Supplemental materials for MobileClinic: An end-to-end software architecture foranalyzing human movement on a mobile device

March 2018

1. Supplementary materials

1.1. Pose Estimation

The mobile clinic app captures a patient performing a Sit-to-Stand test before passing each frame through a pose estimation model (OpenPose) to output 2 dimensional JSON information about body keypoints. OpenPose works by extracting body parts from heatmaps (matrices representing confidence the network has that a pixel contains a certain body part) and Part Affinity Fields (provide information about the orientation of pairs of body parts).

The OpenPose model was exported to a CoreML (.mlmodel) file and utilized within the MobileClinic app to allow for on-the-fly keypoint detection.

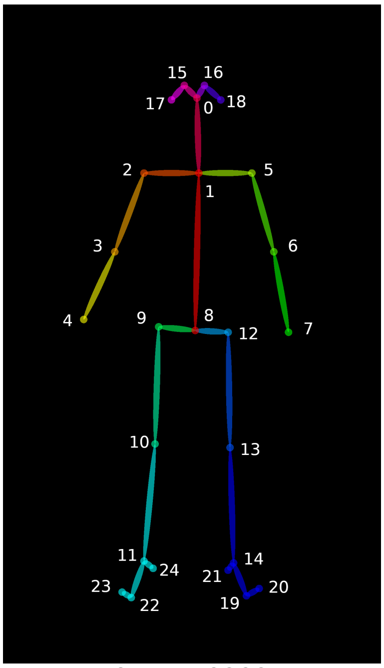

Figure 1: Standard COCO Keypoints

1.2. Signals Computation

The COCO keypoints (See figure 1) were manipulated to store the following information:

- left and right leg angle
- height of head
- x, y position of right and left hip

1.3. Linear Interpolate

Our model is a simplified version of OpenPose, which is unable to identify keypoints in some poses, thus we encounter numerous NaN values in the multivariate time series. Before proceeding in analysis, we linearly interpolate these values to provide a constantly sampled signal. The matrix operations required for linear interpolation were implemented using the Accelerate Swift Library.

Given a point to linearly interpolate (x, y) between (x_1, y_1) and (x_2, y_2), the linear interpolation can be calculated as follows

$$y = y_1 + (x - x_1)\frac{y_2 - y_1}{x_2 - x_1}$$

1.4. Normalization and Filtering

Due to the fact that the different metrics computed operate on different scales, normalization of each signal was necessary to ensure that large ranging metrics didn't wash over finer ones. Each signal (see section on Signals Computation above) was min-max normalized where X represents the signal, X_{min} is the minimum value in the signal and X_{max} is the maximum value in the signal.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

To remove irregularities due to noise and interference, a moving average filter with kernel size of $m = 3$ input vector f of length n was slid over the summed signal in the following linear 1D convolution.

$$(f(t) * g(t))(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + \frac{m}{2})$$

Where $g(t) = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and $f(t)$ is the original signal.

1.5. Periodogram

Periodograms measure magnitude of information on different frequencies of the signal. For our offline analysis we use a numpy implementation of the Fast Fourier Transform. Our computation requires an array of M complex values $x_0, x_1 ... x_{M-1}$ with i being the imaginary unit.

$$X_k = \sum_{j=0}^{M-1} (x_j)(e^{-i2\pi kj}), k = 0, 1, ...M - 1 \quad (1)$$

$$\hat{P} = \frac{2\Delta t}{N} |X_k|^2, f_k = k/\Delta t, k = 0, 1, ..., N - 1 \quad (2)$$

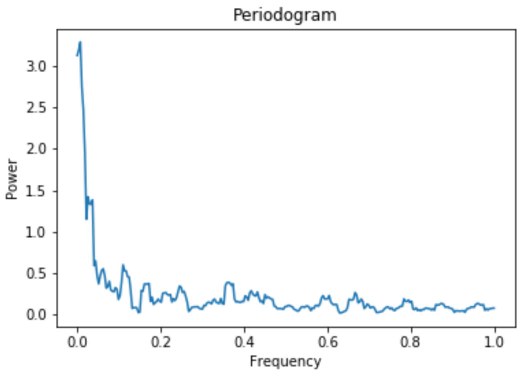

Figure 2: Periodogram

1.6. Regularity

Regularity was calculated to determine the ease with which the patient performed the squat. The cubic spline interpolation simulated a fluid and smooth squat common in physically fit individuals. The spline interpolation was then compared with the original signal using the cosine similarity.

The cosine similarity compares two nonzero vectors by taking the cosine of the angle between them. Cosine similarity solely takes into account the angle between the two vectors, therefore it does not measure magnitude, only the orientation. Let A be the spline interpolation and be B the original signal. Then $cos(\theta)$ represents the cosine similarity.

$$cos(\theta) = \frac{A \cdot B}{||A||||B||}$$

1.7. Analyze

In order to determine the number of squats attempted, the fundamental frequency of the signal needed to be computed - involving interpretation of the periodogram computed. The fundamental frequency (frequency with the greatest power) was determined by finding the argmax of the periodogram. To ensure that less pronounced peaks could be flagged by the system, a sliding window peak finding algorithm was written with window size of 3. Then the frequency of the peak with the greatest power was chosen from those flagged by the peak-detector. Computing the period from the frequency, allowed us to determine the number of squats the patient performed in the given amount of time.