# Supervised learning for heart rate detection in photoplethysmography

Talha Agcayazi

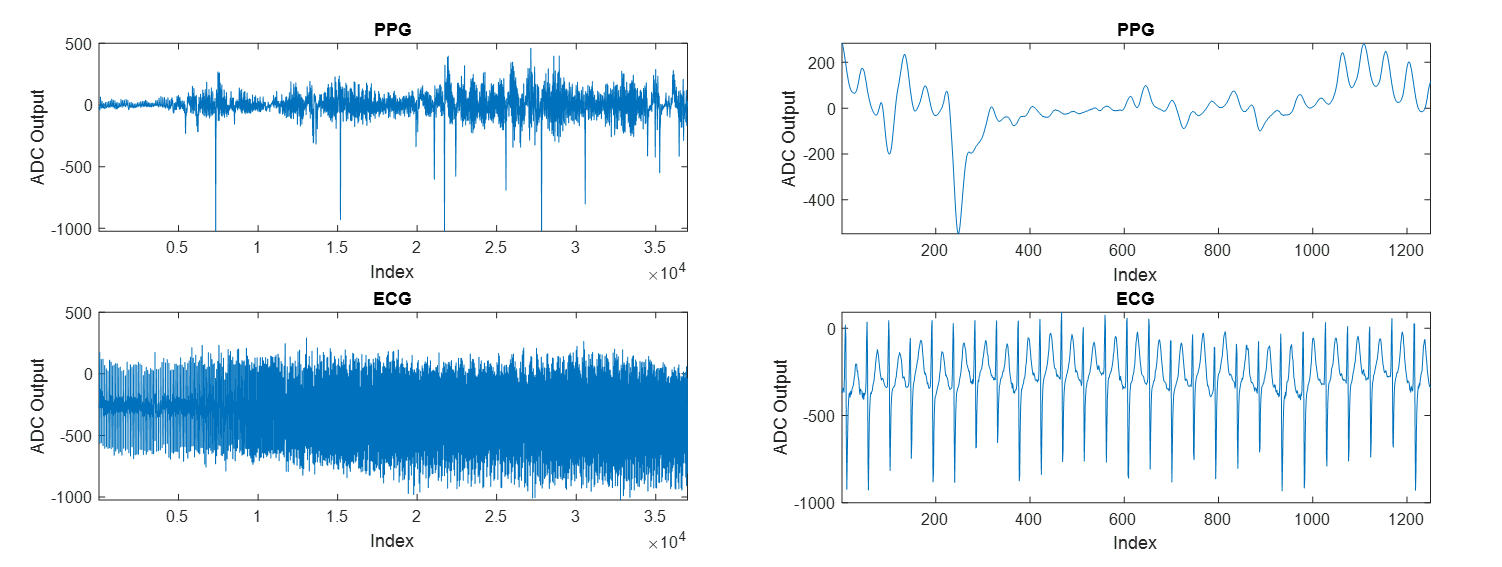
Tanner Songkakul

**Introduction and Motivation**

Photoplethysmography (PPG) is an optical biosensing technique which uses interactions of photons with oxygenated and deoxygenated hemoglobin to determine blood volume in a vessel. This volume is related to heart rate because blood volume varies over time with the pumping of blood by the heart. This technique is used to measure heart rate in almost all modern wearable health trackers. However, commercial fitness trackers demonstrate inaccuracy in their measurement of heart rate. In this project, we used a convolutional neural network (CNN) to identify heartbeat in PPG segments and compared its accuracy in identifying heart beats with other traditional clustering and classification techniques.

**Overview of Supervised Learning Approach**

Data from the TROIKA dataset was used as the source data for this project [x]. The TROIKA dataset contains 2 channels of wrist PPG data, electrocardiogram data (ECG), and accelerometer data. This data was collected from patients running on a treadmill over the course of a total of 5 minutes, varying in speeds of 1–2 km/h, 6–8 km/h, and 12–15 km/h at set intervals. The sampling rate was 125 Hz. During data collection, patients were encouraged to pull clothes, wipe sweat, push buttons, and otherwise mimic a real-world exercise situation.



After PPG and ECG data from TROIKA is imported into Matlab, it is preprocessed to reduce noise and normalize the signal. The ground truth heartbeat data is identified from the ECG data and used to extract segments of the PPG data containing heartbeat vs no heartbeat. Features are extracted from the PPG segments, and various classifiers are applied to distinguish heartbeat from no heartbeat.

**Preprocessing PPG Data**

Preprocessing was carried out as defined by Grisan et al [x]. The preprocessing stage is necessary to reduce dynamic range of the PPG signal from motion, remove any trends, and reduce noise. Mean and standard deviation is found for windows of length FsN for N=8. These means and standard deviations are used to normalize the PPG signal.

The normalized PPG signal trend is found by a moving average filter of length F\_sN for N=2. This trend is then subtracted from the normalized PPG signal, resulting in a detrended signal. The final step is a 7th order Butterworth filter with Fc=18 Hz. The output of this filter is the preprocessed data.



**Ground Truth Extraction**

Grisan et. al approach was also followed for ground truth extraction. Peak locations in the ECG signal were identified using Pan-Tompkins Algorithm.



These peak locations were used to find the heartbeat (HB) segments contained in the PPG signal starting at Ri and spanning ΔRi=Ri+1-Ri. These same locations were used to find the no heartbeat (NHB) segments by offsetting by half of the interval, therefore centering this the NHB PPG signal midway between beats using Ri+0.5ΔRi. The HB and NHB segments of the PPG signal were then resized using stretching and interpolation to the maximum segment size of length 140. 56632 segments in total were extracted from the PPG data.



**Feature Extraction**

The features in Table 1 were extracted and labeled with HB or NHB. The 3 level quantized version is derived in Equation 2. **Principle component analysis was….** Segment energy was calculated by finding the frequency domain energy spectrum from the FFT.

|  |  |
| --- | --- |
| **Feature Description** | **Features** |
| Sample | 140 |
| Sample Mean | 1 |
| Sample Standard Deviation | 1 |
| 3 level quantized version of sample | 140 |
| Principle Component Analysis | 12 |
| Maximum Intensity Position | 1 |
| Minimum Intensity Position | 1 |
| Time difference between maximum and minimum position | 1 |
| Segment energy in [0.04, 0.09], [0.09,0.15], [0.15,0.60] Hz bands | 3 |

**Machine Learning Techniques**

To classify between HB and NHB segments we decided to first try traditional classification techniques as in Grisan et al’s work. We tested linear discriminant, k-nearest neighbor (1 and 5-nn), decision tree and support vector machine (linear and cubic) classifiers using Matlab’s Classification Learner Applet. For these tests, we used all 302 of the features with 5 fold validation.

**Convolutional Neural Network Implementation**

Convolutional Neural Networks became increasingly popular because of their ability to learn the important features during training. We implemented this technique with the help of a tutorial.

**Results and Discussion**

**85% accuracy discussion**

**Conclusions**

**Future Work**

**Sources**