## EigenFaces (PCA) on LFW Dataset

```
# !pip install opencv-python
# !apt update
# !apt install ffmpeg libsm6 libxext6  -y
```

```
from sklearn.datasets import fetch_lfw_people
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import numpy as np
```

```
def plot_gallery(images, h, w, n_row = 5, n_col = 5):
    plt.figure(figsize =(1.8 * n_col, 2.4 * n_row))
    plt.subplots_adjust(bottom = 0, left =.01, right =.99, top =.90,
hspace =.35)
    for i in range(n_row * n_col):
        plt.subplot(n_row, n_col, i + 1)
        plt.imshow(images[i].reshape((h, w)), cmap = plt.cm.gray)
        plt.xticks(())
        plt.yticks(())
```

```
lfw_dataset = fetch_lfw_people(min_faces_per_person=100, resize=0.4)
_, h, w = lfw_dataset.images.shape
X = lfw_dataset.data
print(X.shape)
plot_gallery(X,h,w)
```

```
(1140, 1850)
```

```
n_components = 25
pca = PCA(n_components=n_components, whiten=True).fit(X)
plot_gallery(pca.components_,h,w)
```

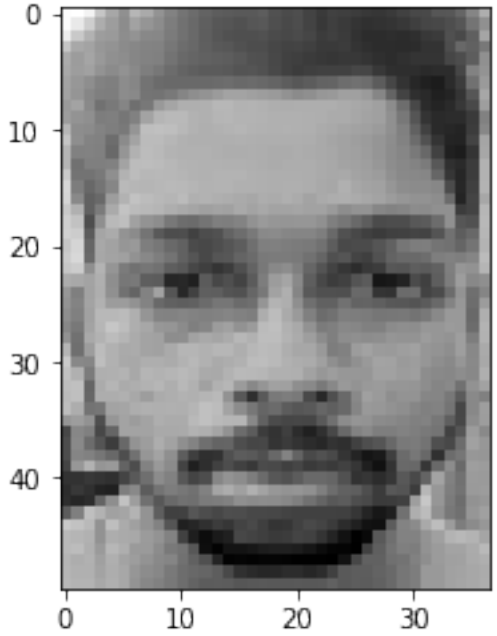
## EigenFaces on My Face

```
import os
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
im = cv2.imread('test2.jpg',0)

plt.imshow(im,cmap='gray')
print(im.shape)
print(im.flatten().shape)
```

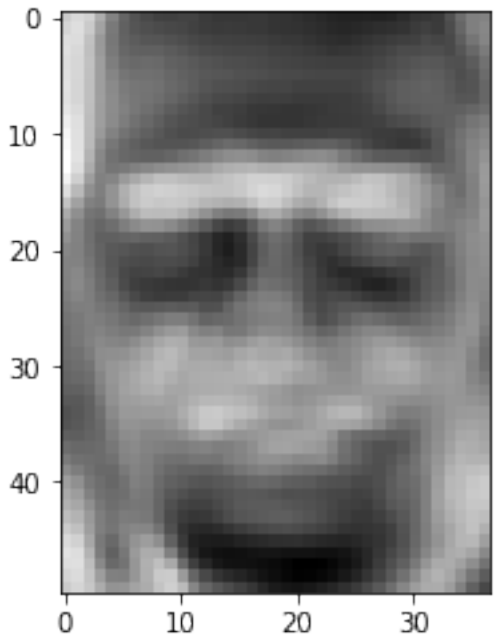
```
(50, 37)
(1850,)
```




```
im_transform = pca.transform(im.flatten().reshape((1,1850)))
print(im_transform.shape)
im_recons = np.matmul(pca.components_.T,test.T)
im_recons = im_recons.reshape(50,37)
print(im_recons.shape)
plt.imshow(im_recons, cmap='gray')
```

```
(1, 25)
(50, 37)

<matplotlib.image.AxesImage at 0x7f2cf01515e0>
```




## PCA on Celebrity Faces

```
#!wget http://conradsanderson.id.au/lfwcrop/lfwcrop_grey.zip
#!unzip lfwcrop_grey.zip
```

```
from sklearn.decomposition import PCA
from matplotlib.pyplot import imread
import glob
import numpy as np
```

```
images = [imread(path) for path in
glob.glob("lfwcrop_grey/faces/*.pgm")]
images = np.asarray(images[1:1001])
n_samples, h, w = images.shape

images = images.reshape(n_samples, h*w)

print(images.shape)

# pca=PCA(n_components=25, whiten=True)
# pca.fit(images)
```

```
(1000, 4096)
```

```
plot_gallery(images, h, w)
```

```
plot_gallery(pca.components_,h,w)
```

```
pca.components_.shape
```

```
reconstruct = pca.inverse_transform(pca.fit_transform(images))
plot_gallery(reconstruct, h, w)
```