2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)

Artifact Abstract: CNNs for Heart Rate Estimation and Human Activity Recognition in Wrist Worn Sensing Applications

Eoin Brophy*†, Willie Muehlhausen*, Alan F. Smeaton*, Tomás E. Ward*
*Insight Centre for Data Analytics, Dublin City University
{eoin.brophy, willie.muehlhausen, tomas.ward,
alan.smeaton}@insight-centre.org
†Infant Research Centre, University College Cork

ARTIFACT USAGE GUIDE

This is a guide on how to obtain and deploy the artifact and the expected results.

ACCESSING THE ARTIFACT

The full code used for our paper is available on GitHub at the following link: *https://github.com/Brophy-E/CNNs_HAR_and_HR*. (Note: PDF links and bookmarks are not available, you are required to copy and paste.)

The code is available as Jupyter Notebooks and is designed to be run on Google Colaboratory in a free GPU enabled environment using a Python 3 runtime. It is also necessary to have a Google Drive account to store your data.

You can clone the whole repository with: *git clone https://github.com/Brophy-E/CNNs_HAR_and_HR.git*

GETTING READY TO RUN

You should execute the artifact in the following order:

1) Create a directory in Google Drive named 'WristSense'
2) Clone the GitHub repository into 'WristSense'
3) Run the notebook *Download_Data.ipynb*
4) Run the notebook *CNN_Recurrent.ipynb*
5) Run the notebook *HAR_Data.ipynb*
6) Run the notebook *Transfer_Learning_HAR.ipynb*

The Dataset

The dataset is downloaded using the notebook *Download_Data.ipynb*. This notebook will also create some folders in your drive for storing the data and saving results from the notebook *CNN_Recurrent.ipynb*. The dataset is available on PhysioNet at the following link: *https://physionet.org/content/wrist/1.0.0/*

Directories

You are required to create one directory manually in Google Drive (See point 1 in GETTING READY TO RUN). The other directories will be handled through running the provided notebooks. The resulting structure of your should be as follows:

1) '/content/drive/My Drive/WristSense/CNNs_and_HAR/'
   - Contains all notebooks
2) '.../CNNs_and_HAR/Data/'
   - Contains downloaded and processed data
3) '.../CNNs_and_HAR/Results/'
   - Contains results from *CNN_Recurrent.ipynb*

UNDERSTANDING THE FILES

The available README.md file describes the purpose of each file in detail. Each .ipynb file provides detailed descriptions on most cells to make it easy to understand and work through the code.

EXPECTED RESULTS

A. CNN_Recurrent.ipynb

When you call the RCNN function you can specify Conv-Pooling params which will affect the outcome of your heart rate error.

Your choice of conv-pooling $filter(cv\_k)$ and $stride(cv\_k)$ sizes will be dependent on seq_len that changes with you downsampling factors $dwns\_factor$. You can set these in the $call\_RCNN()$ function.

Depending on the chosen batch size and CNN hyper-parameters the errors will be between roughly 14 - 25%.

B. HAR_Data.ipynb

This notebook preprocesses the PPG data; it segments, plots and saves the images to be used in *Transfer_Learning_HAR.ipynb*.

C. Transfer_Learning_HAR.ipynb

You should expect to see classification performance of your retrained model in term of accuracy, a confusion matrix and F1-score.

This work is part-funded by Science Foundation Ireland under grant numbers 17/RC-PhD/3482 SFI/12/RC/2289 and SAP SE

978-1-7281-4657-7/20/$31.00 ©2020 IEEE