

Detection of the type of physical activity based on an IMU sensor

Auteur : Paolino, Alessia

Promoteur(s) : Bruls, Olivier; Schwartz, Cédric

Faculté : Faculté des Sciences appliquées

Diplôme : Cours supplémentaires destinés aux étudiants d'échange (Erasmus, ...)

Année académique : 2023-2024

URI/URL : <http://hdl.handle.net/2268.2/20850>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Appendix A

Matlab R2024a scripts

The analysis employed the following functions for feature extraction.

The function `fmedian` calculates the median frequency of a given signal epoch. It takes three inputs: the signal epoch x , the sampling frequency f_{sam} , and the length of the epoch in seconds `epoch_len`.

```
1 function fmedianv = fmedian(x, fsamp, epoch_len)
2 % Function to calculate the median frequency of a signal
   epoch.
3 % Input parameters:      x           signal epoch (vector)
4 %                       fsamp        sampling frequency
5 %                       epoch_len    epoch length (in seconds
   )
6 % Output parameters:    fmedianv     median frequency
7
8 % Calculate P using a simple periodogram
9 x = x - mean(x);
10
11 win = rectwin(length(x));
12 noverlap = 0;
13 fNy = fsamp / 2;
```

```
14 df = 1 / epoch_len;
15 NFFT = fsamp / epoch_len;
16
17 [P, f] = pwelch(x, win, noverlap, NFFT, fsamp);
18
19 A = sum(P) / 2;
20 i = 1;
21 S = 0;
22 while S < A
23     S = S + P(i);
24     i = i + 1;
25 end
26
27 fmedianv = (f(i) + f(i - 1)) / 2;
28 end
```

Listing 1: MNP function's script

The function `fmean` calculates the mean frequency of a given signal epoch. It takes three inputs: the signal epoch x , the sampling frequency f_{samp} , and the length of the epoch in seconds `epoch_len`.

```
1 function fmeanv = fmean(x, fsamp, epoch_len)
2 % Function to calculate the mean frequency of a signal epoch.
3 % Input parameters:      x          signal epoch (vector)
4 %                       fsamp       sampling frequency
5 %                       epoch_len   epoch length (in seconds
6 % Output parameters:    fmeanv     mean frequency
7 NFFT=fsamp/epoch_len;
8
9 % Body of the function
10 x=x-mean(x);
11 [Pxx,f]=pwelch(x,[],[],NFFT,fsamp);
12 fmeanv=sum(f.*Pxx)/sum(Pxx);
13
14 end
```

Listing 2: MDP function's script

```
1 % Hjorth parameters
2 % Activity
3 Act = var(EMG_fil);
```

```

4
5 % Mobility
6 Mob = sqrt(var(diff(EMG_fil))./var(EMG_fil));
7
8 % Complexity
9 Com = sqrt(var(diff(diff(EMG_fil)))./var(EMG_fil));
10
11 % signal parameters
12 fs = 1000; % sample frequency
13
14 % Evaluation of the Power spectrum
15 [pxx, f] = pwelch(EMG_fil, [], [], [], fs);
16
17 % Defining the cut-off frequencies
18 f1 = 10; % Low cut-off frequency (Hz)
19 f2 = 500; % High cut-off frequency (Hz)
20
21 % Finding indices for f1 and f2
22 idx_f1 = find(f >=f1, 1);
23 idx_f2 = find(f <= f2, 1, 'last');
24
25 % Evaluation of the power spectrum of interest (f1 - f2)
26 P_f = sum(pxx(idx_f1:idx_f2));
27
28 % Evaluation of the total Power content (0 - f_max)

```

```

29 P_t = sum(pxx);
30
31 % Spectral Purity Index
32 SPI = P_f / P_t;
33
34 % MEAN ABSOLUTE VALUE
35 MAV = sum(abs(x))/(length(x)/fsamp);
36 % Root Mean Square
37 RMS=std(x)

```

Listing 3: Hijort parameters' script

```

1 % Create feature extractor objects
2 meanFE = signalTimeFeatureExtractor("Mean", true, "SampleRate
   ", fs);
3
4 timeFE = signalTimeFeatureExtractor("RMS", true, ...
5   "StandardDeviation", true, ...
6   "ShapeFactor", true, ...
7   "PeakValue", true, ...
8   "CrestFactor", true, ...
9   "ClearanceFactor", true, ...
10  "ImpulseFactor", true, ...
11  "SampleRate", fs);
12

```

```

13 freqFE = signalFrequencyFeatureExtractor("PeakAmplitude",
      true, ...
14     "PeakLocation", true, ...
15     "MeanFrequency", true, ...
16     "BandPower", true, ...
17     "PowerBandwidth", true, ...
18     "SampleRate", fs);
19
20 % Create datastores
21 meanFeatureDs = arrayDatastore(Data', "IterationDimension",
      2);
22 timeFeatureDs = arrayDatastore(Filtered', "IterationDimension
      ", 2);
23
24 % Extract features using the datastores and feature
      extractors
25 meanFeatureDs = transform(meanFeatureDs, @(x) meanFE.extract(
      x{:}));
26 timeFeatureDs = transform(timeFeatureDs, @(x) timeFE.extract(
      x{:}));
27
28 % Read the features
29 meanFeatures = readall(meanFeatureDs, "UseParallel", true);
30 timeFeatures = readall(timeFeatureDs, "UseParallel", true);
31

```

```

32 % Combine the features
33 features = [meanFeatures timeFeatures];
34
35 % Convert channel labels to categorical
36 Channels = cellstr(Channels);
37 categories = categorical(Channels);
38 disp(categories);
39
40 % Create the feature table
41 featureTable = array2table(features);
42 actioncats = categorical(categories)';
43 % featureTable.ActivityID = ''; % Uncomment and adapt if
    necessary
44
45 % Display the first rows of the table
46 head(featureTable)

```

Listing 4: Extraction of features through Matlab Signal Processing Toolbox and Statistics and Machine Learning Toolbox