Machine Learning Project
Predicting Parkinson's Disease with neural networks

Anas Machraoui, Camil Hamdane, Nathan Girard
Section of Life Science Engineering, EPFL, Switzerland

***Abstract*—With videos of subjects performing a robotic task, we aim at predicting whether the latter have Parkinson's Disease. However, we initially had to create a consistent data set which could be further analyzed with regression methods. In this report we will go over the guidelines of the project, and give guidelines as to how the task should be conducted.**

I. Introduction

For this prediction task, we are first provided with videos of both healthy control (hereafter HC) subjects and Parkinson's Disease (hereafter PD) patients. In those videos, the subjects are performing a robotic task so as to induce a feeling of presence (hereafter FoP). From the researches done, an FoP is defined as an illusory own-body perception with well-defined characteristics that is associated with sensorimotor loss and caused by lesions in distinct regions [1].

In this way, we had to do a multiple step pre-processing task in order to clean and retrieve specific features to be given to a neural network. Unfortunately, we have understood that creating a data set for this analysis from raw data was not possible in the short amount of time available before the deadline of this project.

II. Robotic task

The participant moved the master robot via his right index finger (that was attached to the robot), which actuated the movements of the slave robot that applied touches to the participant's back. In order to test the impact of robotically controlled sensorimotor conflicts to induce changes in bodily self-consciousness, we tested the following four experimental conditions:

- **left synchronous : 0ms** between movement and touch on the back by the robot
- **left asynchronous: 500ms** between movement and touch on the back by the
- **right synchronous: 0ms** between movement and touch on the back by the robot
- **right asynchronous: 500ms** between movement and touch on the back by the robot

A "virtual back" in front of participants was created in order to have a mechanical stop (occurring synchronously or asynchronously) to the touch that the participant received on the back.

III. Preprocessing

Because those were just the raw videos of the experiment, we first had to cut the latter into 4 new videos describing four experimental conditions (the videos should be cut from the beginning of the robotic task to the end of it). With deeplabcut, we should retrieve the position of the finger and its torso (precisely the point where the robot applies pressure on the back of the subject) with time; and get a *.csv file*. Then, the all the files of should be merged wisely to create the "raw" version of the dataset. A treatment of the data must be done so that we keep only the features of interest.

With that in hand, we should standardize and remove outliers from the data. Next, we should split the dataset in two. This way we can train a model for predicting Parkinson disease in subjects on one part, and test the accuracy of the latter on the second one.

IV. DeepLabCut

DeepLabCut is a powerful tool that can extract the 3D positions of the different body parts of an individual from a videos [2]. Those positions can be shown in two different manners:

- on the videos itself
- or, in a **.csv file**

With the .csv file, one can extract usefull features instead of simply using the raw data. In this way, it will reduce the dimensionality of the dataset to be analyzed, making it easier to train a model. Here is a list of what features could be intersting to our understanting:

- for each of the 4 robotic tasks:
  - frequency of movements (or speed of movements)
  - Amplitude of movements (horizontal and vertical as two independent features)
- Right/Left-handed

With that there are already **9 features** for each subjects, with a particular care for correlation of features. Indeed, it is of great importance to minimize the correlation between the features of the data.

V. How to use it

A. Installation of the environment for local use

To run properly deeplabcut in local, one should make sure one's laptop is powerfull enough. Because deeplabcut is an image processing algorithm working with neural nets, one's laptop need a good GPU (or CPU).

First, one must install a conda environment to be able to run deeplabcut. Thus one need to install Anaconda and then download (git clone) repository found on the github of deeplabcut. Then, one should go to the folder named conda-environments (using "cd" command), and type depending if GPU or CPU want to be used for computation:

- conda env create -f DLC-CPU.yaml
- conda env create -f DLC-GPU.yaml

One can now use this environment from anywhere on the computer and just enter your environment by running: *conda activate DLC-CP*. For more details on the installation process, one can always follow the guide found on the deeplabcut *github*.

1) Google Colab: Google Colab is a simple tool that allows to use an hosted environment given by Google. Therefore, you won't need a specific hardware to train and compute a models for recognition of the body parts. The creator of deeplabcut, Mathis Alexander, has created a notebook on which you can analyze all your videos (*DLC-Google Colab*).

However, as you run this notebook on the server of google, make sure you croped the videos before analyzing them. It is very simple to run on Google Colab, you simply need to follow the steps indicated in the notebook.

VI. Discussion

For deeplabcut to be able to recognize the different body parts of the subjects, the videos must show the **complete body**. Otherwise, deeplabcut won't be able to recognize a human body, and will place the different nodes, associated to the corresponding body parts, randomly. Hence, the results won't be usable for further analysis.

For better results, the videos should be filmed with the background the most neutral as possible, with no other person appearing in the videos. As you will run a unique human recognition task, the appearance of a different person from the subject in the video, will most likely yield unwanted results (nodes could jump onto the other person, or even disappear for a moment).

Also, for future tasks, each videos should be filmed with the same angle compared with the subject and the same distance, because the distance won't be the same. Even if the data is normalized after (and it must be done because of the difference on height between the subjects), it will add supplementary error that could be avoided easily.

VII. Tools and Libraries

If using Google Colab, there is no supplementary tools or libraries to install. However, if you run deeplabcut in local, you need :

- TensorFlow
- wxpython
- ffmpeg *only for windows computer/laptop*

To install those tools and librarires, you can use **pip install "tool_to_install"** or **brew install "tool_to_install"** in the terminal.

Acknowledgements

We would like to thank Bruno Herbelin and Fosco Bernasconi especially for agreeing our participation on this project even if we did not succeed in our task. Also, we would like to draw your attention on the fact that this report is primarily intended for their laboratory.

References

[1] Blanke O, Pozeg P, Hara M, Heydrich L, Serino A, Yamamoto A, Higuchi T, Salomon R, Seeck M, Landis T, Arzy S, Herbelin B, Bleuler H, Rognini G. *Neurological and robot-controlled induction of an apparition*. Curr Biol. 2014 Nov 17;24(22):2681-6. doi: 10.1016/j.cub.2014.09.049. Epub 2014 Nov 6. PMID: 25447995.

[2] Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge Mackenzie Weygandt Mathis. *Using DeepLabCut for 3D markerless pose estimation across species and behaviors*. Nat Protoc 14, 2152–2176 (2019). https://doi.org/10.1038/s41596-019-0176-0.