for the detection of objects at various sizes and the production of a tighter bounding box. The easiest way is to apply a convolution on this feature map and transform it to a NxN grid. SSD goes one step further, as it applies additional convolutional layers to the backbone feature map, resulting in each of these convolution layers producing object detection results. Predictions from previous layers can aid when dealing with smaller sized things since earlier layers with smaller receptive fields can represent smaller sized objects. As a result, SSD allows you to create a grid cells hierarchy at different tiers. For example, a 4x4 grid can be used to locate little items, a 2x2 grid for medium-sized objects, and a 1x1 grid for objects that span the whole image.

Advantages

- The most accurate of the four methods.
- Functions in real-time on CPU.
- OpenCV version performs well with not frontal faces too.

Disadvantages

- Long training time
- Might be hard to detect small objects

3. **Histogram of Oriented Gradients (HOG) detector**:
The histogram of oriented gradients (HOG) has been first applied to the human detection domain in Dalal and Triggs 2005. As clearly described in Nagaraja and Prabhakar 2015, the method counts the number of times a gradient orientation appears in a certain picture area. This method differs from other feature descriptors in that it is computed on a dense grid of evenly spaced cells and it employs overlapping local contrast normalization for increased accuracy. The basic idea behind the histogram of directed gradients descriptor is that the distribution of intensity gradients or edge directions may be used to characterize local objects' look and form within an image. The image is split into small linked areas called cells, and a histogram of gradient directions is created for the pixels within each cell. The concatenation of these histograms is the descriptor. Local histograms can be contrast-normalized for better accuracy by computing an intensity measure across a wider portion of the picture, known as a block, and then using this value to normalize all cells inside the block. Because of this normalization, the invariance to changes in light and shadowing is improved.

Advantages

- Dlib implementation works fast even on the CPU.
- Functions very well for frontal and slightly non-frontal faces.
- Lightweight model.

Disadvantages

- Doesn't detect small faces.
- Bounding box are often miscalculated.

4. **Maximum margin object (MMOD) detector**:
   To achieve state-of-the-art results, the max-margin object-detection method (introduced in King 2015), or MMOD, employs a CNN to extract features from a windowed picture. This is not appropriate for real-time applications, unlike the previously described face detection solution, although it does perform slightly better when identifying faces at unusual angles.

   The way MMOD works is that it calculates the scalar product between a given feature vector and a previously trained vector to determine if it belongs to a specific class. This already trained vector is a vector that generalizes the features that any target object has. The intuition is that MMOD wants to maximize the margin between the correct class and any other class, in this case between faces and anything else.

   Advantages

   - Functions for different face orientations.
   - Dlib implementation resists well to occlusion.
   - Very fast on GPU.

   Disadvantages

   - Very slow on CPU.
   - Doesn't work well with small faces.

5. **Multi-task Cascade CNN (MTCNN) detector**:
   This model is based on K. Zhang et al. 2016 and it sees 3-stage neural network detector. To detect faces of various sizes, the picture is first scaled several times. The P-network (Proposal) then searches the pictures for a first detection. This first detection has a low confidence threshold and it purposely identifies a large number of false positives. The identified faces (again, containing many false positives) are sent into the R-network (Refine), which, as the name implies, filters detected areas to get accurate bounding boxes. The O-network (Output) conducts the final refinement of the bounding boxes in the last step. This method not only detects faces, but also ensures that bounding boxes are accurate and exact.

   Advantages

   - Fast on GPU.
   - High accuracy.

   Disadvantages

   - Very slow on CPU.
   - Not working well on oriented faces.

In order to come to an appropriate conclusion, some numerical and visuals performance tests have been run on a subset of the above-mentioned detectors. When analysing the performances, three key factors have been researched: speed, accuracy and confidence. A summary of the results is shown in Table 3.1. In terms of face found, a confidence threshold value should be introduced for the SSD detector. In fact, other detectors take into account the possibility of not returning a face at all - when they do they also return a confidence score which can be as low as 0.1. The SSD detector, on the other hand would always return a face if the confidence threshold was set to 0.0.

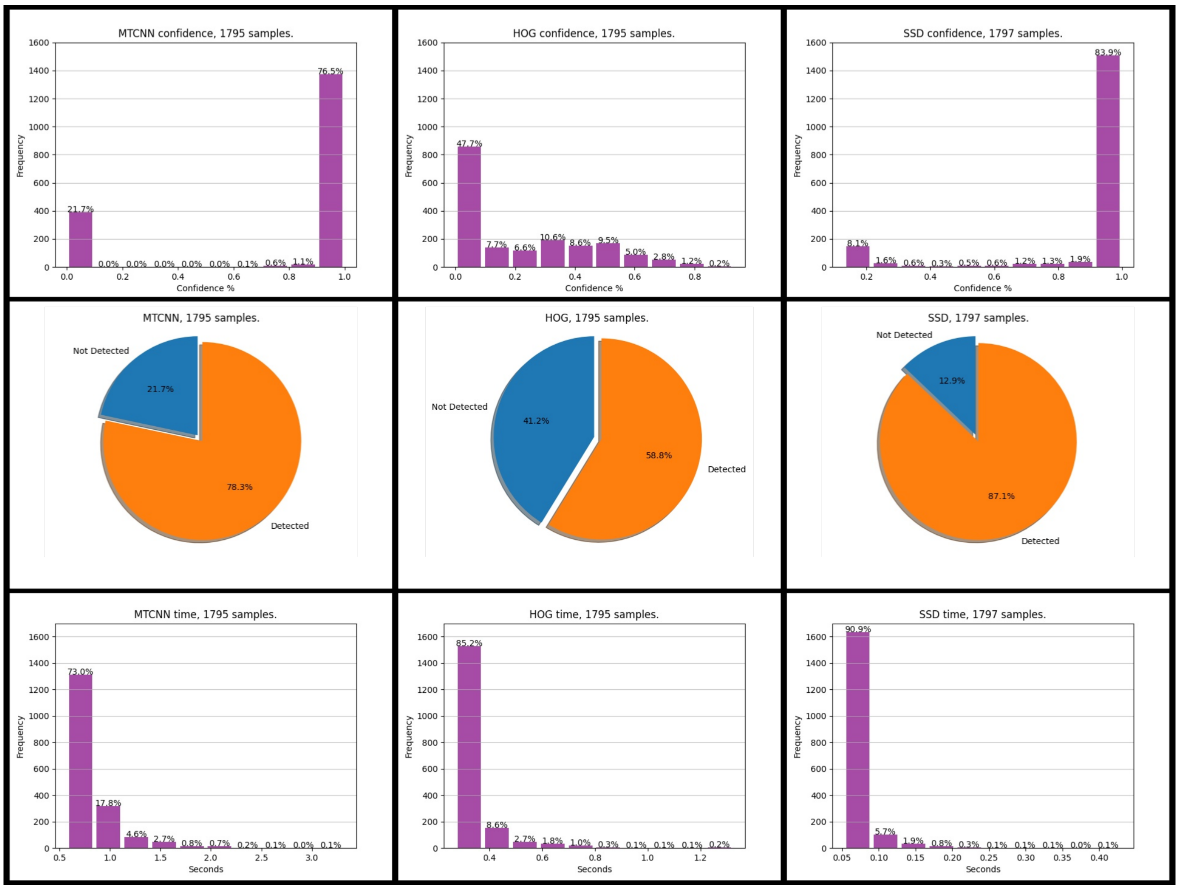


Figure 3.1: The result of the detectors taken into consideration. On the first row we have an histogram of the confidence for each model, on the second row piecharts have been employed to show the percentage of faces detected vs. the ones not detected. Lastly, an histogram shows the time taken to make the prediction in the third row. The above results have been obtained on an Intel Core i7-7500U Intel HD Graphics 620 CPU.

This means that the user needs to set a threshold over which a face can be classified as such. Basing it on literature, this threshold has been set to 0.95 for this experiment.

For each of these factors, an appropriate visualisation of the end results has been selected, as shown in Figure 3.1. Additionally, the methods have been evaluated on two different datasets. The first one is a small custom made dataset containing 125 high resolution pictures (all containing a face) with different sizes. The second one is a known dataset in face recognition (namely extended Yale Face Database B) that contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions, created by Georghiades, Belhumeur, and Kriegman 2001. Out of this last, a subset of 1795 data points has been selected.
The main distinction between the two is that the former one presents way more realistic and challenging situations (varying in picture size and resolution), whereas the latter one presents small pictures of only faces, all of the same size. In Figure 3.1, only the results for the Extended Yale B dataset have been reported. It is evident that SSD is the detector that performs best in almost any category. Though, when tested on the other dataset is noticeable that speed performances majorly decrease.

It must be noted that this experiment does not account for false positives. Only faces containing data was used, so the experiment really only accounts for those cases where the detector does not find a face where the face is present. Additionally, no annotation came along the datasets used, so this would have made it tedious to check every bounding box returned by the detectors with the

	Confidence level	Face found	Time taken
HOG	D1: m = 0.235, std = 0.244 D2: m = 0.209, std = 0.231	D1: m = 0.672, std = 0.471 D2: m = 0.587, std = 0.492	D1: m = 7.312, std = 3.237 D2: m = 0.331, std = 0.101
SSD	D1: m = 0.869, std = 0.255 D2: m = 0.896, std = 0.254	D1: m = 0.87, std = 0.33 D2: m = 0.849, std = 0.357	D1: m = 0.065, std = 0.006 D2: m = 0.068, std = 0.026
MTCNN	D1: m = 0.649, std = 0.467 D2: m = 0.776, std = 0.409	D1: m = 0.664, std = 0.474 D2: m = 0.783, std = 0.412	D1: m = 3.797, std = 1.647 D2: m = 0.815, std = 0.256

Table 3.1: Summary of results for both datasets (D1 being the custom made one and D2 being the Extended Yale B) for each predictor model. For each of the factors studied the mean (m) and standard deviation (std) have been reported.

original position of the face in the picture. Therefore, examining false predictions fell out of the scope of this experiment.

Given the information listed above, it seems clear that the most suitable method for our specific case is the Single Shot Detector. This results to perform best on the accuracy score (without considering possible false positives), but most importantly it is the fastest face detector out of all of those taken in consideration. Additionally, the SSD model is simple compared to methods that require object proposals because it completely eliminates proposal generation and the subsequent pixel or feature resampling step and encapsulates all computations in a single network. This makes SSD extremely fast and simple to integrate, making it the best fit for our specific case scenario.

3.2 Data

The data for this project will be requested from different sources and then one will be selected based on data availability and quality: Generally, a good dataset should contain different setups in both lighting and point of view. This is essential as we aim to make the model generalise across different scenarios well. As previously mentioned, the data comes with labels: a matching EEG (or PPG) signal comes along with the video footage.

3.2.1 ECG-Fitness database

This data has been collected a few years ago by both a PhD student and a fellow junior researcher for his own research at the Czech Technical University. In Špetlík 2018, authors describe the dataset extensively, and here some information are reported.
The data has the following description:

- Data: Realistic corpus of subjects performing physical activities on fitness machines, 6 x 1 minute video recording (no sound) + ECG per subject (207 videos in total).
- Subjects: 17 subjects (14 males, 3 females) with an age range of 20 to 53 years.
- Camera: two RGB Logitech C920 webcameras - 30fps, 1920 x 1080 pixels stored in an uncompressed YUV planar pixel format (approx. 6GB per video).
- Camera setup: one Logitech camera was attached to the currently used fitness machine, while the other was positioned on a tripod as close as possible to the first camera.
- ECG: two-lead Viatom CheckMe™ Pro device with CC5 lead.
- Activities: speaking, rowing, exercising on a stationary bike and on an elliptical trainer. Speaking and rowing performed twice - once with halogen lighting resulting in a strong 50Hz (100Hz) temporal interference, and once without.

- Lighting setup: three lighting setups were used: (i) natural light coming from a nearby window, (ii) 400W halogen light and (iii) 30W led light.
- Hearth rate: the lowest 56 bpm, the highest 159 bpm, the mean 108.96 bpm, standard deviation 23.33 bpm.

This dataset turned out to be very confusing (and inconsistent with what is described in Špetlík 2018). We have been in touch with the author, who could not answer some of the questions extensively enough as he himself had doubts about some of the values. After spending time trying to understand the data, the focus has shifted on a clearer and better-documented one (i.e. COHFACE).

3.2.2 DEAP: A Database for Emotion Analysis using Physiological Signals

This dataset is also available on request. DEAP (by Koelstra et al. 2012) is a collection of data used to study human emotional states. While watching 40 one-minute long clips from music videos, the electroencephalogram (EEG) and peripheral physiological data of 32 individuals were monitored. Each movie was assessed on arousal, like/dislike, dominance, and familiarity by the participants. The only data relevant to this research is face video data along with the matching plethysmographic (lung volume measure) signal, from which HR and RR can be estimated.
Commercial cameras are promising contact-free sensors, and remote photoplethysmography (rPPG) has been studied to monitor heart rate from face videos.

3.2.3 COHFACE

This dataset is publicly available and contains RGB video sequences of faces, synchronized with the heart and breathing rate of the recorded subjects. In Heusch, Anjos, and Marcel 2017 and Špetlík 2018, this dataset is described as a publicly accessible database for evaluating the algorithms performance for remote pulse rate monitoring. This collection contains a video of 12 subjects under a variety of lighting situations. The video sequences have been recorded with a Logitech HD C525 at a resolution of 640x480 pixels and a frame rate of 20Hz. Physiological recordings, namely blood volume pulse (BVP) and breathing rate, have also been recorded. Physiological signals have been acquired using devices from Tought Technologies and using the provided BioGraph Infiniti software suite.

3.2.4 Dataset selection

The above mentioned datasets provide the same kind of data (i.e. video coupled with some sort of signal from which HR can be derived). Though, they do differ in terms of what signal is provided as ground truth as well as data quality/size, subject distribution (in terms of gender and ethnicity) and light and movements conditions.
ECG-Fitness is simply too big to just perform some analysis on it or to be stored in an average range laptop. All the 200 one-minute-videos are uncompressed in order to facilitate the ECG extractor model tasks. In fact, as mentioned in Yu, Peng, et al. 2019, video compression has a great impact on how well the model can predict the HR. Indeed, this dataset aids model prediction capabilities, at the expenses of ease of use. Additionally, confusion occurred on how to extract the HR signal from the data provided and the authors did not seem keen in answering any doubts about it. As mentioned by the author, the confusion might come from the data being the result of a complicated

process of reverse-engineering of a portable ECG device. For these reasons, this dataset has not been considered.
In PURE, the lighting conditions were not particularly varied. In fact, there was only frontal daylight with clouds slightly changing the illumination conditions over time (as mentioned in Boccignone et al. 2020). This dataset has also been discarded as it is crucial for the system to be well resistant to a variety light conditions.
Finally, COHFACE presents an easy to read BVP signal, both being lightweight (less than 1GB) and accurately synced with the videos. The ease of use and the different light scenarios played a major role for this dataset to be selected over the other ones.

3.3 Exploratory data analysis

This section aims at explain the analysis conducted on the COHFACE dataset by Heusch, Anjos, and Marcel 2017. A better explanation of what COHFACE contains in terms of a number of video and light conditions can be found in Section 3.2. Generally, COHFACE contains many videos of relatively stationary people facing a camera. Analysis has been carried on both aggregate (all videos together) and individual (per video) data. The latter makes it easier to see each video condition, while it is also difficult to get a general idea around the dataset. The former gives a more general idea of data distribution overall. All the data taken into consideration is graphically shown by this demo gif, where the numbers on each quadrant of the screen represent its luminance factor. The aim of this analysis is to understand what limitations the data could introduce to the final model. As discussed in section 3.4, the data distribution and the characteristics researched in the following analysis brought to light the model possible limitations or advantages. Finally, Section 3.3.4, will give an overview of the subjects distribution in terms of skin colour and gender.

The goal of this exploratory analysis is to research the following:

1. Distance between the nose tip reference landmark and the centre of the screen
2. Roll, pitch and yaw of the subject's head
3. Illumination for each of the four quadrants of the image

In the next sections, there comes a more in-depth analysis of each factor and its trend for both the individual video analysis and the more coarse one.

3.3.1 Distance nose - screen centre

For point 1 (distance), a line graph has been chosen so that the distance variation could be studied over time. The distance in turn was calculated using the Euclidean formula: $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$, the two points in question being the one at the tip of the nose and the exact center of the frame. The point of the nose is calculated using the same landmark positioning model mentioned in subsection 4.1.1 (presented in Kazemi and Sullivan 2014).

The line graph is meant to represent how distance varies over time. In a situation of complete immobility, we should expect the distance value to remain constant throughout the video. Though the distance is not necessarily zero in the ideal case because the subject may never have the tip of his nose coinciding with the centre of the screen. This is determined by the aspect ratio of the screen and the position of the subject in front of the camera. Additionally, a min-max normalization is

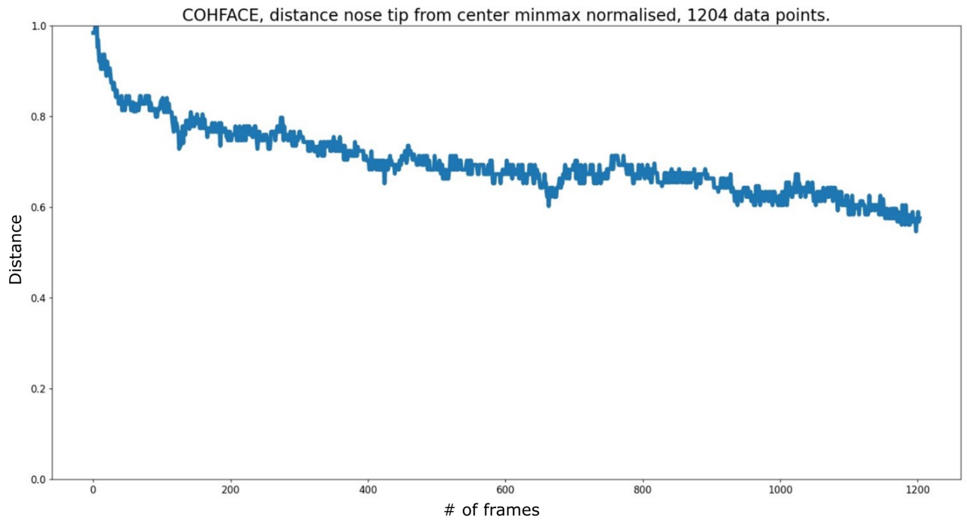


Figure 3.2: The figure shows the value trend for the distance for one specific video. On the x axis there is time and on the y axis the value of the normalised distance.

applied to make the various single graphs more comparable with one another. An example of the generated graphs can be seen in Figure 3.2.

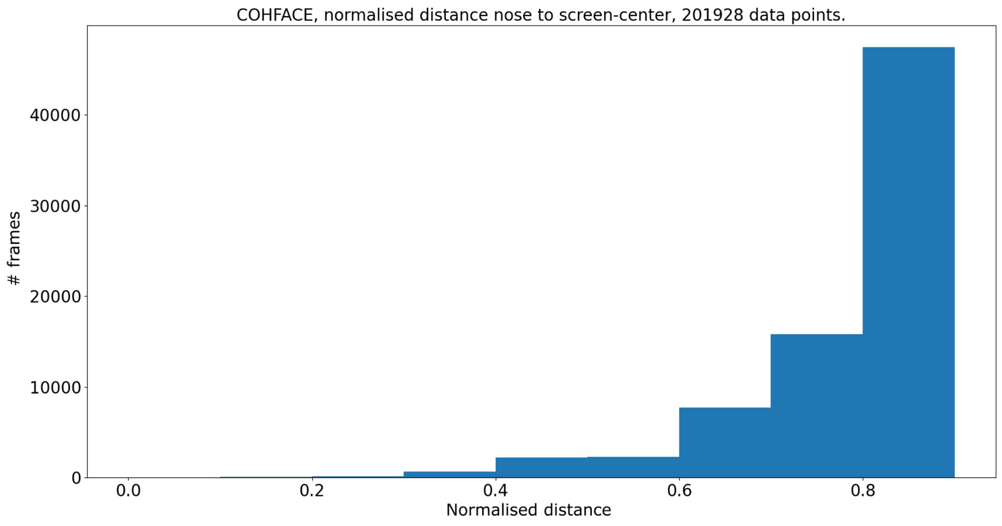


Figure 3.3: The plot shows an histogram of all the normalised distances for all the video in the dataset. Again is noteworthy how the majority falls in the last bin, meaning the subject quite always remained firm when in front of the camera.

It is noticeable how the distance variation is quite insignificant (i.e. the line seems to always remain in the 0.6 - 0.8 range), and this implies that the subject is mostly sitting at the same distance she was when the video began. The same behaviour is reflected in the data aggregated version. In this case, a histogram plot might better convey the distribution of the data overall. Specifically, it is interesting to see how most of the normalised distances follow in the last bin. Again, the distance magnitude is not relevant, but it is rather interesting to look at how "constant" the distance value is. As one might notice from Figure 3.3, the distance between the nose and the screen center mostly seems to be around $0.7 - 0.8$. Because the distance always falls around the same value, it follows that the subjects remain steady (in terms of body motion) during the whole recording and for most of the recordings in the dataset.

In light of this, we can claim that the characters from most of the videos do not move too much, making it easier for a NN to learn the function extracting HR from skin pixels.

3.3.2 Head orientation movements

Another factor that is worth looking at is how the subjects head moves along the three axis (i.e. roll, pitch and yaw). The data for such evaluation has been obtained from a customised implementation of this python framework that, in turn, employs a pre-trained MTCNN model, fine tuned for this task.

From Figure 3.4, it is easy to infer which videos have the most and least movement, but the magnitude of these is hard to interpret. In addition, there were some outliers in the data collected. These have been removed when laying within more than two standard deviations from the mean. Generally, it can be observed that the head movement is almost always minimal - which can be empirically confirmed by watching the analysed videos. Again, an ideal situation is one where the values for the three variables considered remain constant over time (i.e. a flat line).

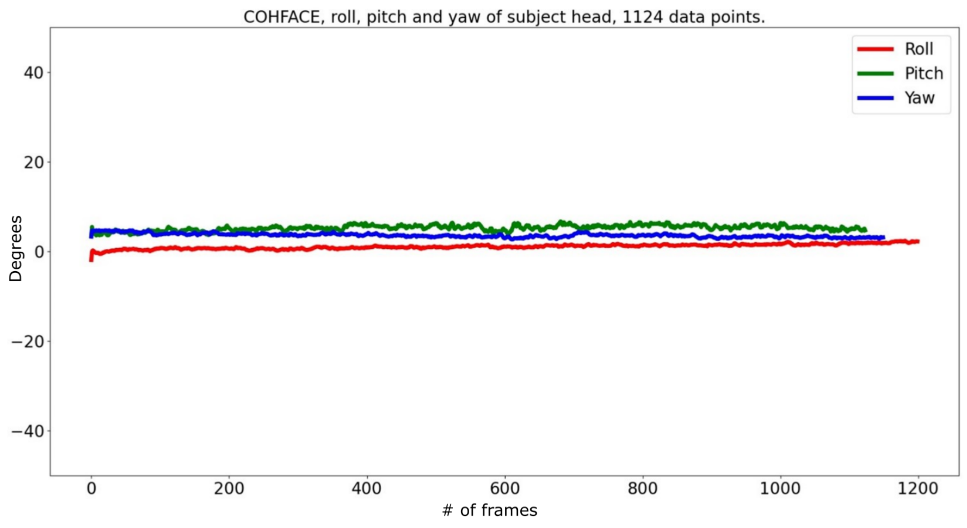


Figure 3.4: The line plot represents the movements of roll pitch and yaw along the whole duration of the video clip. It can be noted that the subject does not move his/her head too much (this visually translates to the line not varying too much from its start value).

The results perfectly match with what is observable from the videos in general. All results show to be very stable in terms of head movements both from the graphs produced and the videos linked to them.

As opposed to the single videos charts, the goal entailed in the coarsed data representation is to show the distribution of the dataset in general. Therefore, for the visualisation of such aggregated data a box and whisker chart has been used. These generally represent five summary ratings, but for this analysis the mean is shown too:

1. Minimum value
2. First quartile
3. Median (in green)
4. Third quartile
5. Maximum value
6. Mean (in red)

The box represents the quartiles bisected by the median. The whiskers represent the difference between the quartiles and the minimum and maximum values - or otherwise exceptional values (outliers) - that are beyond the whiskers. The distribution is symmetric when whisker lengths and rectangle heights are similar to each other. The whiskers length shows the normality (short whiskers) or exceptionality (long whiskers) of the phenomena, and the aberrant values that lie beyond the whiskers. This type of visualization is useful in our case because it represents the deviation of the data set from the mean particularly well. For example, the most evident case of non-deviation is that of head movement. In fact, we have already seen that the subjects (at least for the dataset in question) do not move from the original point. The graph in Figure 3.5 is therefore "skewed" on the vertical axis, indicating the little variation of the value.

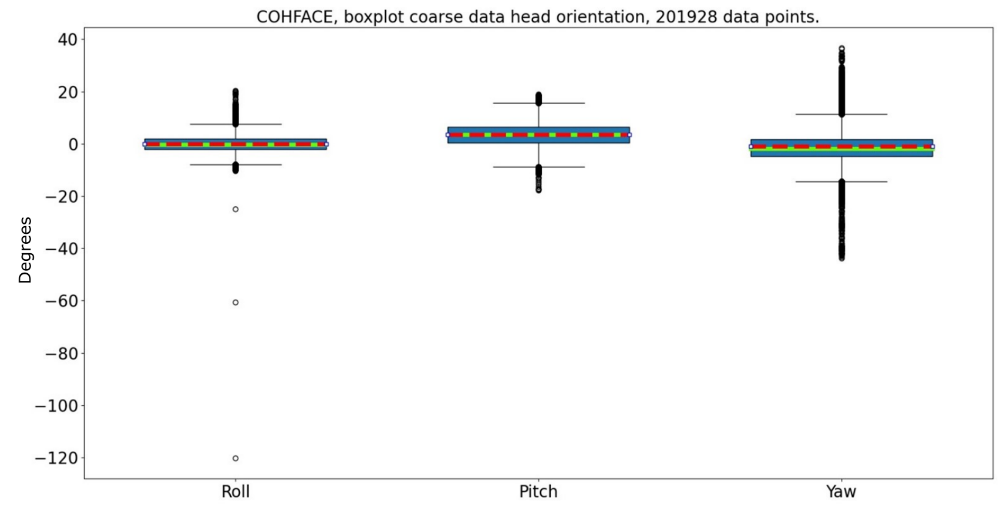


Figure 3.5: The plot shows the distribution of the orientation movements of the head for all the videos in the dataset.

The presence of many outliers must be noticed, as these likely due to the mentioned noisiness of the algorithm used to detect motion.

3.3.3 Per quadrant luminance

Finally, illumination has been analysed to study the light conditions of the room in which the subjects sit. This value is not easy to capture because there is not exactly a univocal way to calculate colour intensity. There are several ways that deal with calculating relative illumination and contrast, trying to make the values obtained more comprehensible to the human eye, emphasizing the physiological aspects: the human eyeball is more sensitive to green light, and less to red and blue. Since we need to observe illumination in terms of how the artificial neural network will read it, intuitively we do not need to worry about emphasizing the physiological component. The approach consisted in dividing the image into four equal parts. Then, for each pixel in each quadrant, colour values (so as to determine the intensity) have been aggregated and normalized w.r.t. the three channels (divided by three). The results are extremely static throughout the dataset which could be explained by the left side of the image having much more light (there seems to be a window or a light source).

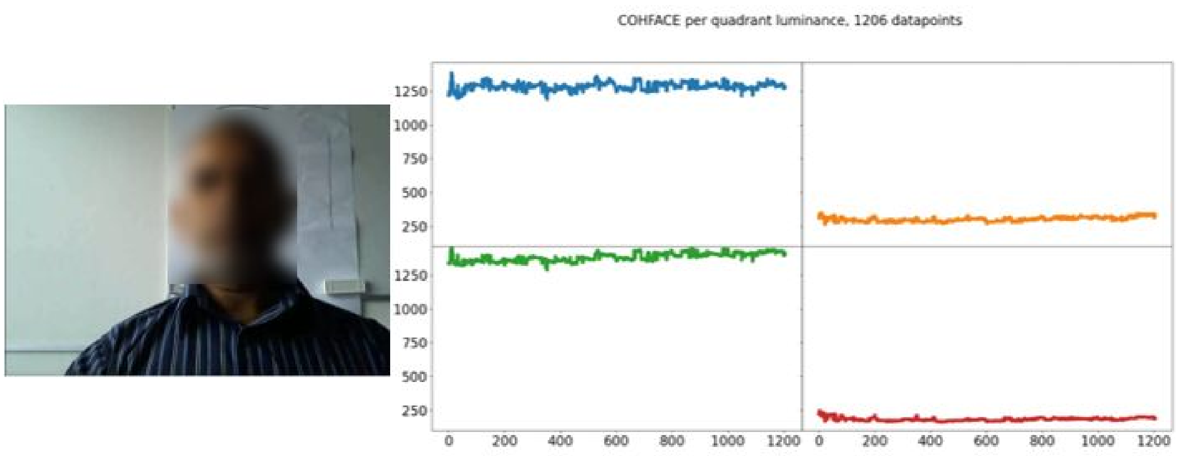


Figure 3.6: In the plot we can see a sample frame on the left and the representation of its luminance values (per quadrant) on the right. The face of the subject has been blurred out due to privacy concerns. On the x axis the number of frames.

In Figure 3.6 the y-axis scale is an arbitrary number that varies between 0 and 1500. In this instance, normalizing would not make sense because we would lose too much useful information when comparing one quadrant with another. In general, we can see that the results also make sense from the confirmation for an opposite situation where the light comes from both sides (depicted below in Figure 3.7).

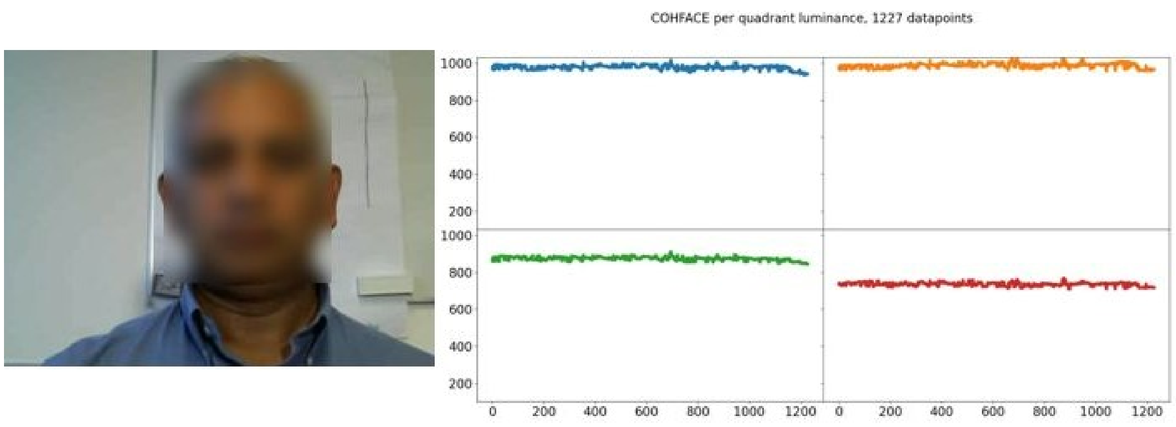


Figure 3.7: In the plot we can see a sample frame on the left and the representation of its luminances values (per quadrant) on the right. The face of the subject has been blurred out due to privacy concerns. On the x axis the number of frames.

Generally, the cases where the light is induced on both sides are way less frequent in the dataset. This characteristic (represented well in Figure 3.8) makes it easier for the model to learn about the dataset but, in turn, it makes the model ability to generalise less robust. In the coarse data representation (once again through a box and whisker plot) we can clearly see how the scenario where the light only comes from the right-hand side of the subject is way more frequent than the one where artificial light is shed on the left-hand side of the subject.

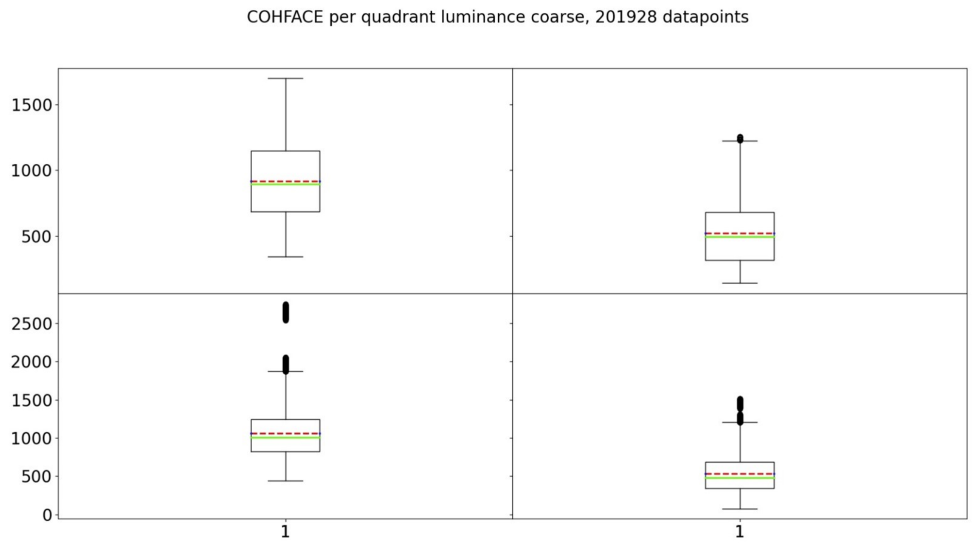


Figure 3.8: In the plot above is easy to determine that the light mostly comes from the RHS of the subject. Just by looking at the mean this information results evident.

3.3.4 Gender and skin colour distribution

The data consists of 40 video from 38 different subjects. The characteristics considered in this analysis are ethnicity and gender. This because subjects from different genders and nationalities carry very different physiological features (e.g. facial characteristics for gender, and skin tone for ethnicity).

Because we don't want any of the nationalities or genders to be over-represented, it is crucial that we try to balance the available data before training. Unfortunately, as shown in Figure 3.9, the data itself is not well balanced. White men are overly represented and we surely cannot discard subjects with these characteristics as the data is already scarce. The best we can do is to balance the subjects when training as much as possible.

As noted in Johnson and Khoshgoftaar 2019, highly skewed data can be dangerous for neural networks performance because most networks will be biased towards the over represented group - and, in certain circumstances, it may completely disregard the less represented group.

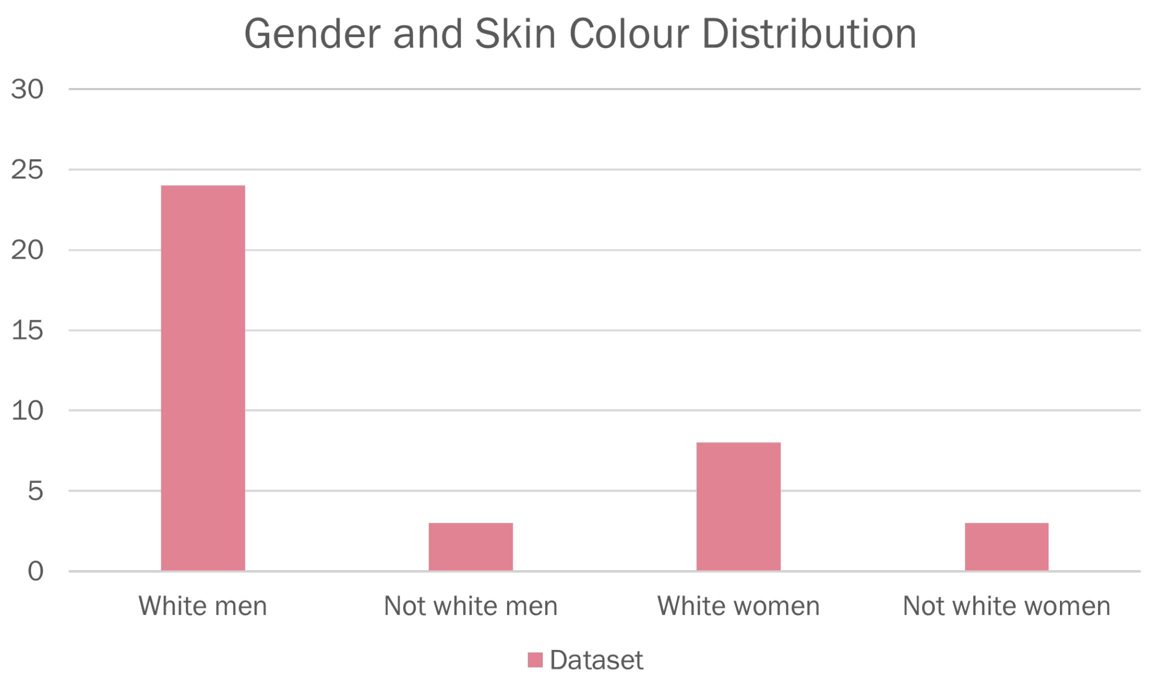


Figure 3.9: Distribution of subjects gender and skin colour.

3.4 Conclusions

Having performed such data analysis, some characteristics became blatant. The data at hand does not contain many challenging scenarios in terms of subject movements, but it does contain different lighting conditions.
Furthermore, there is poor balance between subjects ethnicity - where Caucasian people are present more than any other ethnicity. Additionally, more men than women are represented in the data. From this, it is possible to conclude that the model won't be particularly accurate when predicting women or not-white subjects heart rate. As mentioned in Section 4.3.4, an attempt to create balanced training/validation subsets was made.

Moreover, the model won't be particularly robust to abrupt subjects movements as it has been trained on subjects sitting quite steadily. On the other hand, changes in lighting conditions should not affect the model performance. These characteristics are consistent with the previously set goals.

Chapter 4

Proposed approach

The aim of the following chapter is to explain the method implemented in this research project. Firstly, an overview of the thought pipeline is provided, assessing each step and going through its implementation details. As the pipeline can easily be divided into sections, this chapter will follow the same approach. Section 4.1 will explain the landmarks positioning process as well as the segmentation technique used to divide the image in ROI's.

Section 4.2 will explain how the data resulting from the processing has been handled to create the spatial-temporal map.

Section 4.3 will detail the network architecture and its hyperparameters tuning. Finally, in Section 4.4, an analysis of the issues encountered is reported. Here, problems faced during the development and their solutions will be detailed.

All of the face processing functionalities listed in the following sections have been nicely wrapped in a python package, called FaceManager. This package implements four classes (FaceDetection, FacePoseEstimator, FaceProcessing, FaceStabilizer) that deal with every bit of the face regions processing pipeline. This package will also be made available to the Python community through the Python Package Index.

4.1 Face processing

Face processing in the system proposed concerns a set of steps, depicted in Figure 4.1. This section deals with everything concerning face detection, landmark positioning, face alignment, diving the image in N regions of interest, padding, and projecting the output image onto a new chromatic space (i.e. YUV).

According to the pipeline shown in Figure 4.1, we can identify several steps in the overall process. These steps take a whole video frame as input and output a segmented region containing a face within the same frame. This region is both projected into the YUV chromatic space and is divided into N regions of interest.

An explanatory subsection will follow for each of the identified steps.

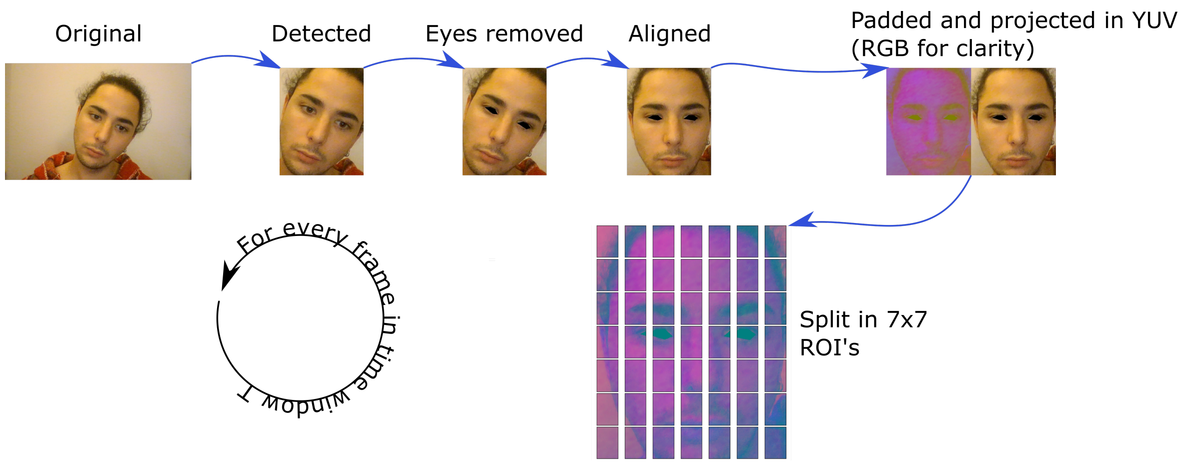

Original
Detected
Eyes removed
Aligned
Padded and projected in YUV (RGB for clarity)
For every frame in time window T
Split in 7x7 ROI's

Figure 4.1: The overview of the pipeline until the segmentation of the detected face. Starting from the original image, a face is detected, and the same face is then aligned. Face alignment allows for the creation of correspondences between different images, enabling the execution of subsequent activities on a consistent basis. The aligned face is then projected on another color space and subsequently padded, so that the number of pixels becomes divisible by the number of ROI's (in the example N ROI's= 49).

4.1.1 Face detection and landmarks positioning

As the performance of different face detectors has already been quantitatively analysed in Section 3.1, the scope of this section is limited to explaining how the chosen detector is employed in the overall pipeline.

In order to remove parts of the image that do not carry useful information towards the HR estimation, only the face region has been taken into account. As previously mentioned, studies report that the areas that mostly pertain the skin colour change due to the heartbeat are the cheeks and forehead. Hence, a robust detector is used to segment out the face region. This segmentation will also make it easier for the next step to place landmarks on the given image. The performance of all the face detectors considered can be seen in Figure 4.2, where both the region that shall be cropped out (i.e. the green rectangle around the face) and the landmarks placement are displayed on three different head orientations and backgrounds.

Once the face has been cropped out, the resulting image will be fed as an input to a face landmark estimator, pre-trained on the ibug 300-W dataset Sagonas et al. 2013. Because this model is meant to work with the HOG face detector, some tweaking in getting it to work as well with the SSD face detector has been performed. Facial landmarks detection has been investigated for decades. Many neural network (NN)-based techniques for detecting landmarks have been suggested, particularly convolutional neural network (CNN)-based approaches. Regression and heatmap methods are the two types of CNN-based methodologies. This study employs a model that was originally presented by Kazemi and Sullivan 2014 and that is made out of an ensemble of regression trees that is used to evaluate the face's landmark positions from a small set of pixels with different intensities, resulting in real-time performances and accurately confident predictions.

The model has been downloaded with pretrained weights from this ddlib repository. This outputs location indices for 68 facial landmarks, from which different parts of the face can be estimated. These landmarks will become extremely relevant when detailing the alignment and eye region removal parts of the pipeline.

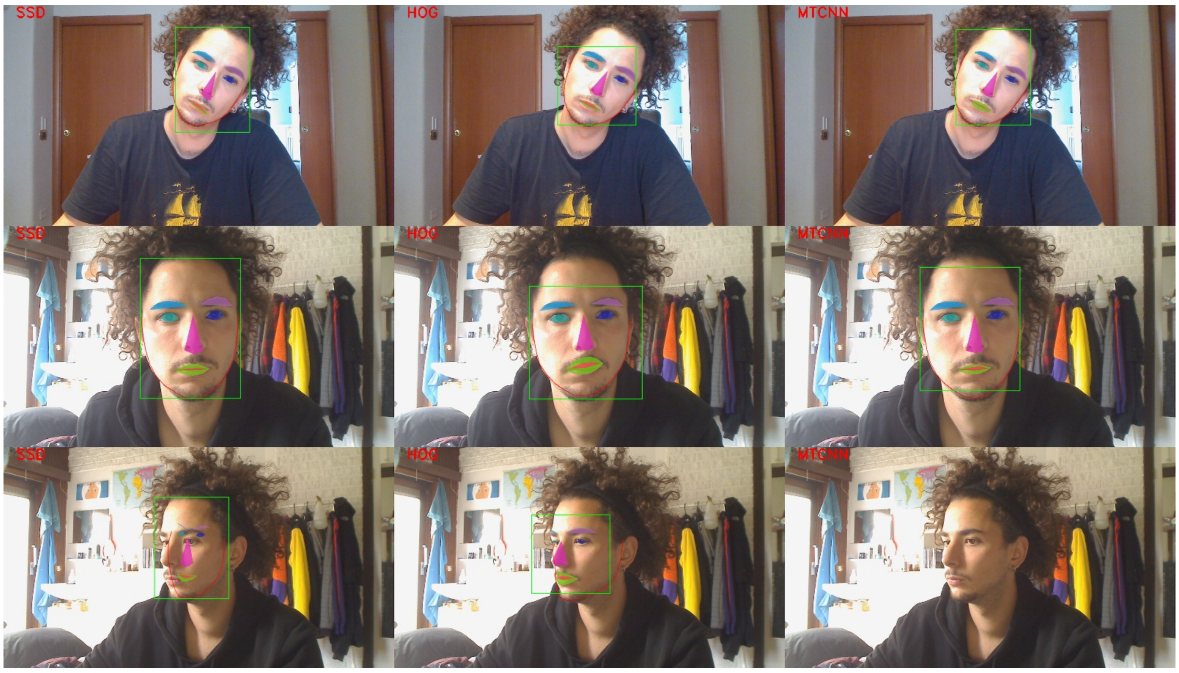


Figure 4.2: A visual representation of how the three face detectors set basis for landmarks positioning. An interpolation has been performed between landmarks, so to see the area they are supposed to refer to.

4.1.2 Alignment, projection and division

The last three steps of the pipeline are alignment, colour space change and segmentation into ROI's. Face alignment may take many different shapes. Some approaches attempt to impose a (pre-built) 3D mesh on the input picture and then transform it so that the landmarks on the input face match the landmarks on the 3D model.

Other more straightforward approaches (as the one used in this research) depend solely on facial landmarks (particularly the eye regions) to generate a normalized rotation, translation, and scale representation of the face. Because many facial recognition algorithms and deep learning techniques benefit from performing facial alignment before attempting to identify the face, we too perform this normalization. The method employed to perform such normalisation does not utilise Deep Learning approach. In fact, speed and simplicity have once again been placed before robustness and completeness. The overall alignment pipeline can be observed in Figure 4.3.

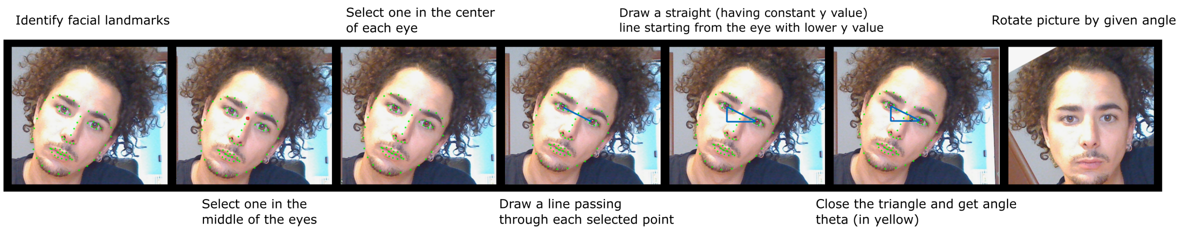

Identify facial landmarks
Select one in the center of each eye
Draw a straight (having constant y value) line starting from the eye with lower y value
Rotate picture by given angle
Select one in the middle of the eyes
Draw a line passing through each selected point
Close the triangle and get angle theta (in yellow)

Figure 4.3: The proposed alignment pipeline.

To align the face image, mainly three landmarks are utilised: two marking the center of the eyes and one placed in-between both eyes. Given these landmarks, a straight line is identified starting from the landmark on one eye and going through the one in the middle and ending on the last landmark on the second eye. Another line is drawn from the end of the previous one to the x coordinate of the first eye, while having a constant y value. Finally, the triangle (i.e. the three identified points) is closed and the angle θ is evaluated as shown in Equation 4.1 - where x is given as all sides are known.

$$\begin{aligned} Opposite/Hypotenuse &= x \\ sin(\Theta) &= x \\ \Theta &= sin^{-1}(x) \end{aligned} \tag{4.1}$$

As previously mentioned, face landmark positioning plays a great role in the picture alignment. Consequently, the face detector originally used to detect the face also becomes of big influence, as it sets the base for the positioning of face landmarks. Given the landmarks, the different performances in terms of alignment are displayed in Figure 4.4. From the latter image, it is possible to see how face-alignment results only change in terms of output image resolution, and no major visual difference can be seen on how the alignment is performed on the examples provided. Accordingly, not much accuracy was given to the landmarks placement, as the focus was placed on how often the detector fails to assign landmarks on the provided image (i.e. find a face in each image).

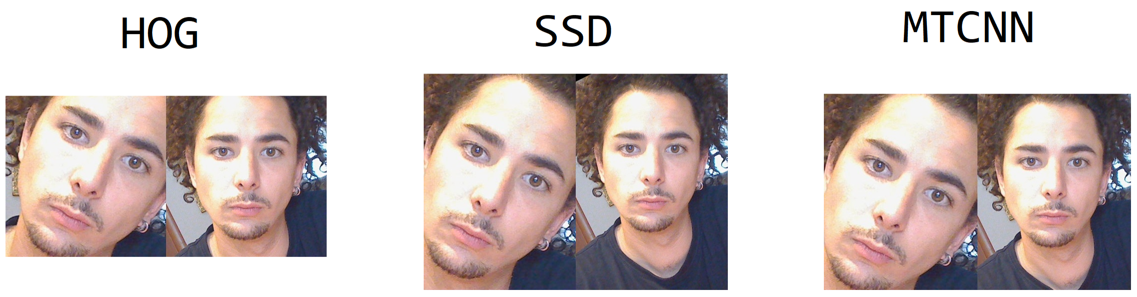


Figure 4.4: A visual comparison of how the alignment is performed given that the landmarks are set on the basis of each face detector considered.

After the alignment, the detected face is projected from RGB onto the YUV chromatic space. As also stated on the relative Wikipedia page, YUV is a colour-coding system typically used as part of a colour image pipeline. It encodes a colour image or video by taking human perception into account, allowing reduced bandwidth for chrominance components and typically allowing transmission errors or compression artefacts to be masked more efficiently by human perception than when using a "direct" RGB representation. The YUV model defines a colour space in terms of one luma (Y) and two chrominance (UV) components. This encoding intensifies the luminance aspect, which is exactly what the neural network should focus on - as this enhances the way skin pixels change when the heart beats. The transformation from RGB to YUV is formalised in Equation (4.2).

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \tag{4.2}$$

Once the face has been projected onto this new chromatic space, it is split into N regions of interest. If the size of the output image is not truly divisible by the chosen number of ROI's, an average pooling padding is applied.

4.1.3 Eye region and background removal

The literature also suggests two additional steps to be beneficial for a better HR prediction from images: the removal of both the eyes and the background.

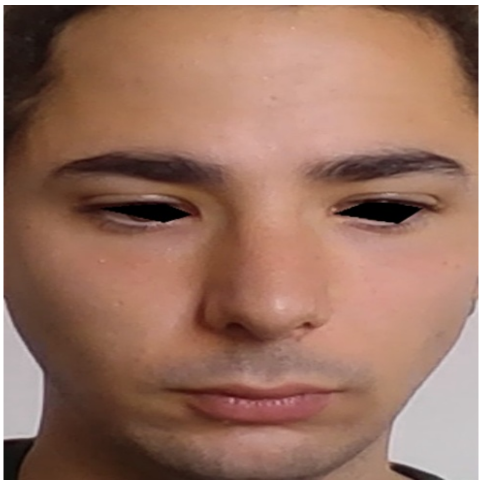


Figure 4.5: Example frame containing a face after the removal of the regions around the eyes.

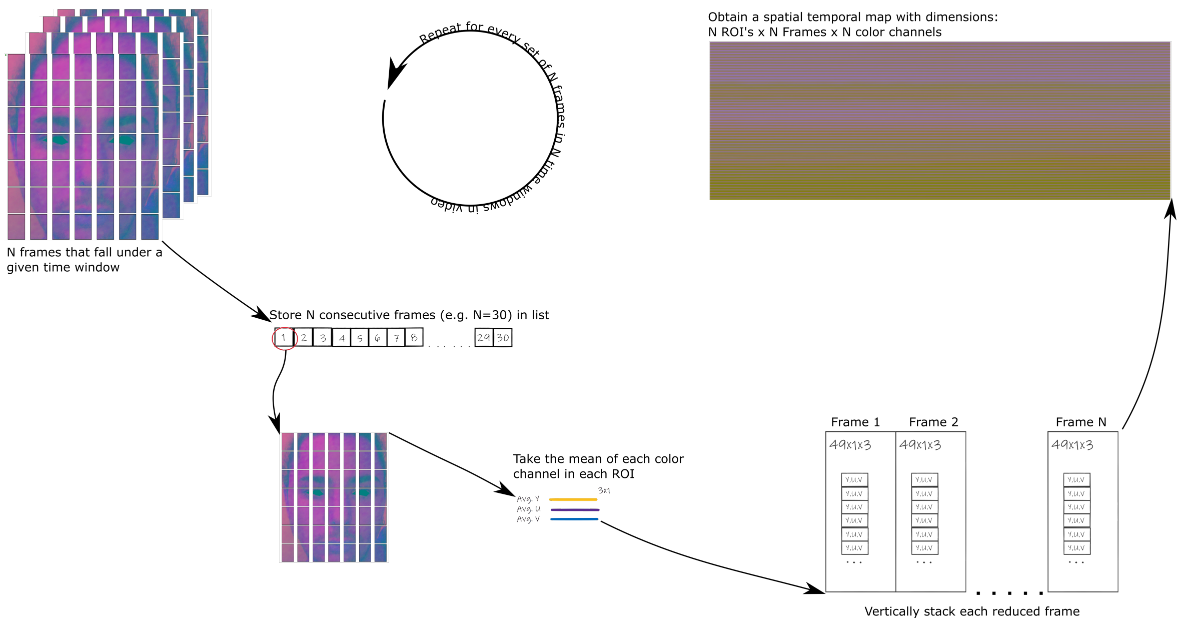
Though, no accurate and real-time method to remove the background has been found yet. Even when using a deep learning approach, the model was too slow on the CPU. Additionally, if we look at the segmented image that is produced by the last-mentioned step (Figure 4.5), the ratio $backgroundpixels/facepixels$ is extremely low - hence this step has been discarded altogether.

The same landmarks that were used to align the face have been employed to remove the eye region (i.e. the landmarks indicating where the eyes are in the image). More specifically, the region around the eyes is detected by taking the convex hull between the landmarks marking the eye region. The region described by the convex hull is the region to be removed (i.e. set to black). The convex hull of a subset A of a real vector space is called the intersection of all the convex sets containing A or, equivalently, the smallest convex set containing A. The convex hull of a convex set coincides with the set itself. An example of the convex hull of the landmarks marking eye regions can be seen in Figure 4.5

4.2 Map creation

It is important to stress that each map encodes a given number of frames that fall into a certain time window. This time window is sliding and overlapping between frames. Better explained, let the window size be 10 seconds, the frame rate be 20fps and the step size be $0,5$. The first map will include information from the very first frame to frame number 200 (*fps x window size*). The origin frame of the second map will then be shifted by 10 frames (*step size x fps*) and will include all the frames between 10 and 210. This means that every map has an overlap with the adjacent one (or ones) of 190 frames (*fps x window size - step size x fps*) or $9,5$ seconds. This overlapping fashion works well for two reasons: first, because it works as a sort of data augmentation (more images equals more training data) and second, because it makes the model understand the relationship between adjacent frames by always looking both at the new data and the previously seen data. More details regarding the choice behind window overlapping are provided in Section 4.4.
Because of the major overlap between maps, the pre-processing steps are first performed for all the frames and then the map creation starts. This way, it does not make the same computations on the same data.

Once the aligned and projected face from each frame has been divided into regions, the mean of each color channel is evaluated for each region of interest (each ROI evaluating to a scalar per



Repeat for every set of N frames in N time windows in video
Obtain a spatial temporal map with dimensions:
N ROI's x N Frames x N color channels
N frames that fall under a given time window
Store N consecutive frames (e.g. N=30) in list
Take the mean of each color channel in each ROI
Avg. Y
Avg. U
Avg. V
3x1
Frame 1
Frame 2
Frame N
49x1x3
Vertically stack each reduced frame

Figure 4.6: In the second step of the pipeline, the nxn blocks resulting from the previous stage are considered and the mean per color channel is taken in order for each block to evaluate to a $3x1$ array (one scalar per color channel). The arrays per frame are thereafter flattened and horizontally stacked one next to the other. This results in a final map having dimensions $TxNxC$ (number of frames, number of ROI's and number of color channels).

colour channel, i.e. $NROI'sx3x1$). The mean resulting for each frame over a given temporal window (i.e. N frames) is then concatenated, so to form a final map with dimensions $TxNxC$ (number of frames, number of ROI's and number of colour channels).
This second phase of the pipeline is depicted in Figure 4.6

Essentially, the map produced has information about both time and space. Additionally, because each map has an overlap with the next one, the map also delivers all the details that tie two adjacent frames together.

The final map will be the input to the network described in Section 4.3, the network will be able to both see humans and their movements, all encoded in a single image. This procedure has the main advantage to be simple. In fact, only a CNN is needed to regress over a time period. Generally, techniques such as 3D CNN's or RNN's are employed when dealing with video data. The notion of these spatial-temporal maps saved a lot of work and computation, that might have been redundant after all.

4.3 NrPPG-NNET

The next sections will elaborate on the creation of the system produced during this thesis work (i.e. NrPPG-NNET). The aim of these sections is not only to detail the training parameters and procedure, but it also provides insights on the challenges faced during the development.

4.3.1 Architecture

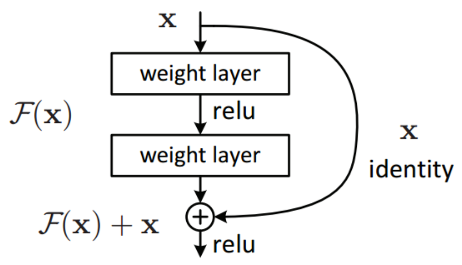
The architecture of the neural network employed in this research consists in a pre-trained ResNet50 Neural Network, presented in He et al. 2015. Residual neural networks (or ResNet) are a particular type of CNN devel, which uses an innovative type of block - the residual block - and exploits the concept of residual learning.

ResNet was born out of a simple observation: "why is it that adding additional layers to deep neural networks does not improve accuracy but, in fact, worsens it?"
It is known that in the Deep Neural Network a training algorithm defined as backpropagation is commonly used, coupled with an optimisation method (e.g. Stochastic Gradient Descent). In this way is possible to calculate the gradient of the cost function, which indicates how much our network is wrong in the predictions. The goal is to calculate the value of all the parameters such that the cost function is minimised, i.e. to reach the global minimum point found by the given optimization algorithm. Once the gradient is evaluated, the network weights are updated and the error back-propagated. However, if the number of levels increases, the gradient mathematically calculated with the help of the chain rule can become:

1. Very large (exploding gradient), causing problems of instability and generating parameters (weights) that exceed those manageable by the computer, giving rise to NaN values not further updatable.

2. Extremely small (vanishing gradient), determining a minimal update of the weights and causing a slowing down of the training process.

In a ResNet the process of computing the gradient is similar, but a trick to speed it up and increase its efficiency is employed. Instead of waiting for the gradient to propagate back one block at a time, the skip connection path allows it to reach the initial nodes effectively by skipping the intermediate ones. This gave rise to the famous ResNet blocks, depicted in Figure 4.7.



x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4.7: Example of a residual block (source).

Each ResNet "block" is made up of a set of layers and an identity mapping that connects the block's input to its output. This "addition" process is carried out one element at a time. Zero-padding or projection techniques can be used to achieve equivalent sizes when the input and output are of different sizes.
This allowed for the built of huge networks with hundreds of layers without having to worry about the gradient vanishing while travelling back to the initial layers as it could travel back directly via these shortcut connections.

Another component that is widely used in ResNets is represented by the layers of batch normalization, used after each convolution and activation. The batch normalization is an operation that allows to normalize the data present in the mini-batches and, thanks to this, to reduce the limitations on the learning rate value that typically exist in the training of deep neural networks. It also makes the weights initialization phase less complex. All this brings to a notable reduction of the time needed for the network training process. All in all, the central idea in ResNet paper is that, in the construction of a neural network with an elevated number of levels, the representation of the input data should remain as unaltered as possible - going deep into the network so as to preserve the information.

In this work, a pre trained ResNet18 was used as a feature extractor. The network is pre-trained on the ImageNet dataset and fine-tuned for our specific map domain created from the COHFACE dataset (by Heusch, Anjos, and Marcel 2017), following the procedure in Section 4.2. The network follows the same architecture as a ResNet18, but the last layer is removed and a Flatten and Dense layer are inserted as a *head* instead. The final dense layer basically just performs matrix multiplication to result in an output matrix, with a desired last dimension to be 1 (as this problem can be modelled as a regression task).

4.3.2 Training

As mentioned, the network input is an image where the width is dependent on the number of ROI chosen (subdividing the image in $5x5$ ROI's means the width will be of 25).

This actually induced a not so trivial problematic: the pre-trained ResNet18 that has been chosen has a minimum input size of $35x35$, so the number of ROI's could not follow the one proposed in the literature of 5. After some experiments, **the optimal number for the regions of interest has been evaluated to 49 (i.e. $7x7$)**. Additionally, it is worth saying that scaling up the image was not an option as it has been mentioned by previous researchers (especially Mironenko et al. 2020): scaling the images makes them loose important information as it induces noise when the scaling function (e.g. interpolation) is applied.

Furthermore, an analysis of the optimal batch size has been carried out and results showed improvements when going from 32 to **64, so this was selected as the final batch size**, with an **initial Learning Rate of 0.001** We could notice a major improvement when **removing the eye regions**, following the procedure stated in Section 4.1.3. The improvement of 1.5 in MAE hinted that not only the eyes were not carrying any relevant information towards the estimation of the HR, but also that they were inducing noise or in any way confusing the model predictions. An intuitive explanation for this could be that the reflectance property of the pupil shed light in ways that mislead the model. The model has been trained for **100 epochs** and **MAE has been used as a loss function**, and MAE has already been formalised in Equation (2.1).

4.3.3 Shuffling

It is not ideal for a neural network to look at the data in the same order it has been generated. This is because many samples from one class or category are put as first when generating the data. If the data is then split in train and test sets, an over representation of the first category might be induced.
In our instance, if all men were to be put first and the model was to read the data in sequential order, the validation on a woman subject (never seen before) would fail. This is not due to the model's inability to generalise (i.e. it has over-fitted on males), but it is rather due to the impossibility to learn features that are peculiar to both sexes as it has only been proposed one while learning.

An effective solution to this problem is to shuffle the data before feeding it to the network. In the case of NrPPG-NNET the input is modelled as a series of videos (around 4 videos per subject). After being processed, each video is transformed in images (spatial-temporal maps), so for each subject we now have different folders with, in turn, several images. The shuffling in this case needs to be done pertaining the sequentiality of one image after the other (within the same video), as well as paying attention to not give videos of the same subject during training and testing. In fact, doing such mistake could possibly make the model overfit on some subjects (already seen during training time), for which it will be better at predicting HR values in the validation phase.

Given the above, the data shuffling is carried out by shuffling *by subject*, so to avoid the aforementioned issue. Additionally, a thought criterion has been applied when doing the data split (train / test) so to avoid imbalanced classes representation in the two sets. An overview of the split is given in Section 4.3.4. Note that for reproducibility purpose, when random shuffling a seed has been set to 12.

4.3.4 Data balance

As mentioned in Section 3.4, the data at hand has quite a disparity in both gender and skin tone. In order to provide the fairest representation possible for each category, the data has been manually split so to make the model see both men and women of any skin tone in both the train and the validation set. The distribution of the two sets is depicted in Figure 4.8.

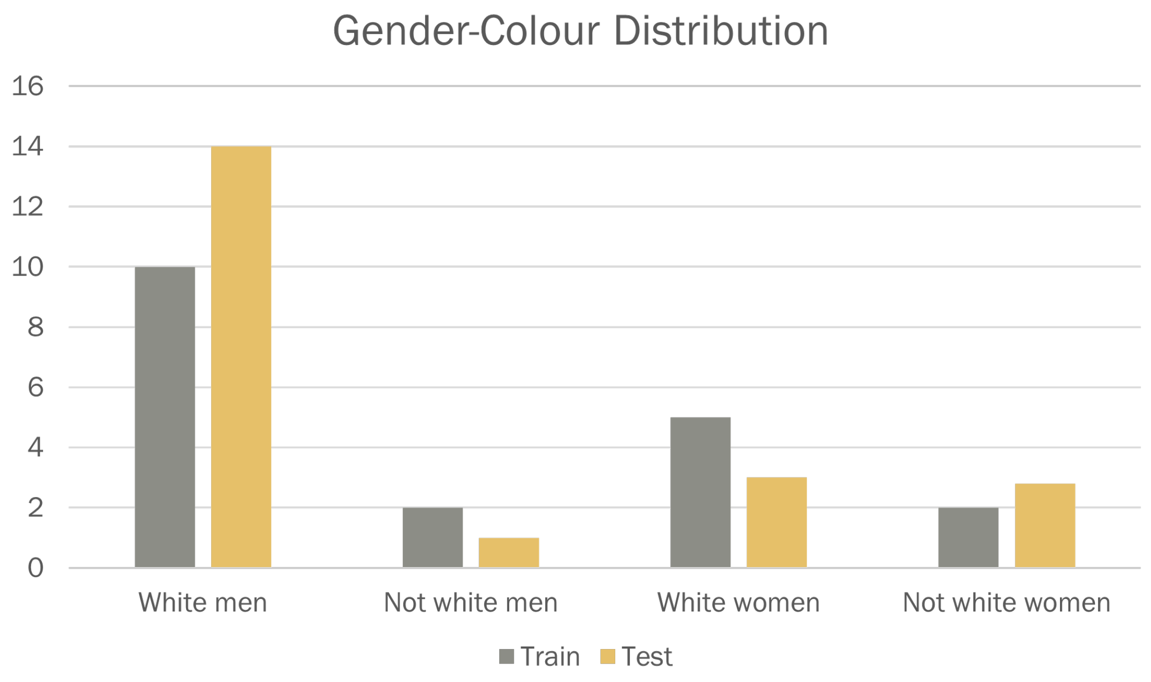
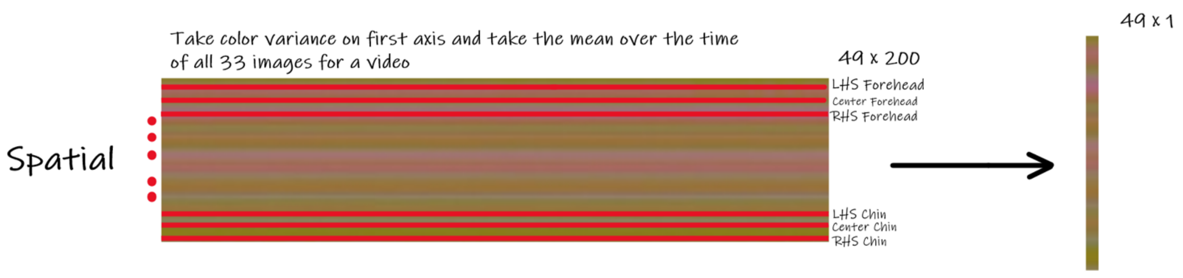


Figure 4.8: Distribution of subjects gender and skin colour when split in train and test, following a criterion so not to have over-represented categories.

4.3.5 Weight Map

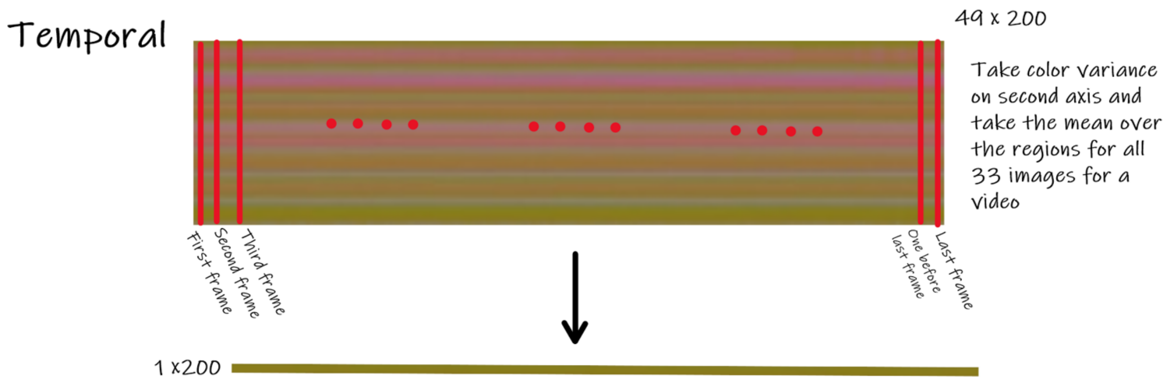
From literature, it becomes clear that some regions of the face hide more information about the HR than others (K. Zhang et al. 2016, Niu, Shan, et al. 2020, Speth et al. 2021). Specifically, regions such as the forehead and the cheeks tend to take more information because of the strong presence of flat skin. Given the flatness of the surface and the concentration of blood vessels underneath, when the light is shed on these regions at the time of a heartbeat, the colour change (due to the increasing volume of blood) is more evident. Human eyes cannot still perceive such changes, but computer vision can. Here the variation of chromatic and brightness values are correlated with the variation in HR.

Stemming from previous research, an analysis has been performed on the regions that vary the most in terms of color change. In better details, the variance has been evaluated on each of the 49 ROI's ($7x7$) for all the video frames.



Take color variance on first axis and take the mean over the time of all 33 images for a video
49 x 200
LHS Forehead
Center Forehead
RHS Forehead
Spatial
LHS Chin
Center Chin
RHS Chin
49 x 1

Figure 4.9: How the variance has been evaluated in space dimension of the map, for all the 33 maps of a video, for all the videos.



Temporal
49 x 200
Take color variance on second axis and take the mean over the regions for all 33 images for a video
First frame
Second frame
Third frame
Last frame
One before last frame
1 x200

Figure 4.10: How the variance has been evaluated in time dimension of the map, for all the 33 maps of a video, for all the videos.

More specifically, the process of evaluating the variances starts with all the spatial-temporal maps of a video (e.g. for a 60s video there are around 100 maps). A single map represents only $0,5s$ and it is not plausible that a relevant variation in HR happens in such small time frame. Hence, the mean of every 3 maps is evaluated, resulting in one video being represented by around 33 spatial temporal maps. Each map has a number of frames (or time) on the horizontal dimension and ROI's (space) on the vertical dimension. For instance, the first row of the map represents how the first ROI evolves over the time frame captured by the map. Conversely, the first column represents all the ROI's at the very first instant in time.
The variance is then taken on both these axis for all the 33 maps of a video, for all the videos in the dataset. A more visual explanation of this is depicted in Figure 4.9 (for space dimension) and Figure 4.10 (for time dimension).

It is interesting to note how some regions vary more than others. A causal relationship has been assumed between these regions chromatic variance and the information about HR. The assumption is based on the knowledge borrowed from literature as well as the correspondence between the regions carrying more of the HR information and the regions that seem to vary the most in the videos presented.
In figure Figure 4.11, it is possible to note how there is an almost periodic increase in variance, where on the x axis there are the 49 regions of interest. The periodicity is due to the fact that not all regions vary in the same way, but some do more than others. Additionally, it is worth to note how it is mainly the first channel (in YUV chromatic space) that varies with such frequency. The intuition behind this could be that the first channel is actually the one carrying the luminance factor, so it is not surprising that the channel Y specifically varies more than the other two in certain regions. More specifically, the regions where most of the variance is encountered are the ones of the forehead

(3,4,5 and 10,11,12), on the right cheek (23,24) and the left cheek (26,27).

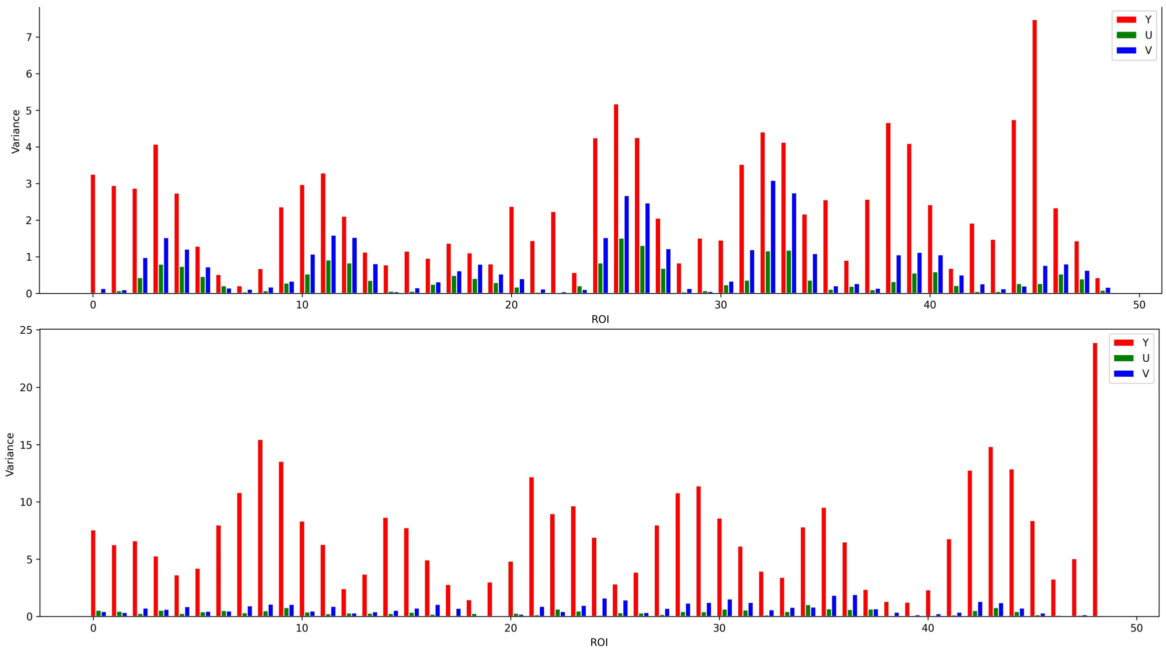


Figure 4.11: Two videos sampled at random from COHFACE have been analysed in terms of per ROI variance. The variance has been evaluated on all three channels of the YUV chromatic space.

After having obtained these results a more pleasant visualisation of said variation has been created so to better match face regions and color variance. The visualisation can be seen per channel in Figure 4.12. It is interesting to note how most of the variance is carried by the first channel in the regions that are assumed to be the ones conveying the HR information. For this sampled video, also the V channel seems to be playing a role in highlighting important regions (such as the forehead ones). This can be explained by this channel ability to capture not just the luminance variance but also the variation on a chromatic level. The second channel (U) on the other hand does not seem to deliver any useful information.

The information on the variance has been used as a weight map for weighting the input of the neural network. The idea behind this design choice is that the model could learn quicker where to look. This intuition was confirmed and it will be shown in Chapter 5. Unfortunately, the model with such weighted input never outperformed the one without this *weight map*, but this was an interesting concept to explore and definitely deserves future work to be done on it.

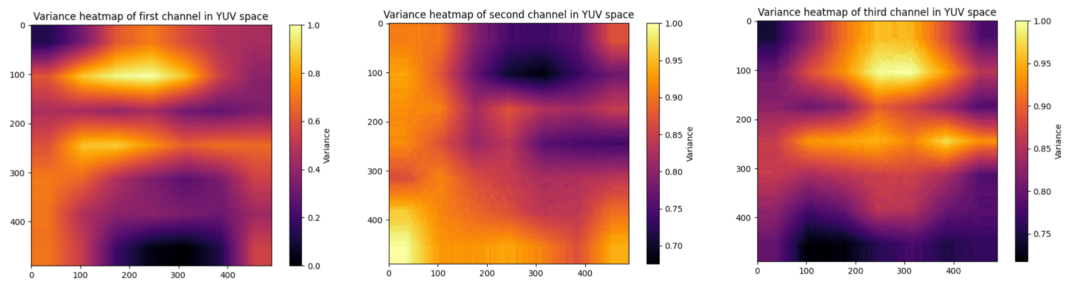


Figure 4.12: Variance for a sampled video in COHFACE, plotted as to recompose the original frame and segregated per color channel (Y, U, V).

4.4 Challenges faced

The main challenges faced during the development have to do with some details missing from the paper on which the architecture of the project is based (i.e. Niu, Shan, et al. 2020). These challenges are here reported as a numbered list.

1. **Overlapping windows**: As discussed already, one created map corresponds to a set of frames that falls under a given time window. When the map is created, the window is moved by a step and the process is repeated. The issue here is that Niu, Shan, et al. 2020 never claimed that the window was moved from the starting or the ending point of the previous one - boiling it down to either having or not having an overlap. These two ways of creating a moving window can be visualised in Figure 4.13, where the overlap in question is the coloured area for each step and in Figure 4.14 where there is no overlap and the step between frames is colored in green.
   This issue has been overcome by looking at the data obtained by the researchers after the processing (i.e. the number of maps given a certain number of frames). Because there is a substantial majority of maps over a number of frames, this induces the reader to think that the window is shifted with an overlap taken into account. Accordingly, this work implements the sliding in the same manner.

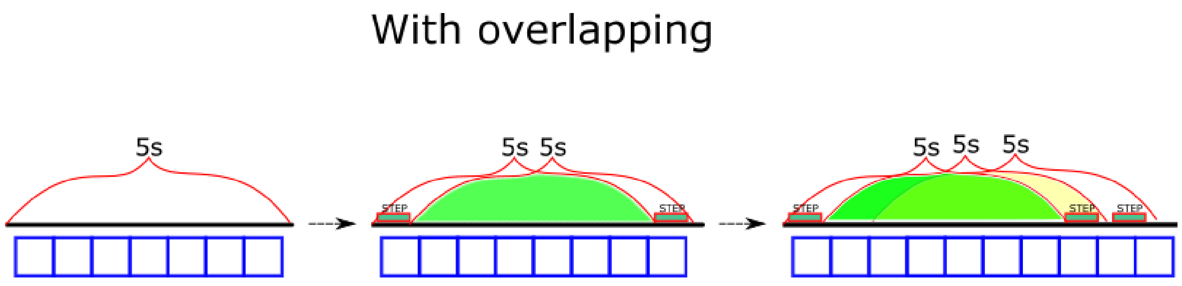


Figure 4.13: Moving a sliding windows with an overlap.

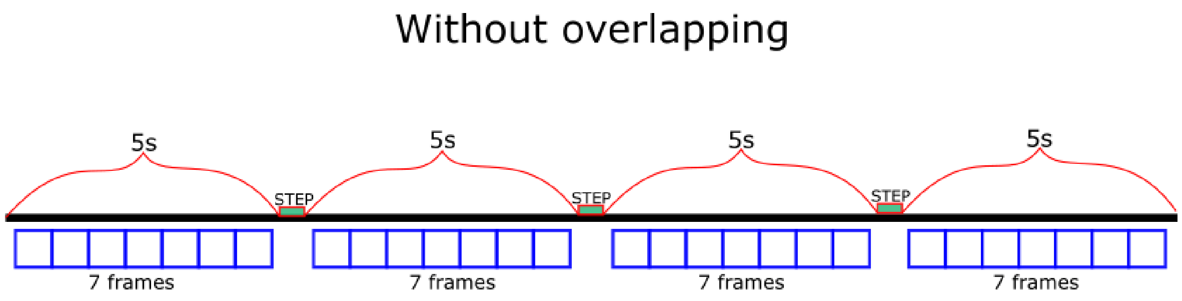


Figure 4.14: Moving a sliding windows without an overlap.

2. **Number of ROI's**: When looking at the previous publication of the authors, a second more predominant issue has been raised. In fact, a previous paper (namely Niu, Han, et al. 2018) offers a slight implementation variation. The discrepancies found mainly concern the number of ROI's. This issue was not tackled as there is no one way to understand whether the authors have changed the method during all those years or if they just assumed that other researchers had read their previous work. In fact, different numbers of ROI's have been tested, and their impact on the model performance has been evaluated accordingly, as detailed in Section 4.3.2.

3. **Synthetic Data**: Another discrepancy presented in the previous work from authors and the newer one is a pre-training run on synthetically generated data. This training is not mentioned in the latest work, but it is explained in the former one. This is why it has been assumed that authors would have at least mentioned it in the most recent paper - as it is a complex process to realistically fake an ECG signal (or a signal in general). Furthermore, even preliminary performances result have proven it to be unnecessary - and therefore disregardable.

Chapter 5

Results

This chapter wants to explore the results obtained by NrPPG-NNET, putting them into context with other methods, and benchmarking them as fairly as possible by using the same metrics and the same validation data.

In Section 5.1, results are presented and different versions of the model are taken into consideration. Additionally, improvements between one version and another are detailed and explained. In Section 5.2, the model will be compared to other methods that utilised the same dataset and metrics for evaluation.

5.1 Results

As previously mentioned in Section 2.3, the aim of NrPPG-NNET was not only to be as accurate as possible but to also satisfy requirements such as being lightweight and working in real-time. Let us assess all these characteristics in different subsections. Firstly, the time taken by the overall system is reported in Section 5.1.1. Additionally, model size and resources consumption are reported in the same section. Secondly, Section 5.1.2 reports the accuracy of the system in terms of prediction.

5.1.1 Speed

As shown in Table 5.1, OpenCV renders a frame from a laptop webcam (Full HD, 2.07 MP, 17.54 MB/s bitrate) in a little over $20ms$. A minor difference is perceived when NrPPG-NNET and all the input pre-processing (i.e. map making, face cropping etc.) are run on a GPU in battery saving mode, but it goes back to the $20ms$ range when the GPU runs at full speed (without overclocking).. In this case, tests have been performed on a GTX 1070 Laptop GPU and an i7 Intel 11th generation CPU. Overall, the time difference is minimal even when running solely on the CPU and no delay can be perceived by the naked eye, making the system function in real-time. This is a great achievement as a lot of matrix operations are performed not only during inference, but also to manipulate the input to suit the Neural Network input layer.

Speed	Specific
38.7fps	w/o Nrppg-NNET
22.7fps	CPU
17.8fps	GPU battery saving
35.4fps	GPU

Table 5.1: This table shows the time taken to render a single frame without NrPPG-NNET, with it being run on a CPU, on a GPU on power saving and on a normally functioning GPU.

Additionally, the lightweight of the model is also satisfying when cross-checking it with the requirements set before development. The model only weights $131,392KB$ on disk space and takes around $2GB$ of RAM when executed. RAM usage has been evaluated by the difference in machine RAM usage when idling and when running the program.

5.1.2 Accuracy

NrPPG-NNET reaches in its final version reaches an MAE of 7.288, given that MAE is expressed in the same units as the original data, it means the model can predict Heart Rate with an error of ± 7.288 beats per minute. As previously mentioned in Section 2.1, a clinical device can always guarantee an MAE of no more than 3 BPM. NrPPG-NNET does not qualify to be used in clinical standards nor do any of the methods previously developed in literature.

It is worth noticing the numerous versions of the model shown in Table 5.2. Originally a naïve implementation saw the model having the full frames as an input and no research on the network parameters. As noticed in Section 4.3.2, different training parameters changed during the development, yielding better results most of the time. Firstly, the input of the network caused some issues - as noted in Section 4.4, there was a minimal length and width the input needs to have in order to be fed to a pre-trained ResNet. The input shape originally was the same as the input of the pre-trained network ($224x224$), but this also implied that the images were stretched using bilinear interpolation (first applied to the image resizing domain by Ke-jian 2008). Although this method is highly reliable, it still adds data to the input which wasn't in the original image and could both induce noise and confuse the network. For this reason, the number of ROI's (that directly controls the height) has been increased from $5x5$ to $7x7$. This is mainly because the minimum size of the ResNet input is $32x32$. Having $5x5$ ROI's results in an input image of 25px height, whereas when increased to $7x7$ it evaluates to 49px height. This change in input size reduced the final MAE from 11.634 to 9.167.

Version	MAE
Original	11.634
Changed input shape	9.167
Removed eyes	8.972
Chuncks Shuffled	8.455
Fine tuned	**7.288**
w/ Variance Map avg.	8.456
w/ Variance Map single	8.55

Table 5.2: This table shows different model versions: from the first stable one to the last fine-tuned and changed in parameters and input size.

Additionally, the pre-processing step involving eye region removal has been performed on the input (this procedure is better detailed in Section 4.1.3). Intuitively, eyes were not taking any useful information because there is no blood vessel going through them. Additionally, the extreme reflectance property of the pupils (due to them being always moist) might have confused the model on where to focus and how to interpret that light variation. When eyes have been removed, the MAE dropped to 8.87.

As explained in Section 4.3.3 and in Section 4.3.4, it was most likely the case that train and test sets were both balanced in terms of gender and skin colour. Shuffling helped the model ability to learn as it made it see videos of different subjects all at random (not necessarily N videos from one subject in a row), while manually balancing the data ensured that the model would be able to generalise among different genders and ethnicity. After these two operations on the dataset, the MAE went down to 8.455. This was the last change made to the model training procedure overall and the selected training run can be seen in Figure 5.1.

Using a pre-trained network means that during training, all layers of that network are frozen (i.e., not learning). Usually only the last layers added are actually change their weights. The frozen layers are still used to extract features and approximate the objective function, but actually only the

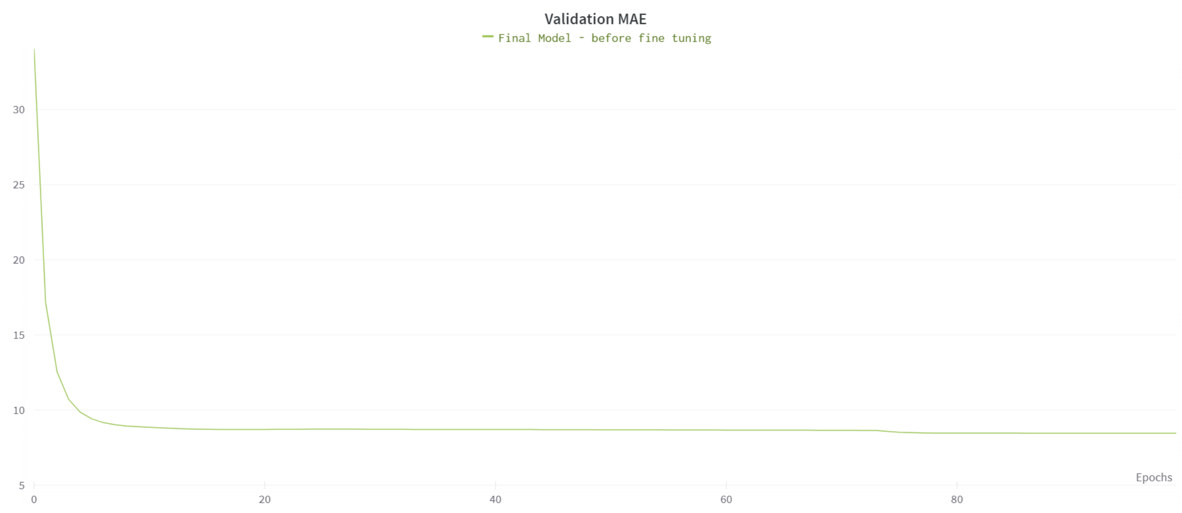


Figure 5.1: The best run of the model that was picked as a final one with parameters explained in Section 4.3.2. Final MAE reached is: 8.455

last layers adapt to the new task. In the case of NrPPG-NNET, the pre-trained ResNet18 was trained on ImageNet. The last flattening layer and a dense layer (with one-dimensional output given the regression task) adjusted their weights according to the HR detection task. After selecting the best model and training its final layers, it is possible to adapt the previously frozen layers to the new domain. This procedure is very common in the field of transfer learning (i.e., adapting a network from one task to another) and means that the weights of all the layers of the pre-trained model are unfrozen and made trainable. To avoid over-fitting to the new task, the learning rate was changed and set to a very low number (in the case of NrPPG-NNET, this number is $1.0e-7$).

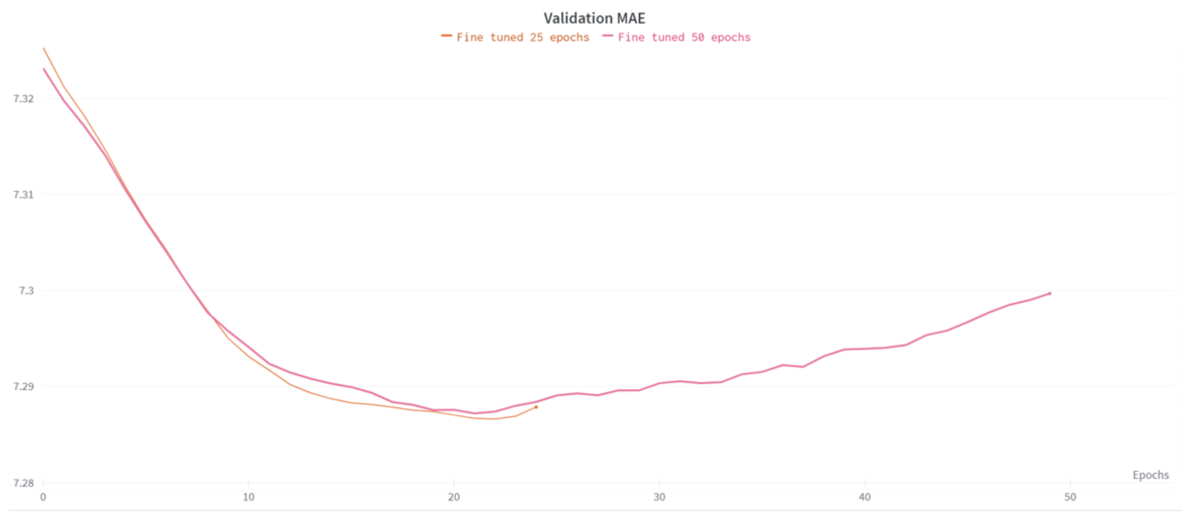


Figure 5.2: The two runs shown are the fine-tuning ones: the red one has been training for 50 epochs, while the orange one only for 25. Finally, the orange has been selected as the final model yielding an MAE of 7.288.

According to the literature, the network has been trained for just 10 epochs during the fine-tuning, but more epochs numbers have been tested as this did not show convergence. In figure Figure 5.2, it is possible to see how after around 15 epochs the MAE tends to plateau (the wiggliness of the line is due to the scale of the y axis). Conversely, it seems that after 50 epochs the MAE starts rising up again, hence the model which training is displayed in orange is the one selected as the best one, achieving a final MAE of 7.288.

Additionally, during the last stage of development, the idea of weighting the input based on a so-called *variance-map* has been implemented. The map is obtained following the procedure explained in Section 4.3.5. The resulting variance map is min-max normalised and the weights (ranging from 0 to 1) are element-wise multiplied (i.e. Hadamard product) to the input of the neural network. The weight map has dimensions $7x7$, but it is flattened so to obtain a column vector, which is multiplied by each column of the input matrix. In fact, the input matrix is sized $49x200x3$.
A simplified example of this procedure is given in Equation (5.1), where column vector $\vec{W}$ is a flattened input of weights and matrix I is the input matrix (a whole spatial-temporal map).

$$\vec{W} \odot I = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \odot \begin{bmatrix} i_{1,1} & i_{1,2} & i_{1,3} \\ i_{2,1} & i_{2,2} & i_{2,3} \\ i_{3,1} & i_{3,2} & i_{3,3} \end{bmatrix} = \begin{bmatrix} w_1 i_{1,1} & w_1 i_{1,2} & w_1 i_{1,3} \\ w_2 i_{2,1} & w_2 i_{2,2} & w_2 i_{2,3} \\ w_3 i_{3,1} & w_3 i_{3,2} & w_3 i_{3,3} \end{bmatrix} \tag{5.1}$$

The input is weighted with two different versions of these variance maps. The first version is evaluated by taking the variance over one single video, the other one is the average of all the variances of all videos in COHFACE. Evaluating on one single video finds its explanation in the stillness of the subjects in the dataset. As explained in Section 3.3.1, the subjects don't move much and they are all around the same point in the screen space. Additionally, because the same face detector is used to crop the subjects' face regions, the mapping *ROI to face section* will be consistent throughout the all the subjects.

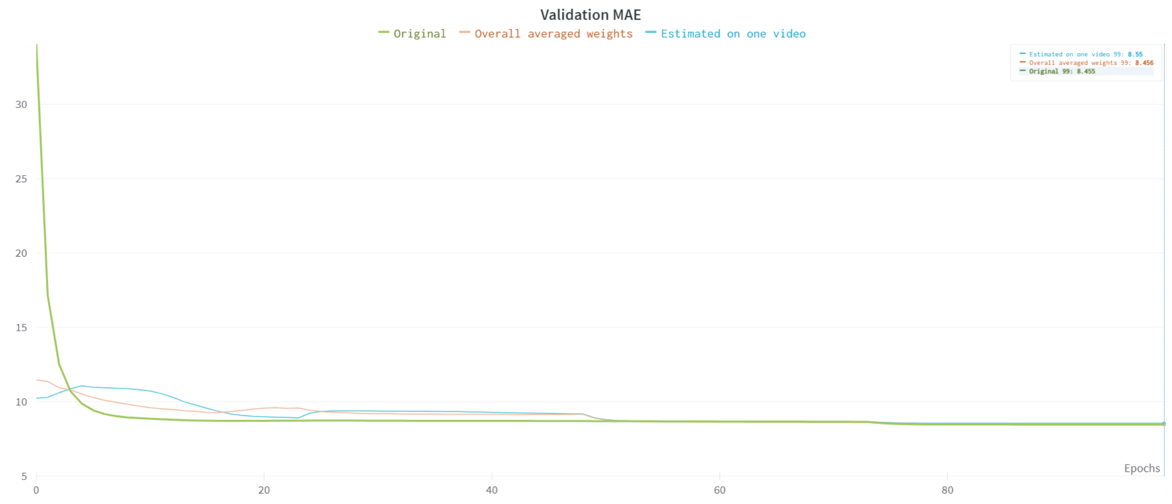


Figure 5.3: In the graph above, three runs are shown. The blue and orange ones have a weighted input based on the procedure explained in Section 4.3.5. The green one is the previously trained model, with no weight applied on the input.

When training with these two versions of the variance map, the results are as shown in Figure 5.3. All in all, the three runs achieve approximately the same result. As stated in Table 5.2, the model that yields the lowest result (lower only by .01) is still the one that has been trained without the variance map.
Though it is interesting to note the behaviour the models have during the training run. The two models trained with a weighted input seem to start with a significantly lower MAE. On the other hand, the model trained without the variance maps starts with an MAE of around 30 bpm and steadily decreases in the first 5 epochs or so. Intuitively, this behaviour could be explained in that the regions that carry the most information are already emphasized, whereas it needs to learn what regions to focus on when it comes to the model without pre-applied weights.
The model without variance maps is selected as the final one and the reason is twofold: first (of

course) because it yields the best MAE. Secondly, because it is better for the model to learn to not pay attention to some of the input data rather than being manually forced to do so.

5.2 Benchmarks

In terms of a quantitative assessment of how NrPPG-NNET performs, it is fair to claim that the final MAE score is not the best when compared to methods using the same metrics and the same data.
More specifically, NrPPG-NNET qualifies as third-best among the methods proposed in the literature. As shown in Table 5.3, only two other methods outperform the one proposed in this work, and it generally performs better than the mean of all MAE's (i.e. 10.46).

Table 5.3: A comparison of several approaches' results in terms of mean absolute error in beats per minute (bpm). The root mean squared error - which is always higher than or equal to the mean absolute error - is represented as $<$ as no MAE was provided by the authors. The precision attained by constantly forecasting a heart rate of 75 beats per minute (average HR over COHFACE) is referred to as baseline.

Authors	Method	MAE
Us	Baseline (75bpm)	9.35
Haan and Jeanne 2013	CHROM	7.8
Li et al. 2014	LiCVPR	19.98
Wang, Stuijk, and Haan 2016	2SR	20.98
Špetlík 2018	HR-CNN	7.8
P. Zhang et al. 2021	Zhang2021	5.57
Gudi, Bittner, and Gemert 2020	FaceRppg	10.8
Us	**NrPPG-NNET**	7.28

Though, it must be noted that methods such as FaceRppg (Gudi, Bittner, and Gemert 2020), CHROM (Haan and Jeanne 2013), 2SR (Wang, Stuijk, and Haan 2016), LiCVPR (Li et al. 2014), are fully unsupervised - so the method employed was not trained on part of the COHFACE dataset. This could make the comparison slightly biased in favour of those other methods (including NrPPG-NNET) that learnt directly on a subset of the COHFACE dataset. In P. Zhang et al. 2021, there is no mention or description of the evaluation protocol. On the other hand, the model presented in Špetlík 2018 has been trained on its own custom dataset and performed validation on the whole COHFACE dataset. All these different evaluation protocols hint that a fair comparison would need more time to be performed. More validation data would need to be processed so to have a ground truth signal that matches the shape of the one provided by COHFACE. Unfortunately, such an evaluation could not be performed due to time constraints.

Additional empirical validation has been performed by matching the output predicted HR with the one predicted by an Android app (namely Heart Rate Monitor). The predictions error never exceeded the $\pm$7bpm and overall seemed to match almost exactly the output of the mobile app. Some recording of such validation experiments have been recorded and made available on a shared folder, for everyone to investigate. Videos show various scenarios (e.g. different light conditions and after sport activity). Additionally, videos show the ability of the network to not predict sensible values when no face is present in frame, in such a case the system outputs an "invalid" error message.

Even though the MAE does not prove to be the best, there are other factors to consider when evaluating these kinds of work. In fact, as previously mentioned, there are other factors that play a role in evaluating the quality of this system overall. As stated in Section 2.3, a crucial factor was

for the model to be as lightweight as possible, due to the possibility of it running on an embedded device in the future. The Neural Network was trained on a machine having a *GTX 1070 Laptop GPU* and a *i7 Intel 11th generation CPU*. Even though these specifics belong to a mid-high end laptop, in inference time the model makes a prediction in tens of milliseconds (full statistics are reported in Table 5.1).
When trying to compare NrPPG-NNET with the other published so far in the field, it has been hard to find proper benchmarks in terms of inference time, but some things are assumable. For instance, the method proposed by Yu, Li, et al. 2020 uses two high-end GPU's for training (P100 GPU). So it is fair to assume that the model needs a great amount of VRAM to run. Additionally, even Niu, Shan, et al. 2020 trivially produced a bigger model as they used both ResNet18 architecture *and* a Gated Recurrent Unit, so they will necessarily have a bigger final trained Neural Network. Lastly, Špetlík 2018 proposes a method that *requires at least 12GB of GPU memory* when tested (quoting from their GitHub repository). Many more methods do not present benchmarking in terms of space taken by the final model and resources consumed during the inference - so a fair comparison is not really possible.

Chapter 6

Discussion and Conclusion

The goal of this work was to develop a method that could predict HR based on video footage alone. In addition, the system had to run in real-time on the CPU and not take up too much memory. NrPPG-NNET meets all of these requirements as well as the other requirements proposed in section 2.3. The final MAE (i.e. 7.28 bpm) of the model compares well to other methods in the literature, although it does not reach the state of the art. The presented work also makes good use of spatial-temporal maps and manages to process the *map making pipeline* in real time. The potential applications of this model are outside the clinical domain, as the accuracy is not comparable to specialised medical devices. However, systems such as NrPPG-NNET are finding applications in HCI field. The model will be deployed as soon as possible to monitor users' HR using only the laptop's webcam.

Due to the lack of time, many different adjustments, tests and experiments have been postponed. As previously mentioned, validation was done differently for all methods found in literature. For better validation of NrPPG-NNET, the method would need to be tested on different datasets, and such validation may be performed in future work. In addition, further studies will focus on a more in-depth investigation of specific mechanisms. In particular, a better training procedure could be applied that includes a greater variety of data in terms of number and characteristics of subjects.

Additionally, the analysis explained in Section 4.3.5 could potentially lead to interesting results. Therefore, further research could be done on how variation in light and colour can be used to weight some facial regions. Understanding that these changes are present in the videos was easy, but really conveying what determines these changes and understanding the underlying function that better approximates the covariance of skin colours with HR might be a fundamental improvement.
Furthermore, the method could be extended to predict an entire signal rather than just the HR value every n seconds. Predicting signals is something that many authors in the field have attempted to do and that we believe could potentially be achieved with NrPPG-NNET. For example, a different formulation of both the ground truth and the spatial-temporal maps could lead the model to a more continuous regression of HR values.

Everything about this method has been described as precisely as possible so that it fully reproducible and allow for it to be extended by future researchers. In addition, the code for creating the maps, retraining and evaluating the model is available at this GitHub repository.

Appendix

7.1 Timeline

Each task takes a maximum of 3 weeks as the workload has been subdivided in many smaller steps. Additionally, thanks to the modularity of the approach selected, it will be possible for me to test each step as soon as I have finished working on it. Considering week one starting on the 26th of April 2021, we can calculate the remaining weeks until the 1st of November 2021. The number of weeks adds up to 27. In these 27 weeks, holidays are to be included and will be marked as such on the Grantt Chart shown in Figure 7.1.

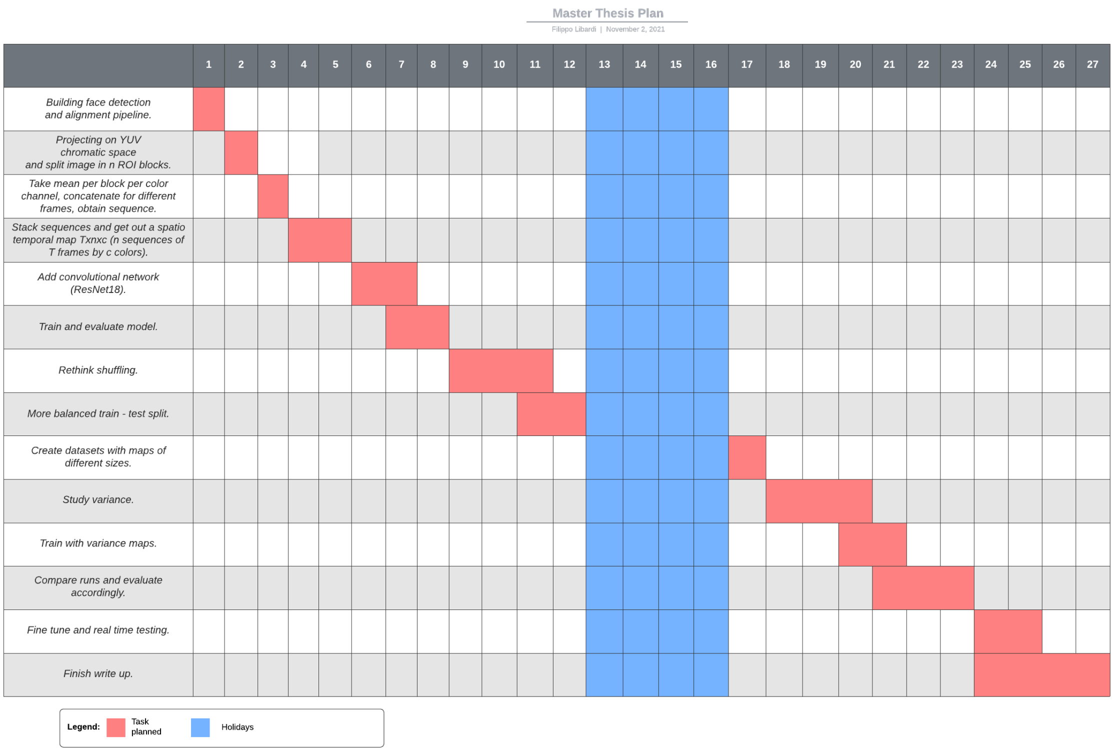


Figure 7.1: Timeline of the project.

7.2 List of Abbreviations

1. **NN**: Neural Networks or Artificial Neural Networks (ANN) are mathematical models composed of artificial neurons that are inspired by the biological functioning of the human brain.
2. **CNN**: A convolutional neural network (CNN or ConvNet) is a network architecture for deep learning that learns directly from data, eliminating the need to manually extract features.
3. **RGB**: The "RGB" color space means red, green and blue, the primary colors of additive synthesis.
4. **YUV**: YUV is a color space used for encoding images or videos. Designed to mirror the behavior of human vision, it allows for a reduced chrominance bandwidth, where Y provides the brightness information and U and V the color information.
5. **PPG**: PPG is a non-invasive optical measurement technique that can be used to detect blood volume changes in the microvascular bed of tissues.
6. **HR**: Heart Rate indicates the number of heartbeats per minute (BPM).
7. **RR**: Respiratory rate is defined as the number of breaths taken by an individual over the course of a minute.
8. **GAN**: Generative Adversarial Networks.
9. **LED**: A semiconductor diode that emits light radiation (Light Emitting Diode) when an electric current passes through it.

Acknowledgements

I hereby want to thank all the people whose assistance was a milestone in the completion of this project.
I would like to pay my special regards to the people from Teoresi and expecially to Vincenza and Annalisa. The positions covered by them, have proven to be fundamental for the development of this thesis project. Mostly, this is because they have both graduated with a Biomedical Engineer master degree, therefore have been bridging between my engineering knowledge and the medical field (in which I had little if no expertise).
Additionally I wish to show my gratitude to my supervisors within University of Utrecht for the support provided at any time. Their suggestions have been of extreme value for the realisation of this research project.
I would also like to acknowledge that part of the research in this paper used the COHFACE Dataset made available by the Idiap Research Institute, Martigny, Switzerland.

Bibliography

[1] Daejoon Anh, Subramaniam Krishnan, and Frank Bogun. “Accuracy of electrocardiogram interpretation by cardiologists in the setting of incorrect computer analysis”. In: *Journal of Electrocardiology* 39.3 (2006), pp. 343–345. ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2006.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0022073606000124.

[2] VV Barun and AP Ivanov. “Estimation of the spectral absorption of light by components of human skin”. In: *Optics and Spectroscopy* 106.1 (2009), pp. 84–91.

[3] Giuseppe Boccignone et al. “An Open Framework for Remote-PPG Methods and Their Assessment”. en. In: *IEEE Access* 8 (2020), pp. 216083–216103. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3040936. URL: https://ieeexplore.ieee.org/document/9272290/.

[4] Jorge Brieva, Hiram Ponce, and Ernesto Moya-Albor. “A Contactless Respiratory Rate Estimation Method Using a Hermite Magnification Technique and Convolutional Neural Networks”. en. In: *Applied Sciences* 10.2 (Jan. 2020), p. 607. ISSN: 2076-3417. DOI: 10.3390/app10020607. URL: https://www.mdpi.com/2076-3417/10/2/607.

[5] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)* 1 (2005), 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[6] Yoav Freund and Robert E Schapire. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504. URL: https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[7] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose”. In: *IEEE Trans. Pattern Anal. Mach. Intelligence* 23.6 (2001), pp. 643–660.

[8] Amogh Gudi, Marian Bittner, and Jan van Gemert. “Real-Time Webcam Heart-Rate and Variability Estimation with Clean Ground Truth for Evaluation”. In: *Applied Sciences* 10.23 (2020). ISSN: 2076-3417. DOI: 10.3390/app10238630. URL: https://www.mdpi.com/2076-3417/10/23/8630.

[9] Gerard de Haan and Vincent Jeanne. “Robust Pulse Rate From Chrominance-Based rPPG”. In: *IEEE Transactions on Biomedical Engineering* 60.10 (2013), pp. 2878–2886. DOI: 10.1109/TBME.2013.2266196.

[10] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[11] Guillaume Heusch, André Anjos, and Sébastien Marcel. “A reproducible study on remote heart rate measurement”. In: *arXiv* (Sept. 2017). URL: https://arxiv.org/abs/1709.00962.

[12] Rik Janssen et al. "Video-based respiration monitoring with automatic region of interest detection". en. In: *Physiological Measurement* 37.1 (Jan. 2016), pp. 100–114. ISSN: 0967-3334, 1361-6579. DOI: 10.1088/0967-3334/37/1/100. URL: https://iopscience.iop.org/article/10.1088/0967-3334/37/1/100.

[13] Yang Ke-jian. "An Image Scaling Algorithm Based on Bilinear Interpolation with VC". In: *Techniques of Automation and Applications* (2008).

[14] Justin M. Johnson and Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: 6.1 (Mar. 2019). DOI: 10.1186/s40537-019-0192-5. URL: https://doi.org/10.1186/s40537-019-0192-5.

[15] Vahid Kazemi and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees". In: (2014), pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[16] Davis E. King. "Max-Margin Object Detection". In: *CoRR* abs/1502.00046 (2015). arXiv: 1502.00046. URL: http://arxiv.org/abs/1502.00046.

[17] Sander Koelstra et al. "DEAP: A Database for Emotion Analysis ;Using Physiological Signals". In: *IEEE Transactions on Affective Computing* 3.1 (2012), pp. 18–31. DOI: 10.1109/T-AFFC.2011.15.

[18] Xiaobai Li et al. "Remote Heart Rate Measurement from Face Videos under Realistic Situations". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 4264–4271. DOI: 10.1109/CVPR.2014.543.

[19] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *Lecture Notes in Computer Science* (2016), pp. 21–37. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[20] J.-B. Martens. "The Hermite transform-theory". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.9 (1990), pp. 1595–1606. DOI: 10.1109/29.60086.

[21] D. J. McDuff, E. B. Blackford, and J. R. Estepp. "The Impact of Video Compression on Remote Cardiac Pulse Measurement Using Imaging Photoplethysmography". In: *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*. 2017, pp. 63–70. DOI: 10.1109/FG.2017.17.

[22] Yuriy Mironenko et al. "Remote Photoplethysmography: Rarely Considered Factors". en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Seattle, WA, USA: IEEE, June 2020, pp. 1197–1206. ISBN: 978-1-72819-360-1. DOI: 10.1109/CVPRW50498.2020.00156. URL: https://ieeexplore.ieee.org/document/9150582/.

[23] Jermana Moraes et al. "Advances in Photopletysmography Signal Analysis for Biomedical Applications". en. In: *Sensors* 18.6 (2018), p. 1894. ISSN: 1424-8220. DOI: 10.3390/s18061894. URL: http://www.mdpi.com/1424-8220/18/6/1894.

[24] S. Nagaraja and C. J. Prabhakar. "Low-Level Features for Image Retrieval Based on Extraction of Directional Binary Patterns and Its Oriented Gradients Histogram". In: *CoRR* abs/1503.03606 (2015). arXiv: 1503.03606. URL: http://arxiv.org/abs/1503.03606.

[25] Xuesong Niu, Hu Han, et al. "VIPL-HR: A Multi-modal Database for Pulse Estimation from Less-constrained Face Video". en. In: *arXiv:1810.04927 [cs]* (Nov. 2018). arXiv: 1810.04927. URL: http://arxiv.org/abs/1810.04927.

[26] Xuesong Niu, Shiguang Shan, et al. “RhythmNet: End-to-end Heart Rate Estimation from Face via Spatial-temporal Representation”. en. In: *IEEE Transactions on Image Processing* 29 (2020). arXiv: 1910.11515, pp. 2409–2423. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2019.2947204. URL: http://arxiv.org/abs/1910.11515.

[27] Adrian Rosebrock. *OpenCV Haar Cascades*. 2021. URL: https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/.

[28] Rui Zou et al. “Evaluation of the accuracy of ECG captured by CardioChip through comparison of ECG recording to a standard 12-lead ECG recording device”. In: *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*. 2016, pp. 1094–1098. DOI: 10.1109/ISIE.2016.7745046.

[29] Christos Sagonas et al. “300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge”. In: (2013), pp. 397–403. DOI: 10.1109/ICCVW.2013.59.

[30] Rencheng Song et al. “PulseGAN: Learning to generate realistic pulse waveforms in remote photoplethysmography”. en. In: *arXiv:2006.02699 [eess]* (June 2020). arXiv: 2006.02699. URL: http://arxiv.org/abs/2006.02699.

[31] Jeremy Speth et al. “Remote Pulse Estimation in the Presence of Face Masks”. en. In: *arXiv:2101.04096 [cs, eess]* (Jan. 2021). arXiv: 2101.04096. URL: http://arxiv.org/abs/2101.04096.

[32] Radim Špetlík. “Visual Heart Rate Estimation with Convolutional Neural Network”. en. In: *Proceedings of the BMVC* (2018), p. 12. URL: https://cmp.felk.cvut.cz/~spetlrad/ecg-fitness/visual-heart-rate.pdf.

[33] Wim Verkruysse, Lars O Svaasand, and J Stuart Nelson. “Remote plethysmographic imaging using ambient light”. In: *Optics Express* 16.26 (Dec. 22, 2008), p. 21434. ISSN: 1094-4087. DOI: 10.1364/OE.16.021434. URL: https://www.osapublishing.org/abstract.cfm?URI=oe-16-26-21434.

[34] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* 1 (2001), pp. I–I. DOI: 10.1109/CVPR.2001.990517.

[35] Wenjin Wang, Sander Stuijk, and Gerard de Haan. “A Novel Algorithm for Remote Photoplethysmography: Spatial Subspace Rotation”. In: *IEEE Transactions on Biomedical Engineering* 63.9 (2016), pp. 1974–1984. DOI: 10.1109/TBME.2015.2508602.

[36] Zitong Yu, Xiaobai Li, et al. “AutoHR: A Strong End-to-end Baseline for Remote Heart Rate Measurement with Neural Searching”. en. In: *IEEE Signal Processing Letters* 27 (2020). arXiv: 2004.12292, pp. 1245–1249. ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2020.3007086. URL: http://arxiv.org/abs/2004.12292.

[37] Zitong Yu, Wei Peng, et al. “Remote Heart Rate Measurement from Highly Compressed Facial Videos: an End-to-end Deep Learning Solution with Video Enhancement”. en. In: *arXiv:1907.11921 [cs, eess]* (July 2019). arXiv: 1907.11921. URL: http://arxiv.org/abs/1907.11921.

[38] Kaipeng Zhang et al. “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks”. In: *IEEE Signal Processing Letters* 23.10 (Oct. 2016), pp. 1499–1503. ISSN: 1558-2361. DOI: 10.1109/lsp.2016.2603342. URL: http://dx.doi.org/10.1109/LSP.2016.2603342.

[39] Panpan Zhang et al. "Multi-hierarchical Convolutional Network for Efficient Remote Photoplethysmograph Signal and Heart Rate Estimation from Face Video Clips". In: *CoRR* abs/2104.02260 (2021). arXiv: 2104.02260. URL: https://arxiv.org/abs/2104.02260.