1. MDataExtraction.py
2. CameraInfo.ipynb # read camera information
3. ReadCali.ipynb #read calibration data
4. MergeCali.ipynb #merge calibration data and original data

Procedure:

1. ReadCali.ipynb #read calibration data
2. MergeCali.ipynb #merge calibration data and original data
3. CameraInfo.ipynb # read camera information
4. MDataExtraction.py # extract metadata
5. Plot.R # plotting for metadata visualisation

Original data:

../Data/data\_from\_camera/\*.txt :calibration data

data.csv

Camera.csv

Results:

Cali.txt: temporary txt file generated by ReadCali

Cali.csv, Cali\_Simp.csv: **rReadCali**

**MergeCali: input:** Cali\_Simp.csv ; **output:** Merged\_Data.csv

**CameraInfo : input:** Camera.csv Merged\_Data.csv **output:** Merged\_Data.csv

**MDataExtraction: input:** Merged\_Data.csv **output:** New\_Merged\_Data

con\_compression.py #run it when ssh disconnected

CountBodyParts.ipynb # test Json and Kp file

Cp\_Cali.ipynb #get the sample dataset

Cp\_AllPic.py #Cp all pictures into a single folder

CpCali.py #""" Copy all pictures recorded in calibration data set to the specified folder

This script should be placed under the mian folder of all pictures."""

DensityMap.ipynb #plot the density map

Image\_compression.py # compress image in a folder

Json.py #Json class to read json files

Json\_plot.R # time distribution and count distrition plot for json files

Kp.py : class represented skeleton data

NrJson.py : run it when ssh disconnect when running openpose

ReadJ.py.ipynb : confusion matrix for testdataset

ReadJ.py.ipynb : confusion matrix for testdataset

DensityMap: a density map of obserable poses. And time distribution of skeletons between different sites.(0-24) (7-19)

Compare\_classifier: confusion matrix for the openpose and cnn classifier