UT.md 2021/4/26

## **Dataset Introduction-UT**

Note that, using data\_processing\_ut.py, you can get Image and Label folder.

You should make a new folder named ClusterLabel/train or other name, and run python Cluster\_train\_ut.py in the folder. After clustring, you should remove the Cluster\_train\_ut.py from ClusterLabel/train folder.

You also should make a new folder named ClusterLabel/test or other name, and run python Cluster\_test\_ut.py in the folder. After clustring, you should remove the Cluster\_test\_ut.py from ClusterLabel/test folder.

The ClusterLabel/train contains the data of synthese image. You perform the training stage of cross validation using this data. The ClusterLabel/test contains the data of collected image. You perform the test stage of cross validation using this data.

## File Structure

```
UT-Dataset
    |-Label
        -test
        -synth
            |-{subject}.label
    -Image
        |-{subject}
                 |-test
                 -synth
                    |-1.jpg
                    |-2.jpg
    |-ClusterLabel
         |-train
              |-Cluster{n}.label
        -test
             |-Cluster{n}.label
```

## .label File Format

UT.md 2021/4/26

Each .label file in ClusterLabel contains the data of one subject. Each line contains the data of one image. The first line in .label file is the name of contained variables. Variables are separated by space. As for variables contain more than one value. values are separated by .

- Image string Path of normalized eye image relative to ../Image/.
- Origin string Indicate the original image.
- WhichEye string Denote which eye the frame is.
- 3DGaze (3,) Ground truth of normalized 3D gaze direction vector.
- 3DHead (3,) Ground truth of normalized 3D head orientation vector.
- 2DGaze (2,) Ground truth of normalized 2D gaze direction vector *i.e.* yaw and pitch.
- 2DHead (2,) Ground truth of normalized 2D head orientation vector *i.e.* yaw and pitch.

## Geting Start.

You could read the line in .label file for reading image data.

Assuming the root path is /home/UT. You could:

```
import os
import cv2

# line; One line in `.label` file.
imroot = '/home/UT'

image_path = os.path.join(imroot, 'Image', line.split(' ')[0])

image = cv2.imread(image_path)

label = line.strip().split(' ')[3].split(",")
label = np.array(label).astype('float')
```