Non-Invasive Analysis For Health [1]

Presenter: Hieu

Week 2, November 2025

[1] Based on the paper "A deep learning approach for remote heart rate estimation"

Outline

1: LSTM and general framework

1.1 General framework of VPG-based heart rate estimate

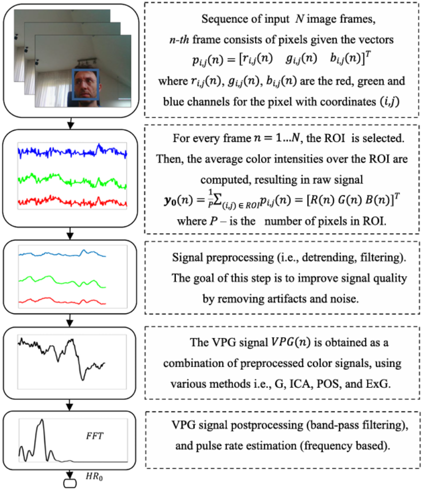


Fig. 1. Steps of pulse rate estimation from facial video using videoplethysmography.

Step 1: ROI selection

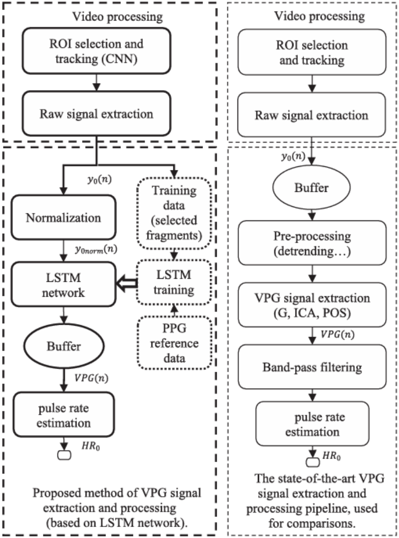
Step 2: raw color signal extraction and preprocessing

Step 3: VPG signal extraction

Step 4: VPG signal postprocessing

Step 5: pulse rate estimation

1.2 LSTM network



Step 1: ROI selection

Step 2: raw color signal extraction and preprocessing

Step 3: LSTM network

Step 4: pulse rate estimation

2. Analysis

2.1: In paper

- **Processing Pipeline:** Proposed a main processing pipeline: `Video -> ROI (CNN) -> Raw Signal -> LSTM Network -> Clean VPG Signal -> HR Estimation (PSD)`.
- **Objective of LSTM:** The LSTM network was designed to **completely replace** complex signal preprocessing steps (like detrending, band-pass filtering).
- **Training:**
  - **Data Augmentation:** A smart data augmentation technique was used. It started with only **3 "clean"signal samples** (70, 88, 96 bpm) and combined them with "realistic"noise samples (which were *shifted* and *scaled*) to create **603 training signal sequences**.
  - **Ground Truth:** The "clean"PPG signal was collected from a specialized sensor (PolarOH1).
  - **Parameters:** Trained in **MATLAB**, using the **Adam** optimizer, **MSE** loss function, **16 Epochs**, and a Minibatch size of **4**.
- **Results**: The LSTM method (RMSE = 3.26 bpm) achieved **highly competitive accuracy**, comparable to advanced methods like POS (2.61 bpm) and ICA (3.02 bpm). G_pre (6.49 bpm)

Training:

- **Environment:** Used Python, TensorFlow, and Keras (instead of MATLAB from the paper).
- **Data Augmentation:** The augmentation technique described in the paper was precisely reproduced (using `np.roll` for "shift"and `scale_factor` for "scale") to generate 603 training samples.
- **Model Architecture:** A specific model was built:
  - 2 stacked LSTM layers (units: 128 and 64).
  - Used `Dropout(0.2)` (compared to 0.5 in the paper).
- **Training Parameters:** Differed from the paper to fit Keras:
  - **Batch Size:** 32 (vs. 4 in paper).
  - **Epochs:** 50 (with `EarlyStopping`) (vs. 16 in paper).

2.2: My implementation

- Dataset for training and evaluation: MERL-Rice Near-Infrared Pulse (MR-NIRP) Car Dataset for Camera-Based Vital Signs Estimation in Narrow-Band Near-Infrared During Driving (https://computationalimaging.rice.edu/mr-nirp-dataset/)
- Result: loss (MSE): 0.4323 - val_loss (MSE): 0.3915