A Ubiquitous Solution to Measuring Pulse Rate

Ayush Anand

Third year Undergraduate, Computer Science and Engineering, Indian Institute of Technology, Gandhinagar.
ayush_anand@iitgn.ac.in

In this paper, we look at how we can use the inbuilt camera in smartphones to estimate the pulse rate of the user. We extract the different color channels from the camera readings and obtain the fluctuation in the mean of intensity values of the pixels, which corresponds to the pulse of the individual. We try to find the pulse rate using these values and compare our results with the ground truth. We take four different readings for each individual at four different times a day and see how the values vary.

Additional Keywords and Phrases: PPG, Ubiquitous Computing, Camera, Pulse rate, Smartphones

1 INTRODUCTION

Pulse rate is an important vital measure of physical activity which indicates whether the heart is working fine. In an emergency situation, the pulse rate can indicate whether the heart is pumping enough blood. Elevated pulse rate can denote heart problems, low levels of potassium in the body, anemia, asthma and many other plausible medical conditions. A lower-than-normal pulse rate, on the other hand, could be an indicator of heart attack, typhoid, underactive thyroid gland and so on. The normal pulse rate for healthy adults ranges between 60 to 100 beats per minute.

The most common method of measuring pulse rate is through a pulse oximeter, which is a clip like device. It requires the individual to place his finger between the clips, and it passes a cold light source that shines a light through the fingertip making it appear red. It then analyzes the light passing through the finger and determines the percentage of oxygen in the blood. This is also the basic concept behind PPG.

Recently, photoplethysmogram (PPG) has emerged as an efficient and accurate alternative for estimating pulse rate. PPG is a method which optically detects changes in blood flow volume and is often used in optical heart rate sensor of smartwatches.

In this paper, we will try to use the inbuilt camera of the smartphone to measure the pulse rate, making use of the same idea behind PPG. The participant is asked to place his finger on the phone covering both the camera and the flash, with the fingertip covering the camera. The finger is illuminated by the flash, and a video of roughly 20 seconds is recorded. As blood flows back and forth from the artery in the fingertip, the intensity values of the recorded video fluctuate. Prior studies have shown that it suffices to analyze these fluctuations only for the red channel, although it could be done for other channels as well. We then make use of OpenCV to separate out the red channel from the frames, and analyze the fluctuations in signal which correspond to pulse rate.

We also perform a FFT on the signal in the later step which further removes the irregularities from the signal. We then count the number of peaks/fluctuations in the signal in the given duration and then calculate the pulse rate.

2. RELATED WORK

There has been some very promising work in this field. The method of PPG was first proposed back in 1938, since when it has been an active topic of research and application. Today, PPG is extensively used in pulse oximeters to measure pulse rates at clinics and homes. [1] is the first paper which incorporates PPG with a ubiquitous solution, making it efficient and effortless to estimate pulse and heart rate anywhere and anytime.

In recent years, PPG has also evolved as a method for biometric authentication. Work in this field differs in terms of measurement device (reflection vs absorption mode), feature types, sample size and consideration of feature stability over time.

Some notable works in this field are mentioned hereafter. [2] by Bao uses heart rate variability derived from PPG as a sole feature. Recent studies have extended the feature set to more comprehensively capture the distinctiveness of each constituent part of the PPG waveform and incorporate non-fiducial methods to simplify the process of peak detection in noisy signals [3]. Kavsaolu's work [4] was the first one which analyzed data from three different recording sessions with 30 participants, unlike the ones before which used data from a single measurement.

3. DATA COLLECTION

We performed the experiment to collect data on 6 participants, which consisted of both male and female candidates. The procedure was as follows: The participant holds the phone in his/her right hand, and places his/her index finger over the camera and the flash in such a way that the fingertip corresponds with the camera. The flash is switched on and a video recording is started. The video is stopped after around 20 seconds, and the file is exported. The participant is instructed to do a light/gentle touch on the camera and not put too much pressure. The same process is repeated for four different times of the day and the dataset is constructed.

Figure 1: Participant holding the phone with index finger over camera and the flash with the video on

4. DATA ANALYSIS

4.1 Explanation:

In this section we explain the different phases of data analysis and how we use the data to estimate the pulse rate of the participant.

The video file is first read into python using OpenCV. Using standard OpenCV functionality, we find out the FPS (frames per second), duration and frame count for the video. We cap all of these values to the greatest integer equal to or less than the value (floor function). It is then read frame by frame, and each frame is split into red, blue and green channels. We find the mean of the pixel values of the red channel for each frame and append this to an array called signal. We limit our function to read only duration*FPS number of frames. We then set a hyperparameter samples_to_skip, which indicates how many initial frames we should skip in our analysis, to avoid auto exposure. We usually keep this to be all the samples covered in the first second. We then normalize our signal array by subtracting the mean values and dividing it by standard deviation. We now plot the normalized values against frames, and get a graph as shown below:

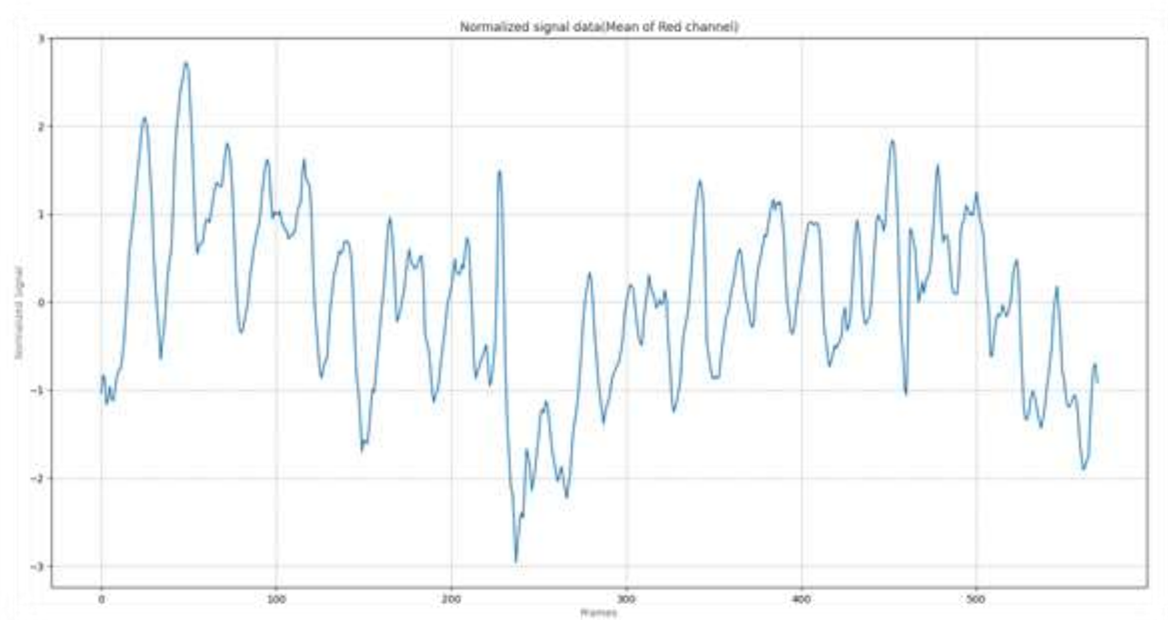

Figure 2: Normalized signal values vs frames

We can see the above signal values are irregular, and thus will not give very accurate readings. To simplify the signal further, we perform a FFT on the signal to note which frequencies contribute the most to the above plot, and obtain the following results:

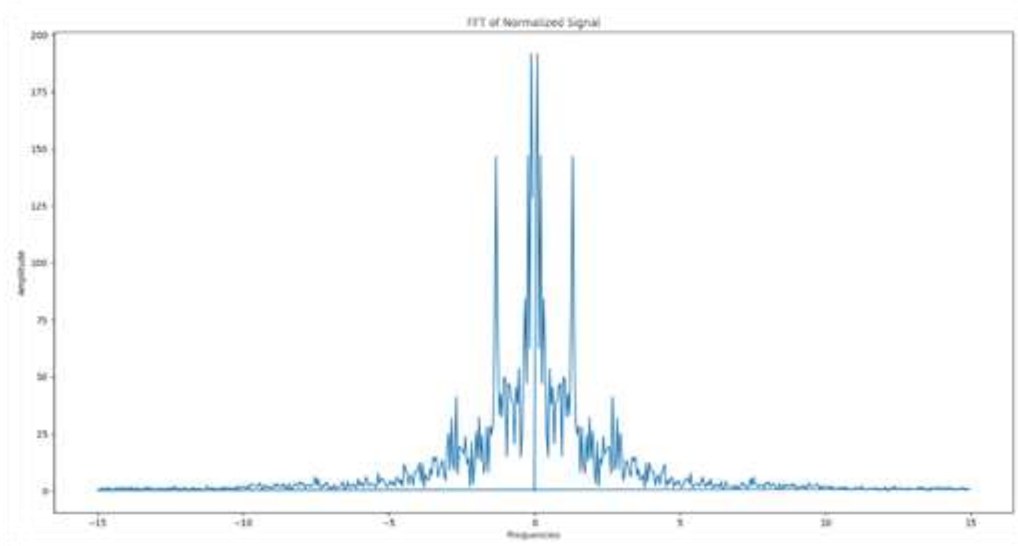

Figure 3: FFT of the normalized signal values

We can see that most of the contribution is by the frequencies below 2.5, and the ones above it are mostly noise. Thus, we create a low pass filter with threshold 2.5 Hz and pass the signal through it to remove the irregularities further and obtain a smoother signal:

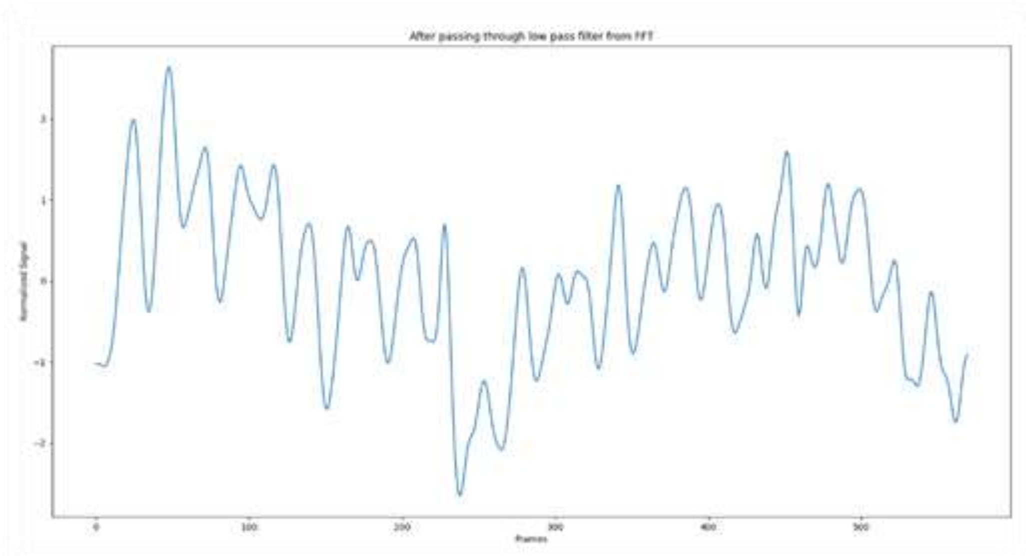

Figure 4: Signal values after passing them through the low pass filter

Finally, we make use of the *findpeaks* function from scipy and mark those peaks which correspond to a value greater than 0. These number of peaks divided by (duration of video-1) multiplied by 60 gives us the number of beats per minute (pulse rate):

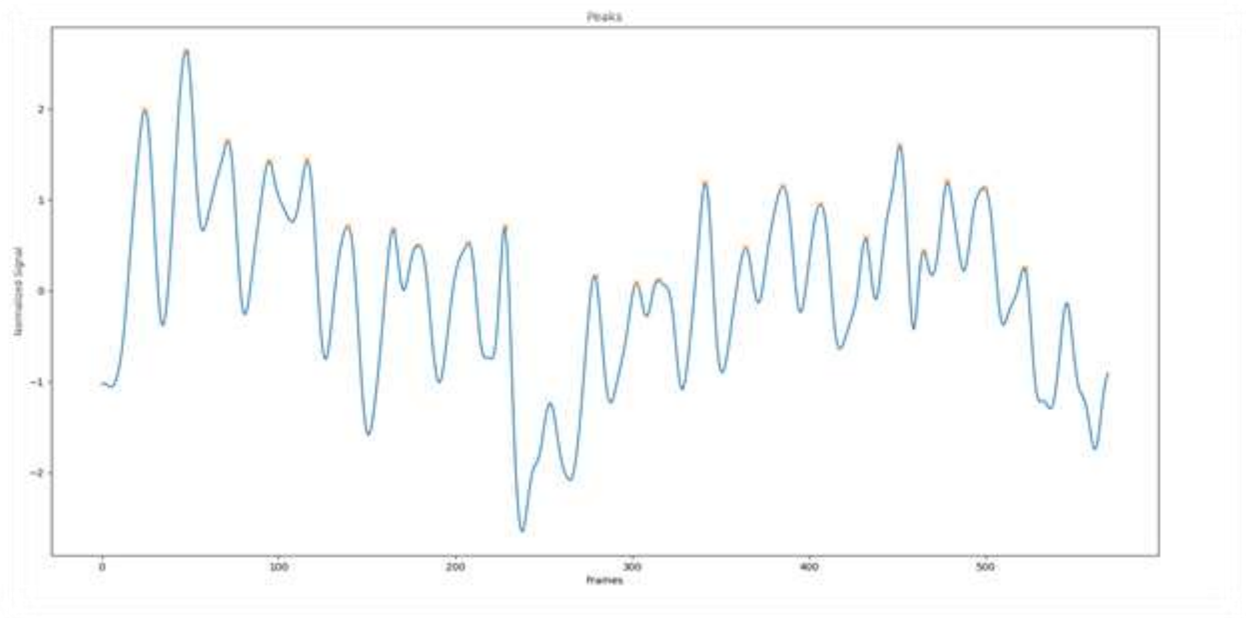

Figure 5: The normalized signal values with peaks having value greater than 0 marked in red

We can see there is a total of 23 peaks for the above sample, within a duration of 19 seconds. Thus, the estimated pulse rate is 72.63 beats/minute.

4.2 ALGORITHM

The method described above is summarized in the algorithm below:

ALGORITHM

```
capture=Read video
fps=Get FPS of Video
duration=Get duration of video
signal=empty array
while Number of frames<duration*fps:
        Extract frame from video
        b,g,r=frame.split() into blue, green and red frames
        signal.append(mean of r)
normalized magnitude of signal=(values-Mean)/standard deviation
Perform FFT to find threshold frequency
Create a low pass filter and pass signal through it
peaks= Count number of peaks in resulting signal value
Pulse rate=60*peaks/(duration-1)
```

5 EVALUATIONS

We carried out an extensive evaluation of our experiments and compared the results with the ground truth, which is the reading from the pulse oximeter. The calculated values and the RMSE obtained can be found in the table below:

Time of the day	RMSE
Morning (Around 10 AM)	7.30
Afternoon (Around 2 PM)	11.75
Evening (Around 5 PM)	6.44
Night (Around 10 PM)	13.25

Here are the sample measurements taken during daytime for the participants:

Participant Name	Gender	Ground Truth (Reading from Pulse Oximeter)	Calculated Pulse Rate(beats/minute)	(Estimated Value-Actual Value) ^2
Ayush Anand	M	84	94.73	115.1329
Somesh Pratap Singh	M	91	82.11	79.0321
Siddhi Pravin Surawar	F	81	87.00	36
Anushka Niti	F	87	94.73	59.7529
Bhoomika Mandloi	F	81	85.26	18.1476
Revant Shah	M	85	88.42	11.6964
			RMSE wrt to Ground Truth	7.300249996

The complete analysis of the RMSE and results obtained for all the experiments can be found in the GitHub repository for this paper. In general, we noticed that our algorithm gives a fair estimate of the pulse rate of the participant and most of the estimates are between 80 and 100 beats/minute, which is consistent with the normal adult pulse rate which lies between 60 and 100 beats per minute. The RMSE with respect to the oximeter reading is also efficient (between 6-15), and close enough to show that this ubiquitous solution can in fact be used.

Since the pulse rate does not vary much contrary to the blood pressure, we did not observe any particular trend in the readings. Some general observations were: (1) The oximeter readings for almost all the participants lies between 80 and 95 (2) It might take 2 or 3 recordings to obtain a correct reading, as the pressure applied by the finger can sometimes disrupt the data and (3) The calculated values are slightly flexible depending on the values of hyperparameters like the minimum height above which peak is to be considered, whether the frame is to be cropped or not before taking the mean of red channel pixel values and so on.

6. CONCLUSION:

In this paper we looked at how to utilize data from smartphone camera to estimate pulse rate of the user. The PPG signal is collected by recording a video from the camera as the individual rests their finger on top of the lens. The signal is extracted based on subtle changes in the video that are due to changes in the light absorption properties of the skin as the blood flows through the finger. We compared our results with the pulse rate measured by a pulse oximeter, and found out that the ubiquitous solution to measuring PPG gives promising results and can, in the very near future, replace the need to opt for separate devices for pulse measurement at all.

REFERENCES

[1] Lovisotto, Giulio, Henry Turner, Simon Eberz, and Ivan Martinovic. "Seeing red: PPG biometrics using smartphone cameras." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 818-819. 2020.

[2] Shu-Di Bao, Yuan-Ting Zhang, and Lian-Feng Shen. Physiological signal-based entity authentication for body area sensor networks and mobile healthcare systems. In 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pages 2455–2458. IEEE, 2006.

[3] Nima Karimian, Mark Tehranipoor, and Domenic Forte. Non-fiducial ppg-based authentication for healthcare application. In 2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), pages 429–432. IEEE, 2017

[4] A Res¸it Kavsaoglu, Kemal Polat, and M Recep Bozkurt. ˘ A novel feature ranking algorithm for biometric recognition with ppg signals. Computers in biology and medicine, 49:1– 14, 2014.

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html

[6] https://www.life2coding.com/split-video-channels-rgb-components-using-opencv-python/