

Dataset Introduction MPIIFaceGaze

File Structure

```

MPIIFaceGaze
|
|-Label
|   |
|   |-{subject}.label
|
|-Image
|   |
|   |-{subject}
|       |-face
|       |-left
|       |-right
|       |-grid
|           |-1.jpg
|           |-2.jpg
|           |
|           |
|           |
|           .
|           .
|           .

```

.label File Format

Each `.label` file 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 `,`.

- **Face** - *string* - Path of normalized face image relative to `../Image/`.
- **Left** - *string* - Path of normalized left eye image relative to `../Image/`.
- **Right** - *string* - Path of normalized right eye image relative to `../Image/`.
- **grid** - *string* - Path of face grid image relative to `../Image/`.
- **Origin** - *string* - Indicate the original image.
- **WhichEye** - *string* - Denote which eye is chosen in standard MPIIGaze test sets.
- **2DPoint** - `(2,)` - Ground truth of POG in pixel coordinates.
- **HeadRot** - `(2,)` - Ground truth of normalized 2D head orientation vector *i.e.* yaw and pitch.
- **HeadTrans** - `(3,)` - Ground truth of 3D head translation in normalized CCS.
- **ratio** - `(2,)` - Ratio from screen pixel to millimeters.
- **FaceCorner** - `(4,)` - Pixel coordinates of top-left and bottom-right corner of face image.
- **LeftEyeCorner** - `(4,)` - Pixel coordinates of top-left and bottom-right corner of left eye image.
- **RightEyeCorner** - `(4,)` - Pixel coordinates of top-left and bottom-right corner of right eye image.

Getting Start.

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

Assuming the root path is `/home/MPIIFaceGaze`. You could:

```
import os
import cv2

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

face_path = os.path.join(imroot, 'Image', line.split(' ')[0])
left_path = os.path.join(imroot, 'Image', line.split(' ')[1])
right_path = os.path.join(imroot, 'Image', line.split(' ')[2])
grid_path = os.path.join(imroot, 'Image', line.split(' ')[3])

face_image = cv2.imread(face_path)
left_image = cv2.imread(left_path)
right_image = cv2.imread(right_path)
grid_image = cv2.imread(grid_path)

label = line.strip().split(' ')[6].split(",")
label = np.array(label).astype('float')
```