BMI598 Embedded Machine Learning

Embedded Heart Rate Estimation Device

Arizona State University

Jacob Sindorf
F3

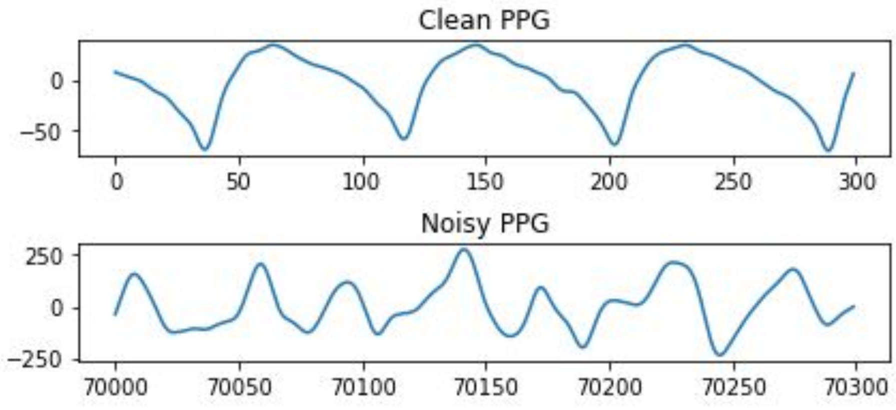
Introduction

Problem Statement

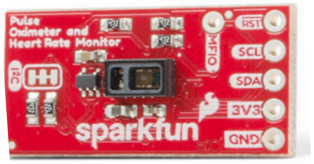
Problem: Accurate heart rate (HR) estimation using Photoplethysmography (PPG) becomes challenging with the introduction of noise, such as motion artifact (MA), limiting the environments a wearable device can be used in.

Importance: Being able to interpret noisy sensor data increases the accuracy and use cases of HR estimation devices

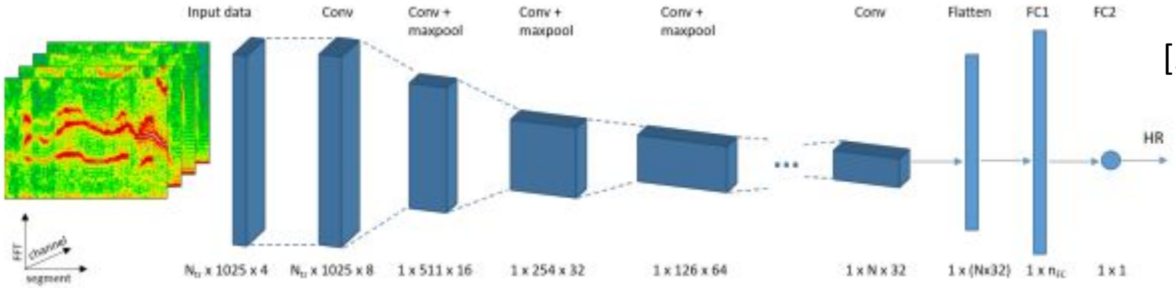
General use example: Real time HR monitoring of someone moving with a smaller constrained device



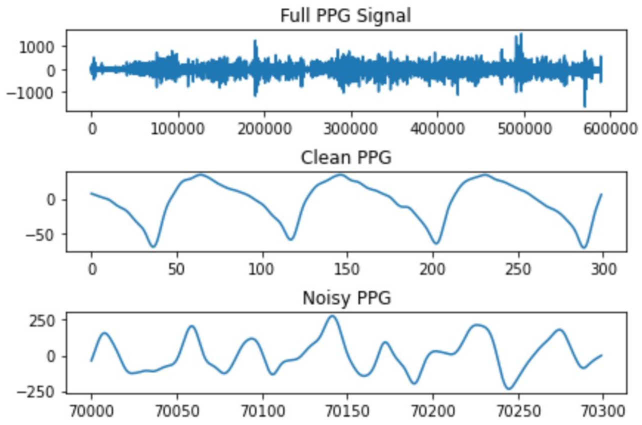
State of the Art

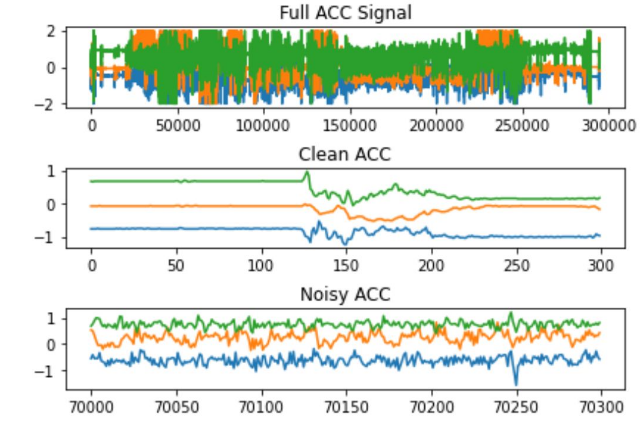


- Many commercially available HR sensors exist but struggle with noisy signals
- Many estimation techniques have been described and summarized in [1]
  - Success found with convolutional neural networks (CNN)
- New PPG dataset (PPG-DaLiA) created [2]
  - Accurate HR estimation achieved
  - Reduced CNN model for embedded application proposed



[2]





*Data from S1

Dataset

Dataset:

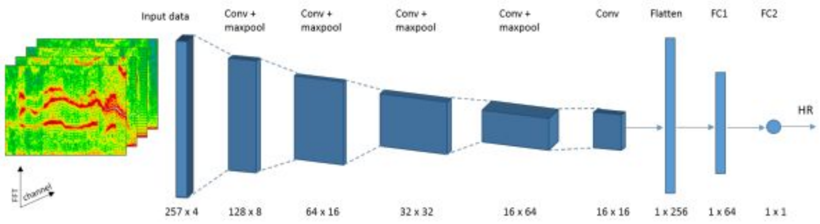
PPG-DaLiA [2]

Link:

https://archive.ics.uci.edu/ml/datasets/PPG-DaLiA

- 15 subject (~9000 seconds each)
  - PPG raw data (64Hz)
  - 3 axis accelerometer (32Hz)
  - Ground truth HR in BPM
- Stored in .pkl files
- Extract S1-S15, keeping:
  - data['label'] (Ground truth BPM)
  - data['signal']['wrist']['BVP'] (PPG)
  - data['signal']['wrist']['ACC'][:,0,1,2] (acc x y z)

Proposed CNN Model

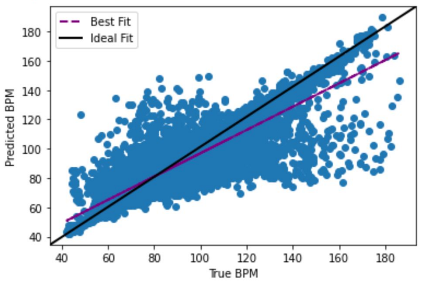


[2]

Preprocessing:
- Windowing 8/2 seconds
- FFT
- Z-score (0 mean, 1 standard deviation)
- Stack: (257x4)

Problems towards Arduino deployment:
- Complex numbers
- Large model
- Arduino compatibility with Conv1D
- Multidimensional Array

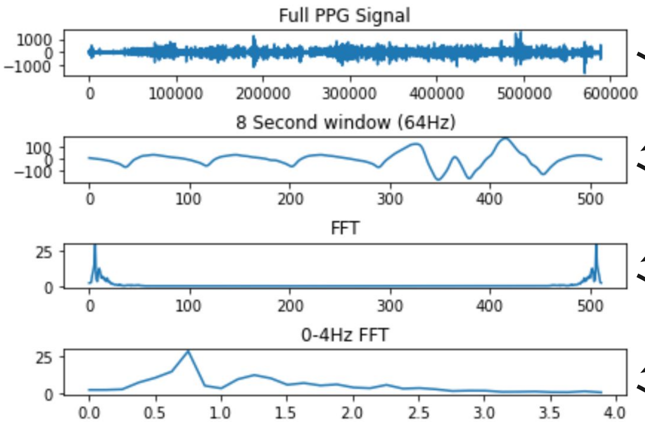
CNN replica results



Results within a threshold:
Within 2 BPM: 45%
Within 5 BPM: 70%
Within 10 BPM: 85%

Methods/Results

Preprocessing Change



Z-score ← 32 real samples

Changes to preproccess from [2]

- Real instead of Complex
- 1D array

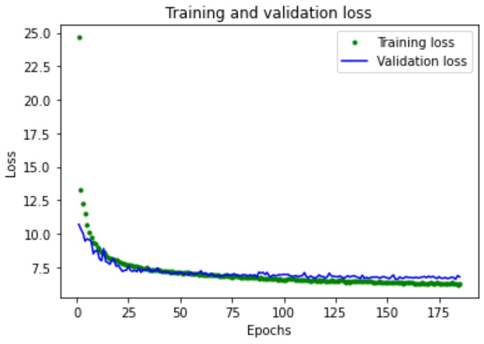
Repeat for accelerometer X,Y,Z → 32 Real PPG, 32 Real Acc X, 32 Real Acc Y, 32 Real Acc Z → Stack → $[\ \text{PPG}\ \ X\ \ Y\ \ Z\ ]_{128x1}$

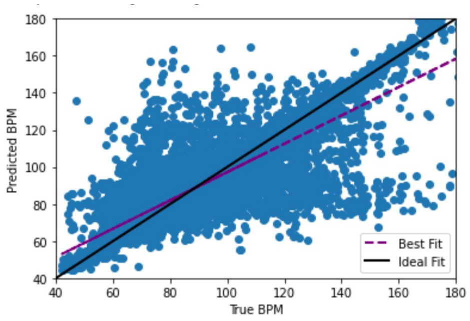
```
Full data shape: (64687, 128)
Full label shape: (64687,)

size of training dataset: 38811
size of validation dataset: 12938
size of testing dataset: 12938
```

Deep Neural Network (DNN) Model

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 64)                8256

 batch_normalization_3 (Batc  (None, 64)               256
 hNormalization)

 dropout_3 (Dropout)         (None, 64)                0

 dense_5 (Dense)             (None, 64)                4160

 batch_normalization_4 (Batc  (None, 64)               256
 hNormalization)

 dropout_4 (Dropout)         (None, 64)                0

 dense_6 (Dense)             (None, 64)                4160

 batch_normalization_5 (Batc  (None, 64)               256
 hNormalization)

 dropout_5 (Dropout)         (None, 64)                0

 dense_7 (Dense)             (None, 32)                2080

 dense_8 (Dense)             (None, 16)                528

 dense_9 (Dense)             (None, 8)                 136

 dense_10 (Dense)            (None, 1)                 9

=================================================================
Total params: 20,097
Trainable params: 19,713
Non-trainable params: 384
_________________________________________________________________
```

MAE



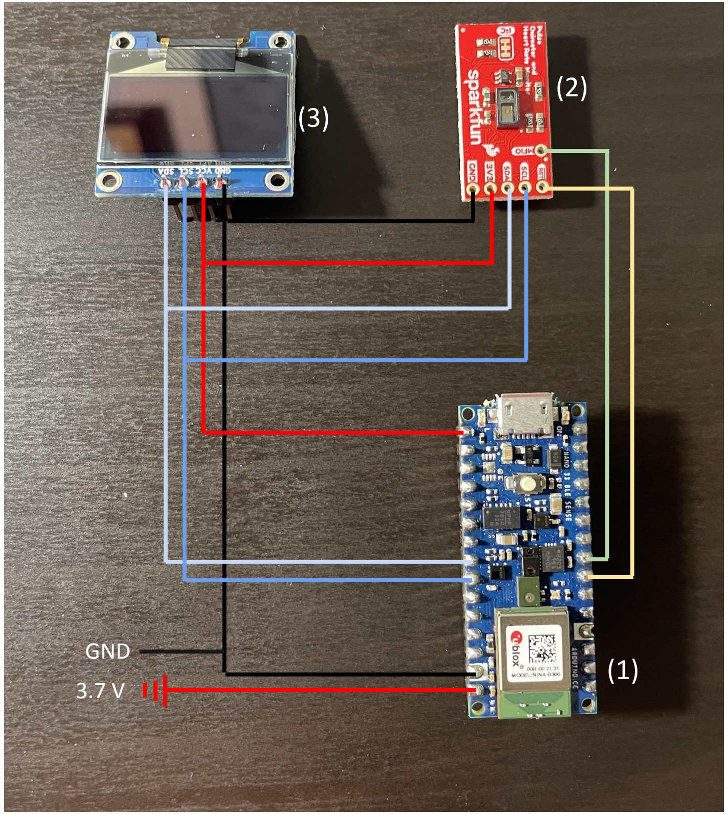


	Train	Validate
Final MAE:	6.27	6.78
Final r^2:	0.75	0.67

Epochs: 317
Batch Size: 64

Results within a threshold:
Within 2 BPM: 42%
Within 5 BPM: 68%
Within 10 BPM: 81%

System Hardware



Arduino Nano 33 BLE Sense (1)

SparkFun POHR (2)

SparkFun Pulse Oximeter and Heart Rate Sensor (POHR) - MAX30101 & MAX32664 (Qwiic)

OLED Screen (3)

UCTRONICS 0.96 Inch OLED Module 12864 128x64 Yellow Blue SSD1306 Driver I2C Serial Self-Luminous Display Board

3.7 V Battery

Arduino Sensor Preprocessing

Raw Data

Arduino LSM9DS1 (X,Y,Z Accelerometer)

- Main Library: Arduino_LSM9DS1
- 104Hz

SparkFun POHR - Raw PPG from IR LED

- Main Library: SparkFun_Bio_Sensor_Hub_Library
- 100 Hz
- Moving Average for Smoothing
- DC Bias removal

$$w(t)=x(t)+0.95*w(t-1)$$
$$y(t)=w(t)-w(t-1)$$

x(t) = raw IR signal
y(t) = bias removed signal

	Output
Collect (1/64Hz delay)	512 PPG 512 Acc X 512 Acc Y 512 Acc Z
Magnitude FFT	32 PPG 32 Acc X 32 Acc Y 32 Acc Z
Z-Score	32 PPG 32 Acc X 32 Acc Y 32 Acc Z
Stack	$[\ \mathrm{PPG}\ \ X\ \ Y\ \ Z\ ]_{128x1}$

Model Deployment

- Convert to tflite (size: 81292 bytes)

- Input model.cpp to main arduino code

```
unsigned char g_model[] = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
  0x04, 0x00, 0x08, 0x00, 0x0c, 0x00, 0x10, 0x00, 0x14, 0x00, 0x00, 0x00,
```

...

```
  0x80, 0x00, 0x00, 0x00, 0xfc, 0xff, 0xff, 0xff, 0x04, 0x00, 0x04, 0x00,
  0x04, 0x00, 0x00, 0x00
};
unsigned int g_model_len = 81292;
```

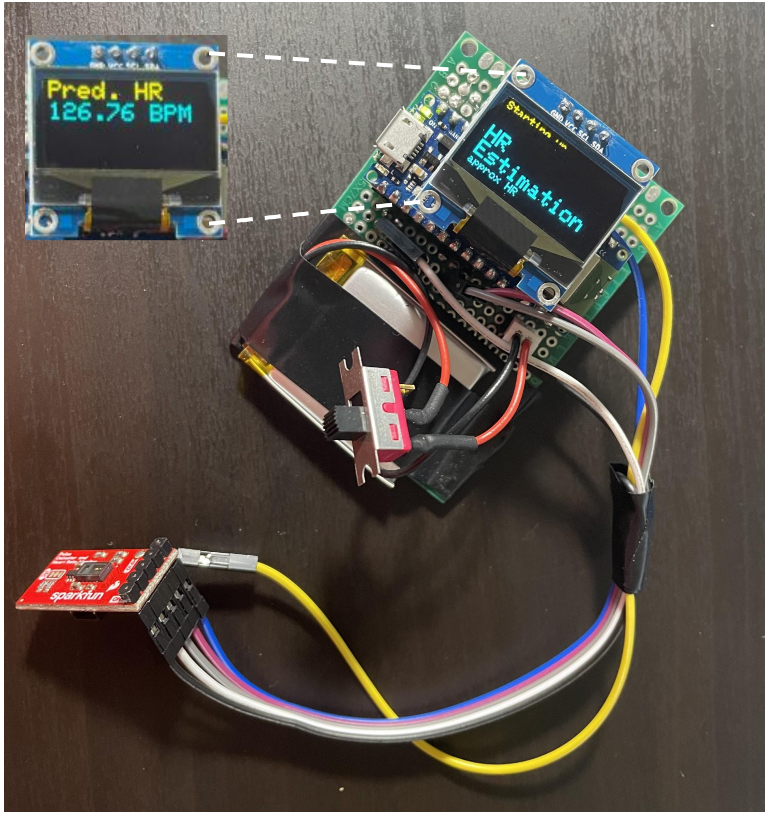
Model Tensors

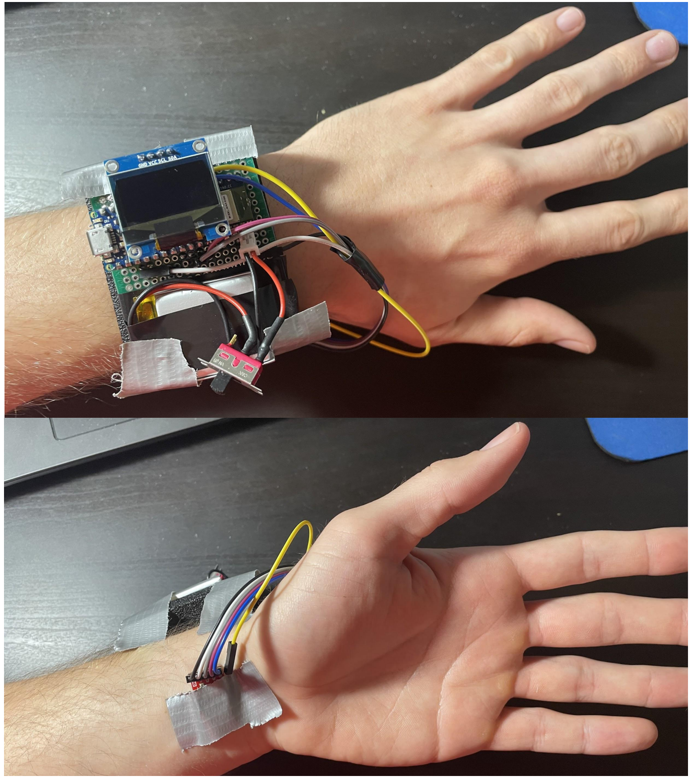
```
input = interpreter->input(0);
```
Size: (128x1)

```
output = interpreter->output(0);
```
Size: 1

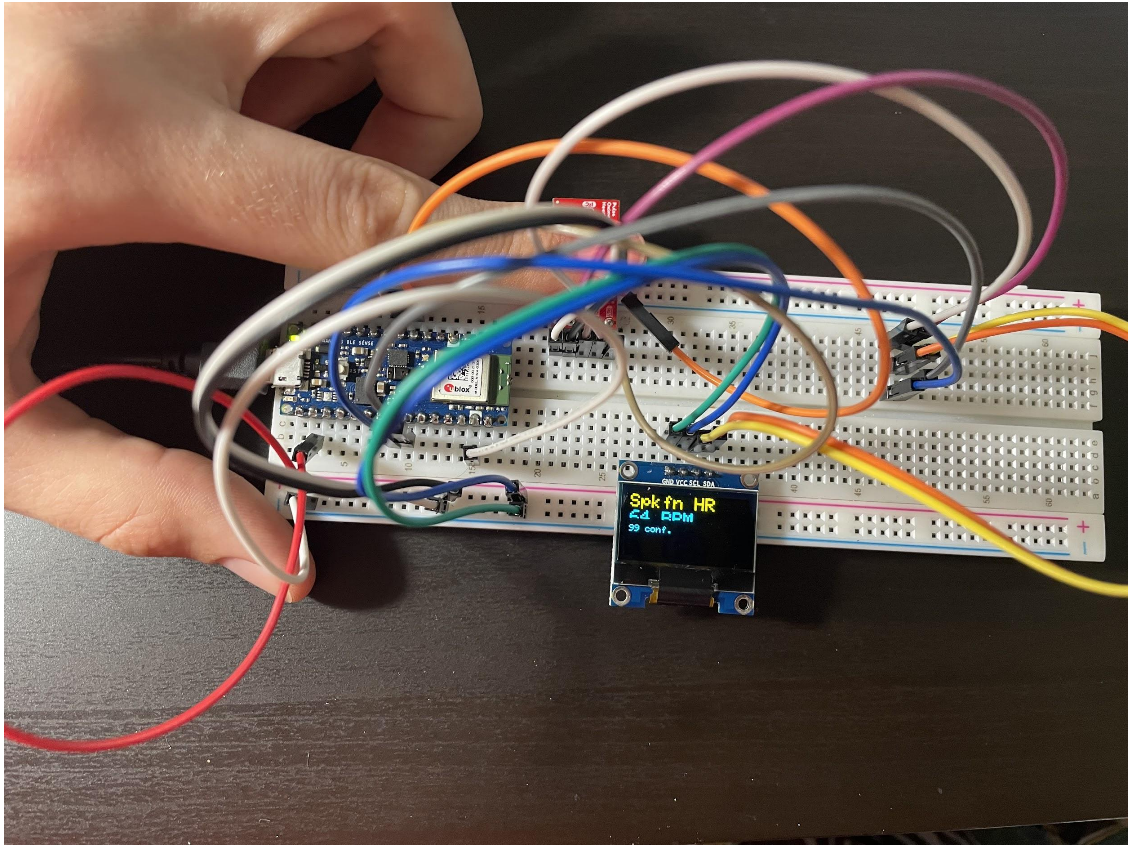
Make Predictions!

Wearable Design





System Verification



SparkFun POHR built in HR Monitor

Displays “confidence” in Approximated HR

Conclusions and Future Work

- Fine tune the model $\longrightarrow$ Get higher accuracy
    - Retry CNN (2D) and Quantization (to reduce larger model)
- Allow for the system to be used with more sensors
    - More work on Sensor preprocessing
    - Explore transfer learning
- Finalize wearable design

A working framework for a wearable HR estimation device is shown. With some future work, this framework can easily be expanded into multiple uses and projects.

Live Demonstration!

Questions?

References

[1] D. Biswas, N. Simões-Capela, C. Van Hoof and N. Van Helleputte, "Heart Rate Estimation From Wrist-Worn Photoplethysmography: A Review," in IEEE Sensors Journal, vol. 19, no. 16, pp. 6560-6570, 15 Aug.15, 2019, doi: 10.1109/JSEN.2019.2914166.

[2] Reiss, A.; Indlekofer, I.; Schmidt, P.; Van Laerhoven, K. Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks. Sensors 2019, 19, 3079. https://doi.org/10.3390/s19143079