IoT based Non-contact Heart rate Monitor system in Raspberry Pi using Webcam

Ganesh Kumar T K
Smart Manufacturing Undergraduate , Department of Mechanical,
Indian Institute of Information Technology , Design and Manufacturing, Kancheepuram,
Chennai, India
Email: msm17b034@iiitdm.ac.in

***Abstract*—In the times of CoVID-19, a non-contact means of measuring heart rate could be beneficial for sensitive populations, and the ability to calculate heart rate using a simple webcam or pi-camera could be useful in tele-medicine. This novel method is termed remote PhotoPlethysmoGraphy (rPPG). This project discusses the re-implementation of one such approach that uses independent component analysis on mean pixel color values within a region of interest (ROI) about the face and simultaneously showing the Heart rate values in ThinkSpeak Dashboard. I am also exploring the idea further by assessing the algorithm's robustness to subject movement and bounding box noise and offering all these in a bundled Python package that can simply installed and run-on-go.**

***Index Terms*—Heartrate, rPPG, remote-health, telemedicine, raspberry-pi, python, opencv, covid**

I. INTRODUCTION

Home health care is nowadays growing and changing discipline. As a blessing in disguise, CoVID-19 situation has helped Remote healthcare services to explore more for greater extent The remote monitoring of vital signs includes not only the high accuracy diagnostic devices but also simple ones and accessible for everyone. One of the most frequent examinations performed in health care monitoring is Heart rate measurement.

There are many different methods of contact measurement of a heart rate among which the golden standard is an ElectroCardioGraphy (ECG). However, recording electric potential generated by the heart requires appropriate application of the electrodes what may be too complicated and annoying in home conditions.

II. PHOTOPLETHYSMOGRAPHY (PPG)

Photoplethysmography (PPG) is a low-cost and non-invasive method of measuring the cardiovascular blood volume pulse (BVP) via light transmitted through, or reflected from, the human body [1]. There are many clinical applications for PPG and it can reveal significant information about health and risk of cardiovascular diseases [2] . The peripheral pulse wave, as detected via PPG, characteristically exhibits systolic and diastolic peaks. The systolic peak is a result of the direct pressure wave traveling from the left ventricle to the periphery of the body, the diastolic peak (or inflection) is a result of reflections of the pressure wave by arteries of the lower body [3]. Figure 1 shows an example of a pulse waveform (black line) with the systolic and diastolic peaks labeled.

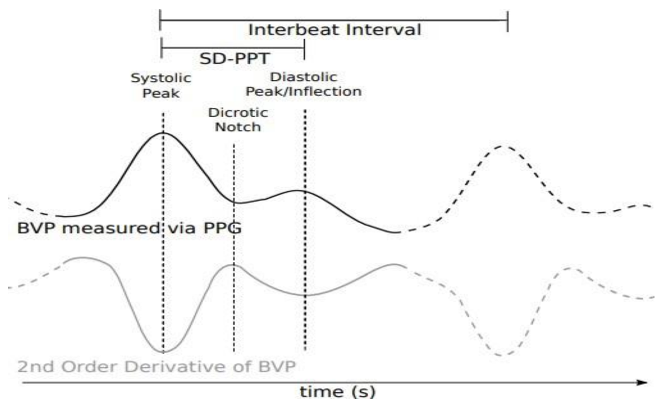

Fig. 1. Systolic-diastolic peak-to-peak time (SD-PPT) is calculated as the time between the systolic and diastolic peaks within the PPG signal.

III. METHODOLOGY

I implemented a method that takes an input video of a person's head and returns a pulse rate as well as a series of beat locations which can be used for the analysis of beat-to-beat variability and the end result is given through Flask based API. First, I extracted the motion of the head using feature tracking and then isolated the motion corresponding to the pulse and projected it onto a 1D signal that allows extracting individual beat boundaries from the peaks of the trajectory. For this, I used PCA and select the component whose temporal power spectrum best matches a pulse. The, I projected the trajectories of feature points onto this component by Matplotlib and extracted the beat locations as local extrema. Fig. 2 presents an overview of the technique.

For this I just assumed the recorded subject is stationary and sitting upright for the duration of the video and start by locating the head region and modeling head motion using trajectories of tracked feature points. The trajectories have extraneous motions at frequencies outside the range of possible pulse rates, and so I temporally filter them and then used PCA to decompose the trajectories into a set of independent source signals that describe the main elements of the head motion. To choose the correct source for analysis and computation of the duration of individual beats, I simply examine the frequency spectra and select the source with the clearest main frequency. Average pulse rate is identified using this frequency. For more fine-grained analysis and calculation of beat duration, I had to perform peak detection in the time-domain.

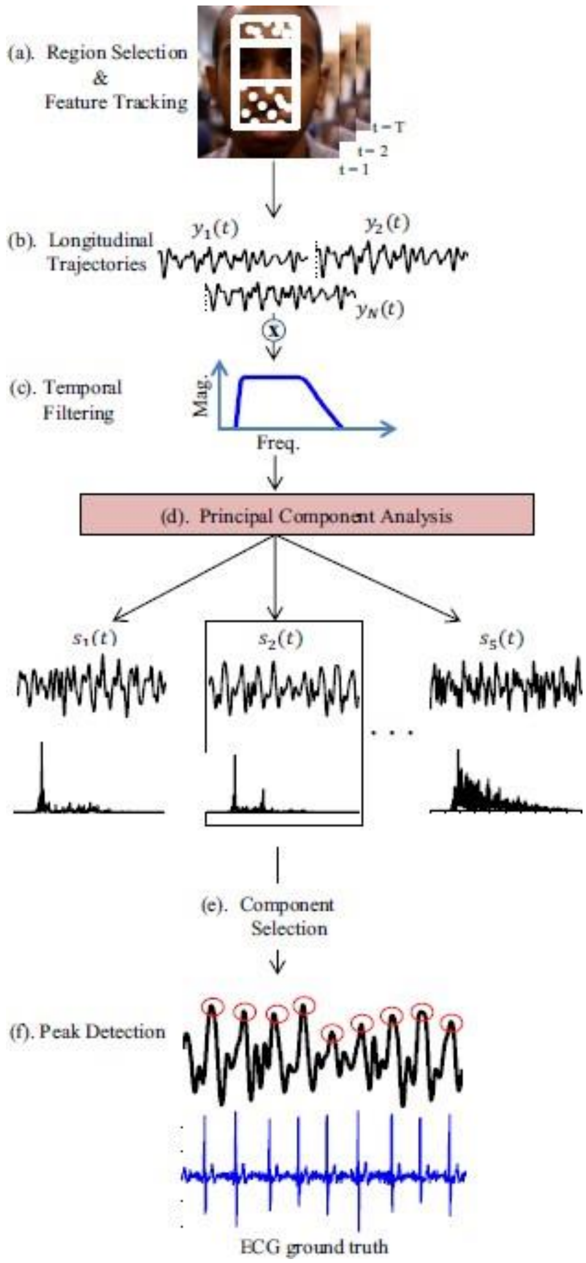

Fig. 2. Overview of our pulse estimation approach. (a) A region is selected within the head and feature points are tracked for all frames of the video. (b) The vertical componentis extracted from each feature point trajectory. (c) Each trajectory is then temporally filtered to remove extraneous frequencies. (d) PCA decomposes the trajectories into a set of source signals s1, s2, s3, s4, s5. (e) The component which has the clearest main frequency is selected. (f) Peak detection identifies the beats of the signal.

IV. Project Setup

A. Hardware Requirements

1) Raspberry Pi (acting as server).
2) Pi Camera.
3) 2.5 A Power supply for Pi.
4) A Laptop with Internet connection (acts as remote client).

B. Software Requirements

1) Python3.7 +
2) OpenCV
3) opencv-contrib-python
4) Scikit-learn library
5) harr-casscade-frontface.xml (Face detection algorithm).
6) nginx server (if you want it to access across internet)
7) ThinkSpeak Account.

C. Method followed

The person was placed in front of the Pi-camera and the face bounding box was applied to the face on the real time video. The highest peak within the power spectrum of the signal was taken to be the reference heart rate for each 30-second window.

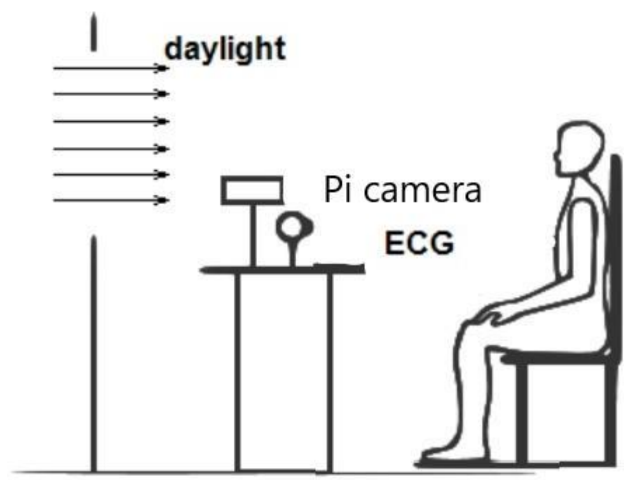

Fig. 3. Project setup for obtaining Heart Rate

To simulate this, we can add artificial noise to the bounding box corner locations found with the Haar cascade classifier. In each frame, corners of the bounding box were shifted horizontally and vertically by percentages of the width and height, where the percentages were chosen uniformly at random up to a maximum noise percent.

Then the GUI (optionally included for reference) is opened up with Heart rate plot and highlighted Average beats per minute (bpm) value. The same value is stored as csv file and also exported to ThinkSpeak dashboard from where the data is sent to the remote client.

ThinkSpeak Dashboard [4] also can be accessed by unique link generated by the client and Heart rate graph along with the latest value is displayed.

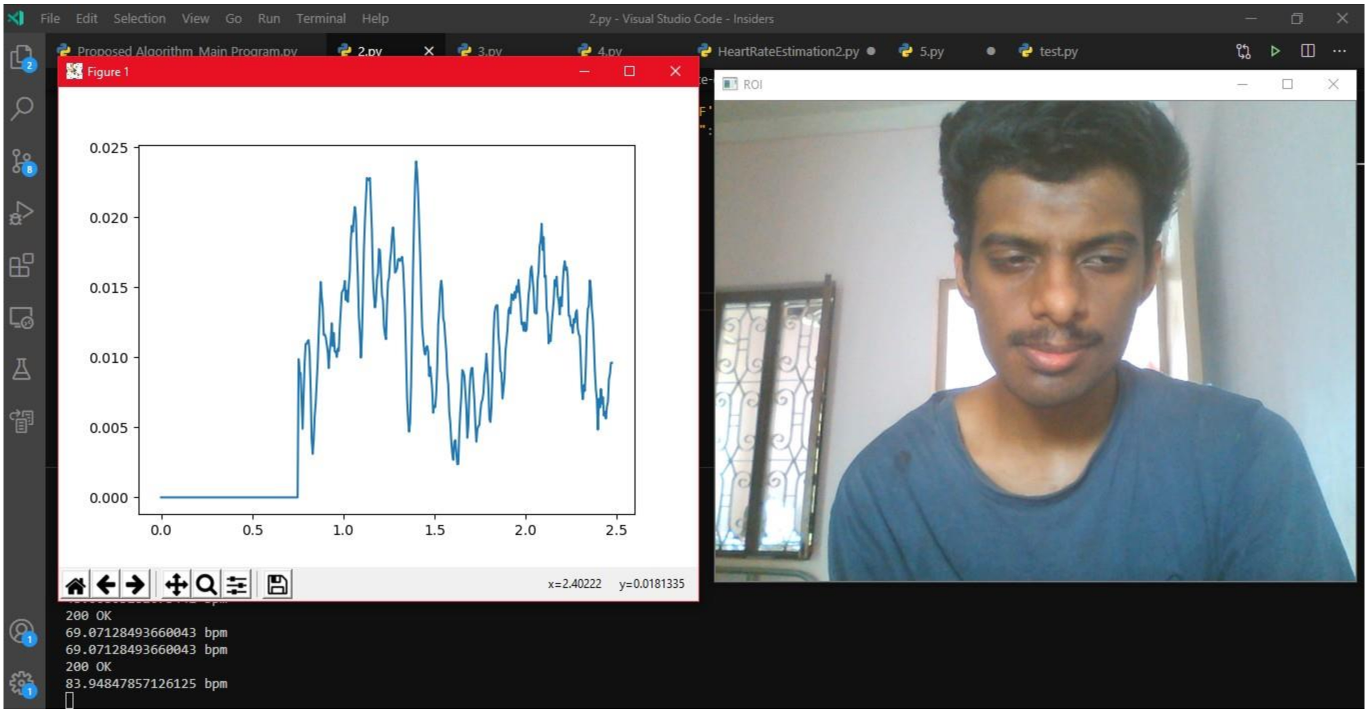

Fig. 4. Detecting the heartrate by locking on to the forehead movement

V. Results

Apart from power fluctuations in Raspberry pi and lower fps in pi-camera, it was running as expected. Once the facial bounding box(es) found using the Haar-cascade classifiers were narrowed down to a single bounding box on the subject's face.

To test the accuracy, I cross-checked the values of bpm with the heartrate sensor from Samsung Gear Fit 2 watch. The results are given below in Table I.

TABLE I

Comparison of Heart rates taken from Pi camera using above methodology and Samsung Gear Fit Watch.

Samples	Pi-camera measurements (Avg. Heart rate in bpm)			Samsung Gear Fit values
	zero-crossing		fft	in bpm
	face	forehead	face/forehead	wrist
1.	87.56	100.57	91.41	87.89
2.	59.32	59.18	58.01	58.59
3.	94.84	104.34	98.44	99.61

The Heart rate data sent from Python console also received in dashboard and it plots bpm vs. time chart properly

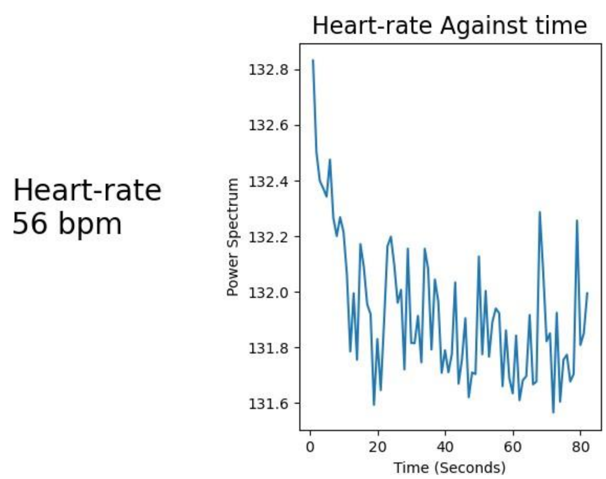

Fig. 5. Heart rate using Power Spectrum analysis

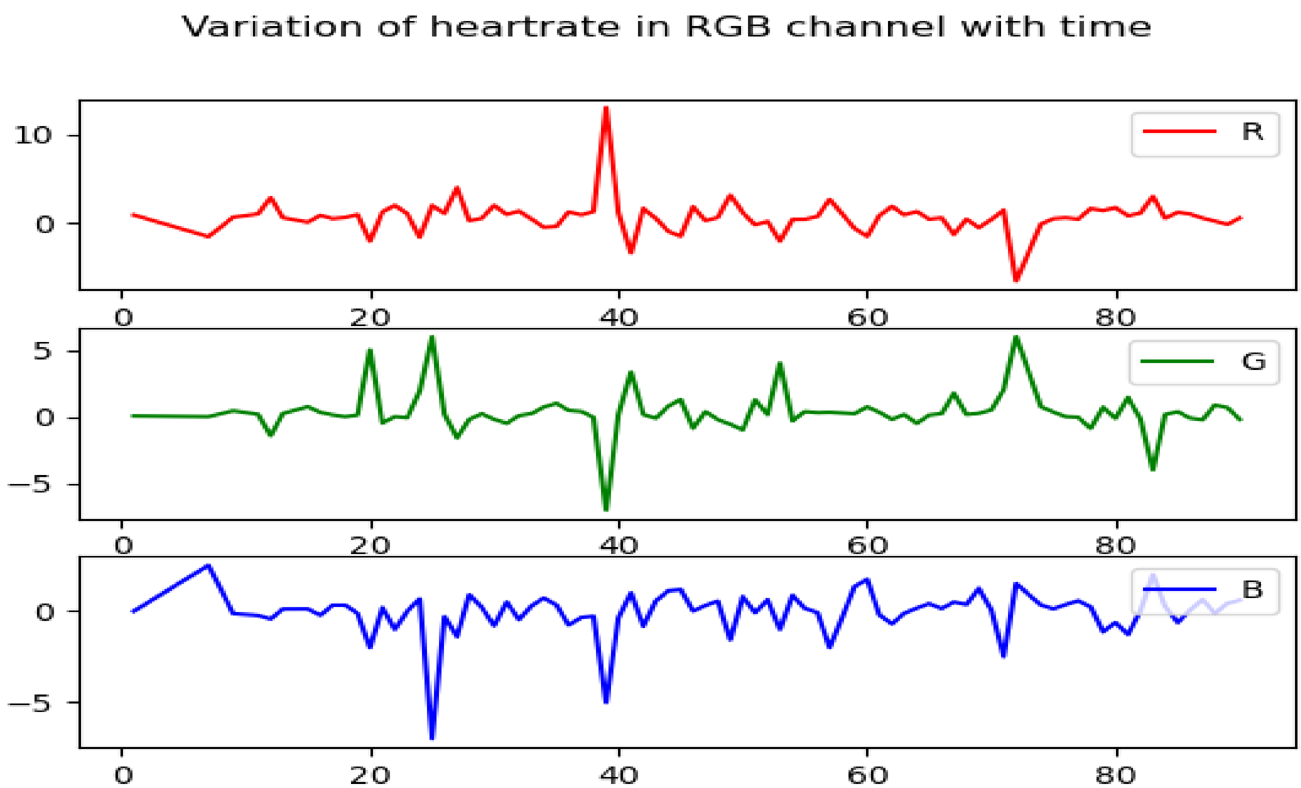

Fig. 6. Heart rate variation in RGB Channels before applying fft and FastICA

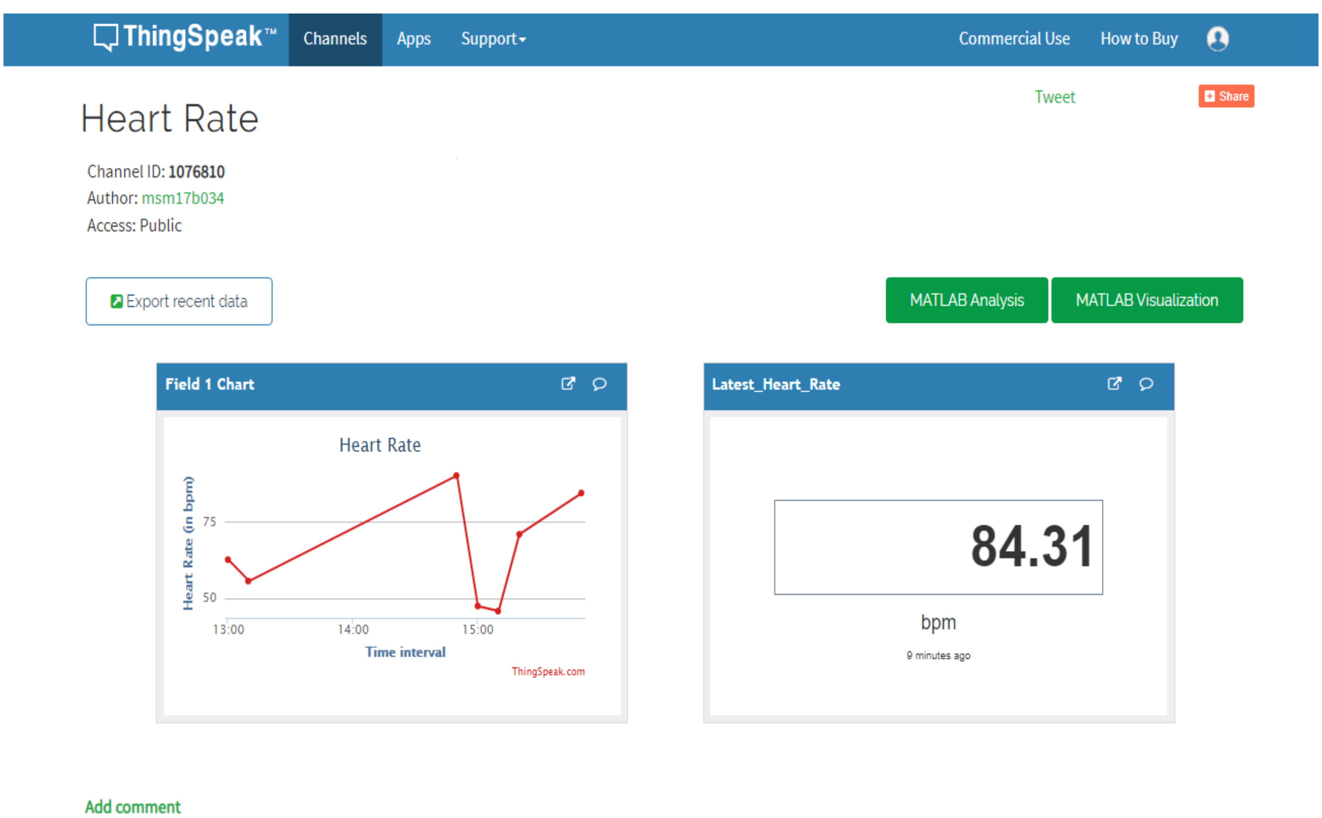

Fig. 7. ThinkSpeak Dashboard Containing Heart rate values sent over IoT

VI. Conclusion

Thus, a simple processing of image data in a microcomputer i.e., Raspberry Pi and then applying PCA allows extracting the changeable component containing information of the heart rate. The presented IoT based Project seems to be quite effective and easy to use in the daily monitoring of home care patients.

References

[1] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiological measurement*, vol. 28, no. 3, pp. 1–1, 2007.

[2] J. Allen, C. P. Oates, T. A. Lees, and A. Murray, "Photoplethysmography detection of lower limb peripheral arterial occlusive disease: a comparison of pulse timing, amplitude and shape characteristics," *Physiological Measurement*, vol. 26, no. 5, pp. 811–821, 2005. [Online]. Available: 10.1088/0967-3334/26/5/018;https://dx.doi.org/10.1088/0967-3334/26/5/018

[3] A. M. Brumfield and M. E. Andrew, "Digital pulse contour analysis: investigating age-dependent indices of arterial compliance," *Physiological Measurement*, vol. 26, no. 5, pp. 599–608, 2005. [Online]. Available: 10.1088/0967-3334/26/5/003;https://dx.doi.org/10.1088/0967-3334/26/5/003

[4] Ganesh T. K, "Heart Rate - ThinkSpeak Dashboard," 2020. [Online]. Available: https://thingspeak.com/channels/1076810

[5] GitHub Repository for the project: https://github.com/coderganesh/heartpi