Investigating the Effects of Image Enhancement Techniques on the Performance of rPPG Models for Physiological Signals Extraction from Facial Videos

Intro to Image Processing (001-2902-31)

Course Project

Alexander Yevsikov

Netanel Madmoni

©

03/09/2025

1 Introduction

1.1 Photoplethysmography (PPG) and Remote Photoplethysmography (rPPG)

Photoplethysmography (PPG) is a common method for displaying blood volume changes over time. First described in the 1930s, PPG signals are widely used nowadays as they provide helpful insights into one's cardiac, respiratory, and autonomic systems. Traditionally PPG signals are obtained by placing a device called a *photoelectric plethysmograph* right over the subject's hand (typically on one of the fingers). The device consists of a light source and a light detector; It shines a light through the skin and then measures its light absorption over time [1]. The signal is cyclic since each heartbeat makes the blood vessels under the skin expand and contract, which affects the amount of light absorbed. Therefore, this signal allows the measurement of vital signs such as heart rate, heart rate variability, and oxygen saturation [2]. For an example of a PPG signal and its components, see Figure 1.

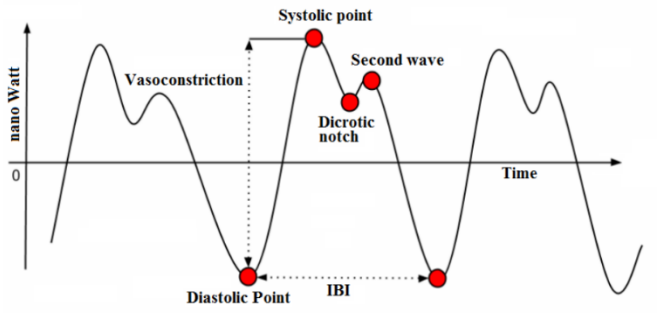

Figure 1. *A Typical PPG signal. Source: [3].*

Remote Photoplethysmography (rPPG) is a camera-driven method of obtaining a PPG signal remotely. It relies on the subtle changes in the color of the subject's skin over time to indicate the timing of cardiovascular events [4]. A typical workflow of rPPG methods is as follows: once a Region Of Interest (ROI) is selected in a video that includes the subject's skin (usually the face), each color channel of each frame is spatially averaged to obtain a pixel intensity time series. This signal is cleaned and fed into a model to extract a clean PPG signal [2]. In the last decades a variety of models, both supervised and unsupervised, were developed for extracting a quality signal and processing it to get reliable results. Some of the notable unsupervised models include GREEN [5], ICA [4], and CHROM [6]. In this work, we focus on the ICA model for its simplicity and explainability.

1.2 Independent Components Analysis (ICA) for rPPG Signal Extraction

The ICA model relies on blind source separation (BSS) techniques to identify and extract the underlying PPG signal from the noisy skin-color time series. Let $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$ be the observed signal for each of the three color channels (red, green, and blue). The model assumes the observed is a linear mixture of three sources with some mixing coefficients:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t),$$

where $\mathbf{s}(t) = [s_1(t), s_2(t), s_3(t)]^T$ are the underlying sources and $\mathbf{A}^{3\times3}$ is the mixing matrix. The goal of the method is to find the unmixing matrix $\mathbf{W}$ such that:

$$\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t),$$

where $\hat{\mathbf{s}}(t)$ is the estimated reconstruction of the original sources [4]. Empirically it was found that $\hat{s}_2$ is the most representative of the PPG signal.

1.3 Project Overview

In this work, we aim to investigate the effects of several image processing techniques on the performance of the ICA model for rPPG signal extraction. By applying several image enhancement algorithms on each of the video frames with different parameters we attempt to improve the resulting PPG signal. We use Bayesian optimization techniques to find the parameters that yield the best results. Our research question is: can image enhancement techniques improve the performance of an rPPG signal extraction model?

2 Data Description

For our experiments we use the UBFC-RPPG dataset [7], a common dataset for testing and benchmarking rPPG algorithms[1].

The dataset consists of video recordings of 42 subjects' faces, along with a ground truth PPG signal. The videos were recorded using a low-cost RGB webcam at 30 FPS with a resolution of 640x480 saved in an 8-bit format. Subjects were seated about one meter away from the camera which pointed at his or her face. The PPG signals were captured using a CMS50E transmissive pulse oximeter.

A sample of the dataset is shown in Figure 2.

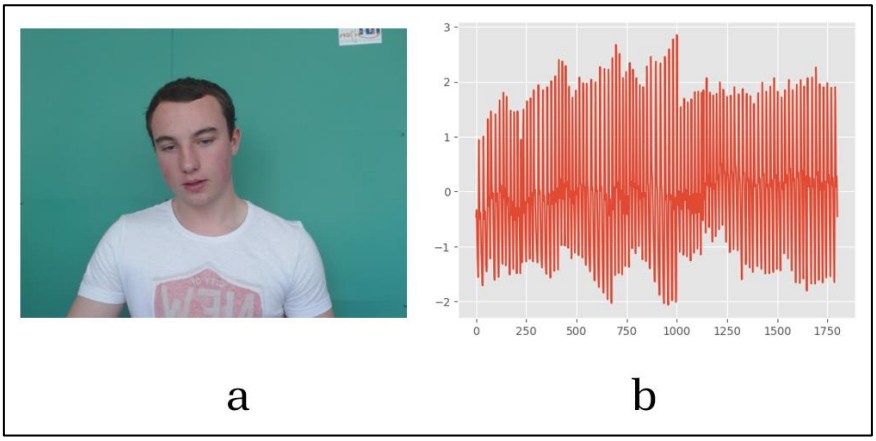

Figure 2. *A sample of the UBFC-RPPG dataset. a) a frame from one of the video recordings, b) the ground-truth PPG signal for that recording*

[1] Homepage: https://sites.google.com/view/ybenezeth/ubfcrppg. The dataset is available upon request from the authors.

3 Method

Our processing pipeline is described below:

1. **Video preprocessing.** The videos were first cropped to remove many of the background pixels and make processing faster. Facial landmarks were then detected every 10th frame, using `dlib`'s face detector[2], and the forehead region was segmented manually by cropping the frames around the relevant landmarks' keypoints. Using a threshold-based segmentation method and known threshold values [8], the skin area of the forehead region was segmented.
2. **Image enhancement.** We experimented with several image enhancement techniques; namely histogram equalization, histogram matching, color space conversion, high high-pass and low-pass filtering. Histogram equalization was used to enhance the contrast of an image by redistributing the intensity values of the pixels. It works by spreading out the intensity values across the entire histogram, making the distribution more uniform. Histogram matching was used to adjust the intensity distribution of an image to match a specified histogram. It works by mapping the intensity values of the input image in HSV format to match the cumulative distribution function (CDF) of a reference histogram. This process can be useful for color correction or matching images with similar lighting conditions. Color space conversion was used to evaluate the effect of changing the color encoding, specifically how it interacts with the Independent Component Analysis (ICA) model that we used for signal extraction. Low-pass filtering was used to suppress high-frequency components while preserving low-frequency components in an image. By using a convolution kernel whose values are between 0 and 1, and together sum to 1, we reduced noise and blurred the image. A high pass filter was used to allow high-frequency components in an image while suppressing low-frequency components. Subtracting the smoothed version of the image (low-frequency components) from the original image using a specialized kernel that sums to 0, resulted in enhanced edges and fine details. By doing this we hoped to emphasize important features in the forehead, such as veins.
3. **Signal extraction.** Using the implementation of the ICA model for rPPG signal extraction [4], an estimated PPG signal was extracted from each video. Independent Component Analysis (ICA) is used to separate a multivariate signal into additive, statistically independent components. ICA can be employed to isolate the PPG signal from a video signal of a person's face. ICA operates under the assumption that the observed signal (video) is a linear combination of independent source signals. It aims to find a transformation matrix that linearly separates the mixed signals into statistically independent components, each representing a distinct source. Typically, one of these independent sources will be a PPG signal.

[2] See http://dlib.net/face_landmark_detection.py.html.

4. **Heart rate (HR) calculation.** For both the estimated and ground truth signal, an estimated heart rate was calculated by applying a fast Fourier transform on the signal using a 10-second sliding window and taking the maximal frequency. This step resulted in two series (estimated and ground truth) of six heart rates each.
5. **Parameter search.** Our goal was to find the best combination of image enhancement techniques (which we considered hyperparameters) that can improve the PPG signal that we extract from the videos. For this purpose, we employed the Optuna hyperparameter optimization library in Python. Optuna uses Bayesian optimization to intelligently search the hyperparameter space. This process balances exploration and exploitation to converge to the optimal hyperparameters with a minimal number of iterations.

   To provide a reliable estimate of our model's performance we chose to use a validation procedure which involves a train test split with cross validation for hyperparameter tuning on the train set. Our dataset contained a total of 42 subjects, we randomly set aside 7 subjects to the test set. The subjects that were chosen are number 25, 8, 23, 24, 16, 12 and 38 in the dataset. The 35 videos of the rest of the subjects were used for training. We chose to use 5-fold cross validation which implies that in each training fold, we used 28 videos for training and 7 videos for validation.

   To not overfit the data, the search for the best hyperparameters was done independently for each fold. First, we used Optuna to suggest the best combination of hyperparameters. The hyperparameters were selected from predefined sets. The first set contained 6 color space options: RGB, Lab, Luv, HSV, HSL, and YUV. The second set contained 4 low-pass filter options: box filter, median filter, gaussian filter, or no low-pass filter. The third set contained 3 high-pass filter options: Laplacian filter, a tunable custom filter we designed, or no high-pass filter. The custom filter draws inspiration from the Sobel and Prewitt filters. Our custom filter sums to 0 (or 1 when *sharpen* is set to True), has either vertical or horizontal symmetry, and can have a range of different numbers. The hyperparameters we tuned for our filter are *base*, *scale*, *midline increment*, *flip* and *sharpen*. The *base* parameter is the number that will be at the four edges of the kernel; The *scale* parameter is a liner multiplier that controls the increase or decrease in the kernel values as we move closer to the central row or column of the kernel; The *midline increment* parameter is the addition to the values in the middle row or column of the kernel (in a similar fashion to Sobel where the increment could be thought of as 1, since the middle value is higher than the value at the adjacent edges by 1); The *flip* parameter dictates whether the kernel will have horizontal or vertical symmetry, where the default is horizontal and the *flip* parameter set to True then the symmetry is vertical. When the *sharpen* parameter is set to False, the middle value of the kernel is 0, when it is set to True, the middle value is set to 1.

   Because the chosen image enhancement techniques we use can fundamentally change the PPG signal we can extract, we chose to also tune the hyperparameters of the ICA model. The original

implementation by `rPPG-Toolbox` [9] used an ICA with a low pass filter of 1, a high pass filter of 5, and a lambda smoothing factor of 100. We decided to look for the best option for a low pass filter from the range of real numbers between 0 and 5. For the high pass filter we looked at the range between 0 and 10, and for lambda, we looked at real values between 50 and 200.

6. **Evaluation**. After a hyperparameter combination was selected from the options above, we applied the selected transformations on the training videos and extracted the PPG signal using ICA. The signal was then compared to the true signal and evaluated using the mean absolute percentage error (MAPE) as a metric, where:

$$\text{MAPE} = \frac{1}{6}\sum_{i=1}^{6}\left|\frac{\text{ground truth HR} - \text{predicted HR}}{\text{ground truth HR}}\right|$$

   The MAPE score on the training set was passed to the acquisition function of Optuna so better parameters can be suggested in future iterations. We then used the same procedure on the validation set and saved the result for later study. However, this result wasn't passed to Optuna to avoid overfitting.

7. **Preliminary results.** Initially, we ran Optuna for 100 iterations for each of the 5 folds. This was done for an evaluation of our technique. The runs were performed before the final project presentations in class. By the end of the process, we had 500 different configurations and the performance they achieved on the whole training and validation set. We were able to conclude that it is possible to reduce the MAPE to around 5% using the right configuration.

   We also tried not using any image enhancement techniques and not changing the ICA model parameters to have a baseline for comparison. This method which included only the segmentation of the forehead, followed by the default ICA signal extraction, achieved 16.88% MAPE.

8. **Final parameter search space**. Based on our preliminary findings, we concluded we need to increase the number of iterations for each fold. We decided to increase the number from 100 to 250. We also concluded that the HSV and HSL color spaces were not suitable for this learning task, so they were removed from the search space. The Laplace high pass filter was also removed for poor initial results. Also, based on input we received during our presentation, we decided to add histogram-based image enhancement techniques. The final search space we used is as follows. First, we apply either histogram equalization, histogram matching, or make no changes. Second, we apply a color transform, either the default RGB, or one of YUV, Lab, or LUV. Third, we apply a low pass filter of customizable size, either a box, a median, or a Gaussian filter, or we don't apply a filter. Fourth, we apply a high pass filter, either the customizable Sobel-like kernel we built or no high pass filter.

   Note that we had to select an image whose histogram we will use as a reference to perform histogram matching. For this image, we chose the first frame of subject 49. We experimented with either using the image of only the forehead, by manually cropping the relevant region or using the full image. The

full image contained the forehead surrounded by a black background. We speculated this could adversely affect the histogram matching method.

4 Results

4.1 Distribution Analysis

MAPE distributions over key parameters are shown in Figure 3.

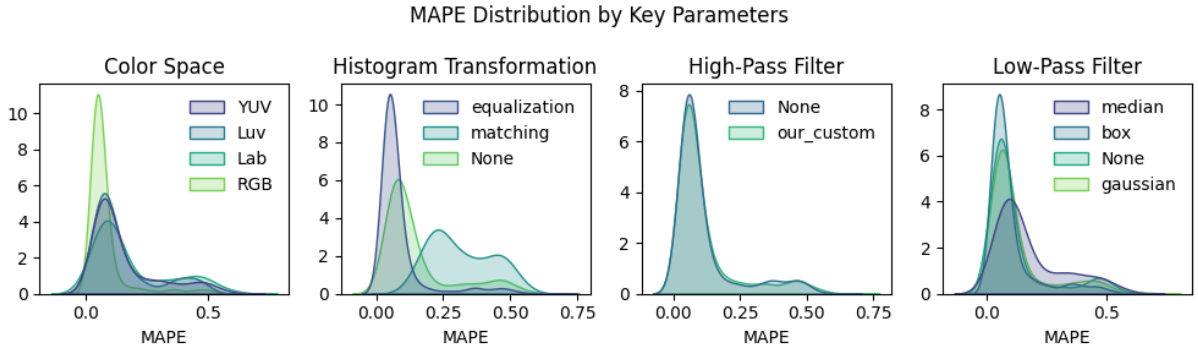

***Figure 3**. MAPE distribution over key parameters.*

From the figure above we learn that some of the configurations – for example, RGB color space and histogram equalization – achieved narrower MAPE distributions near around 0-0.1, highlighting the effectiveness of these parameters in lowering the error. We can observe that the distribution of our custom filter is almost identical to the distribution of applying no high pass filter at all, therefore it is likely unnecessary. Below is the MAPE distribution for different kernel sizes of each of the filters tested.

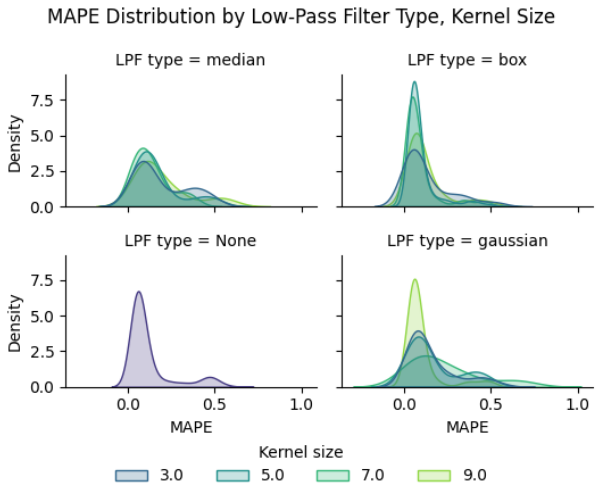

***Figure 4.** MAPE distribution by LPF filter type and kernel size.*

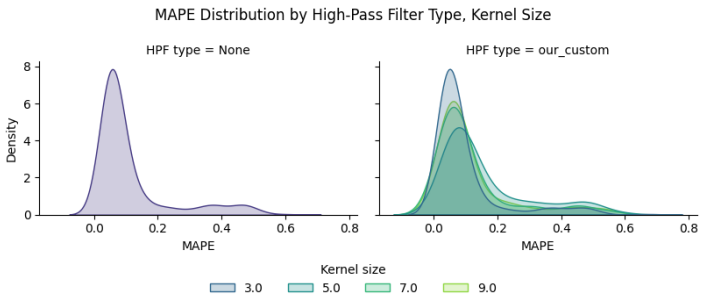

***Figure 5.** MAPE distribution by HPF filter type and kernel size*

For the low-pass filter, box and Gaussian blur yielded the narrowest distribution around low MAPE value. As for kernel sizes, each filter had a different best kernel size: 5x5 for box blur, 7x7 for median blur, and 9x9 for Gaussian blur. The high-pass filter benefited the most from a kernel size of 3x3.

4.2 Best Parameter Configurations

Following are the best parameter configurations for each fold and their MAPE on the training set:

fold	*1*	*2*	*3*	*4*	*5*
MAPE	**3.3%**	**2.6%**	**3.9%**	**3.4%**	**3.2%**
Color space	RGB	RGB	RGB	RGB	RGB
Low-pass filter parameters					
Low-pass type	-	gaussian	gaussian	-	box
Low-pass kernel size	-	9	9	-	5
Low-pass filter repeat amount	-	5	1	-	1
Histogram transformation parameters					
Histogram transformation	equalization	equalization	equalization	equalization	equalization
Crop before histogram matching	-	-	-	-	-
High-pass filter parameters					
High-pass type	-	custom	-	-	custom
High-pass kernel size	-	3	-	-	3
base	-	-5.036	-	-	0.869
flip	-	yes	-	-	no
Midline increment	-	2.388	-	-	3.875
scale	-	1.851	-	-	1.662
sharpen	-	yes	-	-	no
ICA model parameters					
ICA high-pass	3.888	4.284	1.725	3.919	4.884
ICA low-pass	1.035	1.286	1.029	1.046	1.003
ICA lambda	140.473	117.06	93.501	97.96	178.966

RGB color space and histogram equalization yielded the best results on all folds, while low-pass filters are present in 3/5 of the best configurations, and a custom high-pass filter is present in 2/5 of them. These findings match the parameter distributions presented in Figures 3–5.

Different filter configurations were selected for different folds, highlighting the importance of the data subset that is tested. To improve the robustness of our analysis, we tested each of the best configurations on the entire dataset. Out of the best five configurations, configuration number 1 is the one that

performed best on the entire training and validation dataset, with MAPE of 3.07%. That same configuration yielded MAPE of 2.7% on the 7 subjects in the test set.

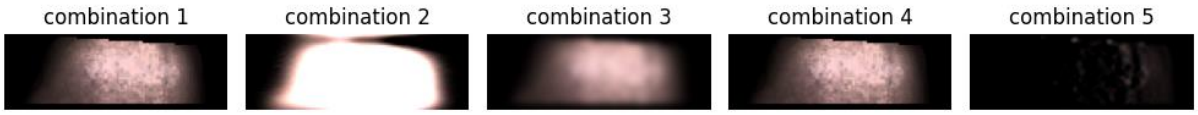

***Figure 6**. Visualizations of the best parameter configurations on a static forehead image.*

Discussion

In this project we set out to improve the performance of the ICA method for rPPG signal extraction. In their recent paper [9] reported achieving 14.34% MAPE on the UBFC dataset. We successfully reduced this number to around 3% MAPE using the right combination of image enhancement techniques and ICA model parameters. We also significantly improved the score compared to the baseline of 16.88% that we started with.

We had high hopes for the histogram matching method, since it had the potential to address different skin colors, but its performance was lacking. One way this method could be adjusted to fit our data is by selecting a better frame to use as a reference. Our simplistic method of choosing a subject at random and choosing the first frame might have contributed to the poor results. There are several methods that could improve the results. One is choosing a reference frame. Second, perhaps two layers of matching are required, one layer for each subject and then another one across all subjects. Another option is to apply the matching only to the section of the image that contains the forehead, by selecting it using a threshold method.

Histogram equalization was the most successful image enhancement technique we tested. The type of equalization we used was very simple and couldn't be adjusted. We believe that adding a tunable parameter that controls the amount of equalization can further improve the results.

Conclusions and Future Work

For future work, it might be of interest to test more image enhancement techniques to improve model performance. The robustness of the model could also be improved, for example, by analysing the distribution of skin tones in the dataset and selecting a representative image for histogram matching.

Another improvement we suggest for improved robustness is using several different preprocessing combinations pipelines and averaging the predictions in an ensemble.

Further improvements could also be made to broaden the scope of the project. For example:

- Add control to the order of image enhancements and search for the optimal order.
- Broaden the search space for each parameter.
- Test results on other datasets, such as PURE [10] and MMPD [11].
- Use other signal extraction models, both unsupervised and supervised.

Bibliography

[1] A. A. Alian and K. H. Shelley, "Photoplethysmography," *Best Practice & Research Clinical Anaesthesiology*, vol. 28, no. 4, pp. 395–406, Dec. 2014, doi: 10.1016/j.bpa.2014.08.006.

[2] P. Pirzada, A. Wilde, G. H. Doherty, and D. Harris-Birtill, "Remote Photoplethysmography (rPPG): A State-of-the-Art Review." medRxiv, p. 2023.10.12.23296882, Oct. 12, 2023. doi: 10.1101/2023.10.12.23296882.

[3] R. K. Nath, H. Thapliyal, and A. Caban-Holt, "Towards Photoplethysmogram Based Non-Invasive Blood Pressure Classification," in *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, Hyderabad, India: IEEE, Dec. 2018, pp. 37–39. doi: 10.1109/iSES.2018.00018.

[4] M.-Z. Poh, D. J. McDuff, and R. W. Picard, "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation," *Opt. Express*, vol. 18, no. 10, p. 10762, May 2010, doi: 10.1364/OE.18.010762.

[5] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, "Remote plethysmographic imaging using ambient light," *Opt. Express*, vol. 16, no. 26, p. 21434, Dec. 2008, doi: 10.1364/OE.16.021434.

[6] G. de Haan and V. Jeanne, "Robust Pulse Rate From Chrominance-Based rPPG," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 10, pp. 2878–2886, Oct. 2013, doi: 10.1109/TBME.2013.2266196.

[7] S. Bobbia, R. Macwan, Y. Benezeth, A. Mansouri, and J. Dubois, "Unsupervised skin tissue segmentation for remote photoplethysmography," *Pattern Recognition Letters*, vol. 124, pp. 82–90, Jun. 2019, doi: 10.1016/j.patrec.2017.10.017.

[8] J. Kovac, P. Peer, and F. Solina, "Human skin color clustering for face detection," in *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, Ljubljana, Slovenia: IEEE, 2003, pp. 144–148. doi: 10.1109/EURCON.2003.1248169.

[9] X. Liu *et al.*, "rPPG-Toolbox: Deep Remote PPG Toolbox." arXiv, Nov. 24, 2023. Accessed: Mar. 20, 2024. [Online]. Available: http://arxiv.org/abs/2210.00716

[10] R. Stricker, S. Muller, and H.-M. Gross, "Non-contact video-based pulse rate measurement on a mobile service robot," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, UK: IEEE, Aug. 2014, pp. 1056–1062. doi: 10.1109/ROMAN.2014.6926392.

[11] J. Tang *et al.*, "MMPD: Multi-Domain Mobile Video Physiology Dataset," 2023, doi: 10.48550/ARXIV.2302.03840.

Supplementary Material

The code for this project can be found on the GitHub repository: https://github.com/MNetanel/image-processing-project.