Blood Pressure Estimation using PPG Signals

•••

CPSC-554X
Machine Learning and Signal Processing

Sahar Mavali - Ali Seyfi

Agenda

- Introduction
- Methods
  - Artificial Neural Network (ANN)
  - Random Forest
  - Convolutional Neural Network (CNN)
- PPG with Mobile Phone
- Future Works

Introduction

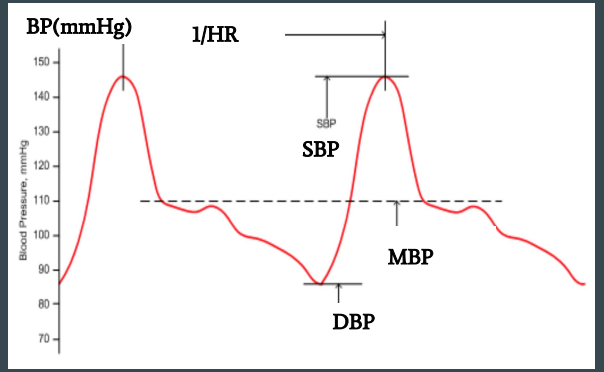
- PPG signals are optically obtained signals that measure the changes in light absorption in our vessels.

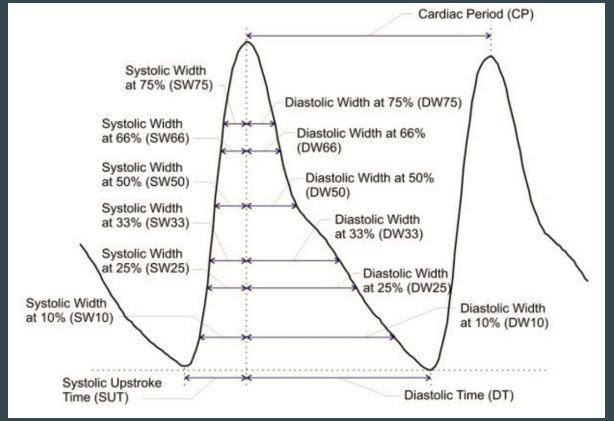
- Cuff-based blood pressure measurement
  - Discomfort
  - Mobility Limitations

- PPG signals can be measured with mobile phones and smart watches.

Feature Extraction (Time Domain)

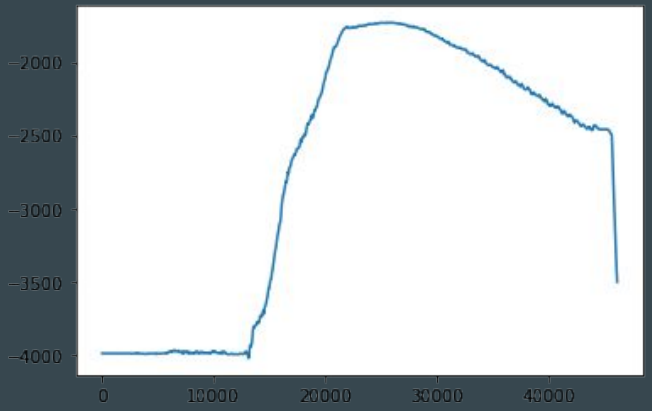
- Extracting windows
- Removing misalignment
- Removing outliers
- 1050002 pulses
- 70% training, 15%validation, 15% testing
- 21 parameters per pulse in PPG

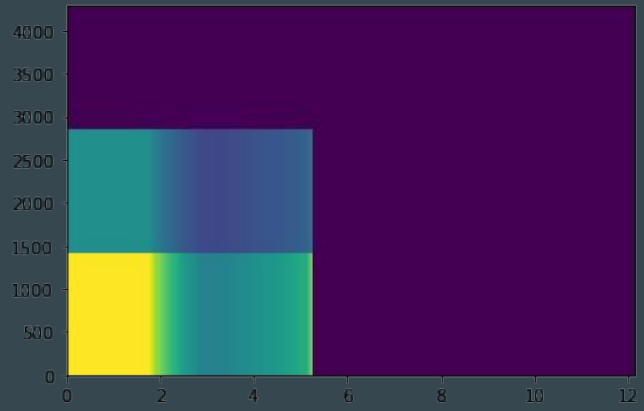




Feature Extraction (Frequency Domain)

- Data cleaning
- Windowing
- Normalizing window sizes
- STFT





Code can be found at https://github.com/aliseyfi75/BP-Estimation-PPG/blob/main/Codes/CNN.ipynb

ANN

- 2 layers -> (50, 30)
- 3 layers -> (60, 60 60)
- Activation function -> relu
- Loss function -> mean squared error
- Optimizer -> adam
- Learning rate -> 0.001

	Mean Absolute Error (mmHg)	Standard Deviation(mmHg)
Systolic Blood Pressure	10.11	13.56
Diastolic Blood Pressure	5.35	6.80

Code can be found at https://github.com/aliseyfi75/BP-Estimation-PPG/blob/main/Codes/ANNpy

Random Forest Regression

- Using time domain features
- Number of trees -> 100
- Criterion -> Mean absolute error

	Mean Absolute Error (mmHg)	Standard Deviation(mmHg)
Systolic Blood Pressure	6.55	12.11
Diastolic Blood Pressure	3.28	5.74

Code can be found at https://github.com/aliseyfi75/BP-Estimation-PPG/blob/main/Codes/RandomForest.py

K Nearest Neighbor

- Neighbors -> 7

CNN

- 2 convolutional layers -> Kernel Size : 3 -> activation function : tanh
- A Max Pooling layer after each layer
- 3 convolutional layers -> Kernel Size : 3 -> activation function : tanh
- A Max Pooling layer after last layer
- 2 Fully Connected layers -> activation function : ReLu
- Loss function -> mean absolute error
- Optimizer -> adam

	Mean Absolute Error (mmHg)	Standard Deviation(mmHg)
Systolic Blood Pressure	19.11	17.82
Diastolic Blood Pressure	10.71	12.05

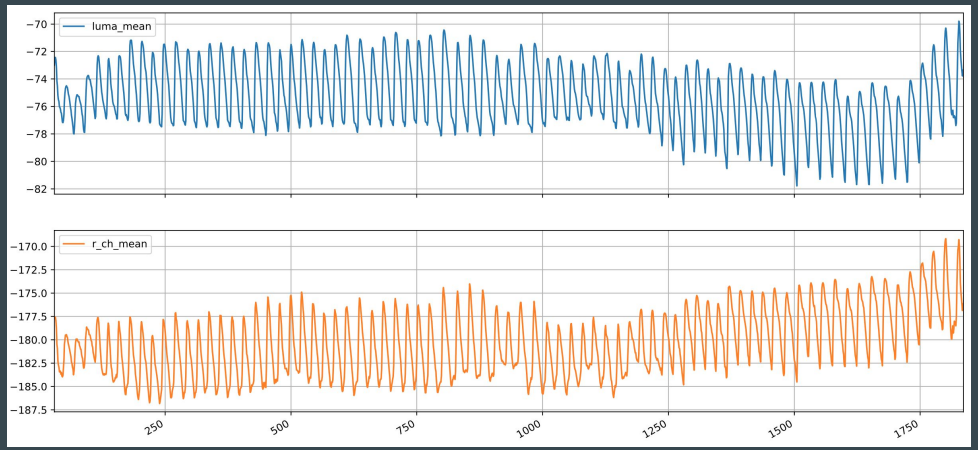
Code can be found at https://github.com/aliseyfi75/BP-Estimation-PPG/blob/main/Codes/CNN_GPU.ipynb

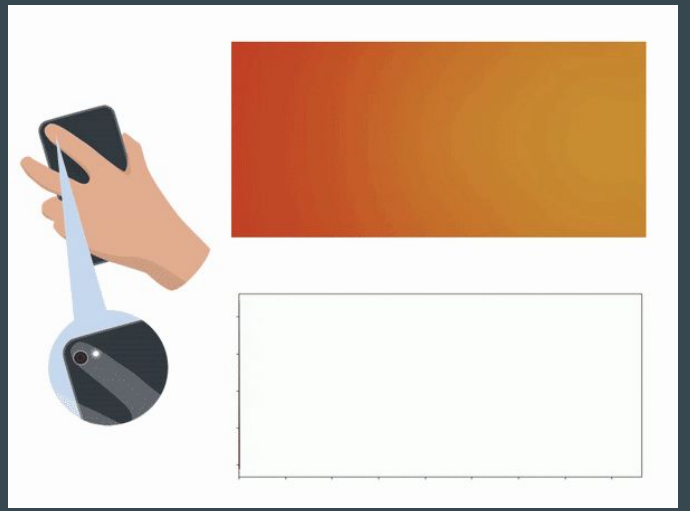
Seeing Red: PPG Biometrics Using Smartphone Cameras

- A collaboration between Giulio Lovisotto, Henry Turner and Simon Eberz from the System Security Lab at University of Oxford
- Based on reflection, not absorption
- Place the finger lightly over both the camera and camera flash
- Fix the camera ISO and exposure time to be the minimum possible
- Set the white balance gain to be the maximum in the red channel, and minimum in the blue and green channels
- Steps: (Just first two steps are needed in our work)
  - Signal Extraction : compute the mean of the pixel-wise luma component
  - Signal Preprocessing : rolling average (window size of 1 second), low pass filter(cutoff freq. : 4Hz)

PPG with Mobile Phone

- Seeing Red: PPG Biometrics Using Smartphone Cameras [Link in the footnote]
- How it works
- Output



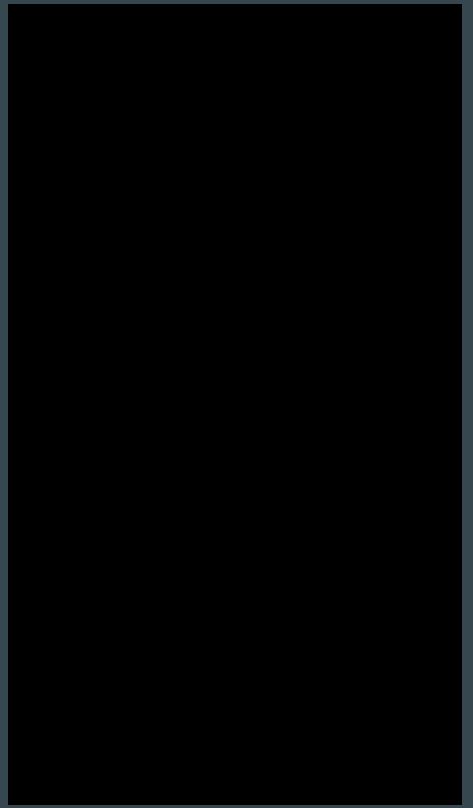


Code can be found at https://github.com/ssloxford/seeing-red

Study

- 6 participants (3 female, 3 male)
- 60 second videos
- Measured 3 times for each user
- Compared with home cuffbased BP measurement

	Mean Absolute Error (mmHg)	Standard Deviation(mmHg)
Systolic Blood Pressure	33.51	19.64
Diastolic Blood Pressure	6.30	4.95



Future Work

- Improve feature extraction and model structures
- Use calibration for better results
- Have a mobile application for extracting PPG signals and sending them to a server for BP estimation
- Expand our user study with more participants