

Cardiopulmonary Resuscitation Assistant

A smartphone app for detecting the chest compression rate during CPR (dissertation research). Evaluates the chest compression rate from the bottom-up view by harnessing and thresholding the acceleration of pixels. Tested with 88% accuracy over ideal and non-ideal environments. Report PDF available in repository. Visualisations available at cpr-assistant.herokuapp.com.

Python

The solution was implemented in Python to prove the feasibility of the concept.

Installation

Requires Python ≥ 3.5 , `pip3` and `virtualenv` to be installed on your machine.

1. Install `virtualenv` globally using `pip3` with `pip3 install virtualenv`.
2. Create a `virtualenv` named "venv" with `virtualenv venv`.
3. Activate the `virtualenv` with `. venv/bin/activate`.
4. Install dependencies with `pip3 install -r requirements.txt`.

Datasets (Test Cases)

Code	Test Case
L	Subject has long, loose hair.
S	Subject has short hair.
B	Background disturbances wholly intended to disrupt performance.
C	Crop at neck-level (the subject's neck-level and below is cropped out).

Graphs

Provided that the CSV files are in the correct directories, interactive graphs may be viewed using the following command:

```
python3 solution.py -d=DATASET where DATASET can be one of L1, L2, L3, S1, S2, S3, S4, LB1, LB2, LB3, SB1, SC1, SCB1 .
```

Ground Truth

- Ground truth data was extracted from front view videos, designated with the suffix **GT** (e.g. **L1_GT.mp4**) using the following command:

```
python3 ground_truth_extractor.py -i=L1_GT.mp4 -v -o
```

(warning: this overwrites the existing data used for graphing).
- This command calculates ground truth for dataset **L1** from the video at `./videos/GT/L1_GT.mp4` and dumps the data in a CSV

at directory `./csv_gt/L1_GT.mp4.csv`.

- CSV files outputted by the script contain the following data per column: (X, Y, S) where X is the X-coordinate of the ball in the frame, Y the Y-coordinate of the ball in the frame, and S the state observed in the frame: `S = { None, Compression, Breathing }`. Each row denotes a frame.
- For more details on command-line arguments, run `python3 ground_truth_extractor.py --help`.

Solution

- Solution data was extracted from the bottom-up view videos, designated with the suffix **BUV** (e.g. **L1_BUV.mp4**) using the following command:
`python3 solution.py -d=L1 -r -o -w=./weights/S1.npy`
(warning: this overwrites the existing CSV used for graphs).
- This command extracts solution data for dataset **L1** from the video at `./videos/BUV/L1_BUV.mp4` using preloaded weights at `./weights/S1.npy`.
- The script `solution.py` contains parameters that may be configured accordingly: `BREATHING_MODE`, `LEARNING_RATE`, `LOOKBACK_TIME`, `MINIMUM_ACCELERATION`, `MOVING_AVG_PERIOD`, `MAX_TIME_FOR_UPWARD_ACCELERATION`, `MIN_MOVEMENT_PCG`, `MIN_FLOW_THRESHOLD`, `SCALE`, `MIN_BREATHING_MOVEMENT`.
- CSV files outputted by the script contain the following data per column: (vertical displacement, upward displacement sum, downward displacement sum, S, total percentage of pixels moved, vertical acceleration) where S is the state observed in

the frame: `S = { None, Compression, Breathing }`. Each row denotes a frame.

- The resulting weights after each execution may be saved by providing the `-s` flag.
- For more details on command-line arguments, run `python3 solution.py --help`.

Android

An Android application was implemented with similar logic to the Python solution. This requires further device-specific optimisations.