COMPUTER VISION PROJECT 2
rus209

a) An MS Words file that contains your source code (with full comments and documentation), the language and compiler used, instructions on how to compile and run your program, and the following:
Source Code:

Ans: Python code can be run using any standard python interpreter

```
# coding: utf-8
```

```
from matplotlib import pyplot as plt
import numpy as np
from scipy.ndimage import imread
from scipy.spatial.distance import euclidean as dist
```

```
#Importing Training Images
img1n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject01.normal.jpg')
img2n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject02.normal.jpg')
img3n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject03.normal.jpg')
img7n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject07.normal.jpg')
img10n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject10.normal.jpg')
img11n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject11.normal.jpg')
img14n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject14.normal.jpg')
img15n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject15.normal.jpg')
```

```
#Importing Testing Images
img1c = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject01.centerlight.jpg')
img1h = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject01.happy.jpg')
img7c = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject07.centerlight.jpg')
img7h = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject07.happy.jpg')
img11c = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject11.centerlight.jpg')
img11h = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject11.happy.jpg')
img12n = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject12.normal.jpg')
img14h = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject14.happy.jpg')
img14s = imread('/Users/rohitsuvarna/NYU/Comp Vision/CVProject2/
Face dataset/subject14.sad.jpg')
img_apple = imread('/Users/rohitsuvarna/NYU/Comp Vision/
CVProject2/Face dataset/apple1_gray.jpg')
```

```
#Stacking rows together
train_image_list =
[img1n,img2n,img3n,img7n,img10n,img11n,img14n,img15n]
col_vecs = [img.flatten() for img in train_image_list]
col_vecs
```

```
#Average Face
m = np.average(col_vecs,axis=0)
m = np.average(col_vecs,axis=0)
mean_face = m.reshape((231,195))
plt.imshow(mean_face,plt.cm.gray)
```

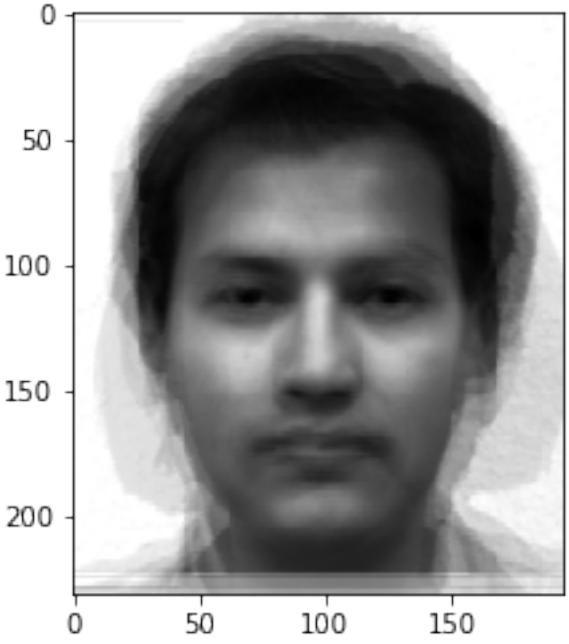
```
#Subtracting the average face
col_vecs_reduced = [(vec - m) for vec in col_vecs ]
```

```
#Computing matrix A
A = np.vstack(col_vecs_reduced)
A = A.T


#Covariance matrix C
C = np.dot(A,A.T)
L = np.dot(A.T,A)
Eig_v = np.linalg.eig(L)
Eig_vals = Eig_v[0]
Eig_vecs = Eig_v[1]
V = Eig_vecs
Eig_vals


#Finding Eigen Space U
U = np.dot(A,V)




#Projecting Training Faces on Face Space
Omega_training = [np.dot(U.T,train_vec) for train_vec in
col_vecs_reduced]



#Face Recognition
comp_image_list =
[img1n,img2n,img3n,img7n,img10n,img11n,img14n,img15n,img1c,img1h
,img7c,img7h,img11c,img11h,img12n,img14h,img14s,img_apple]
Img_dict = {0:1,1:2,2:3,3:7,4:10,5:11,6:14,7:15}
Actual = [1,2,3,7,10,11,14,15,1,1,7,7,11,11,12,14,14,0]
Predicted = []


#Thresholds
T_0 = 7000000000000
T_1 = 140000000
```

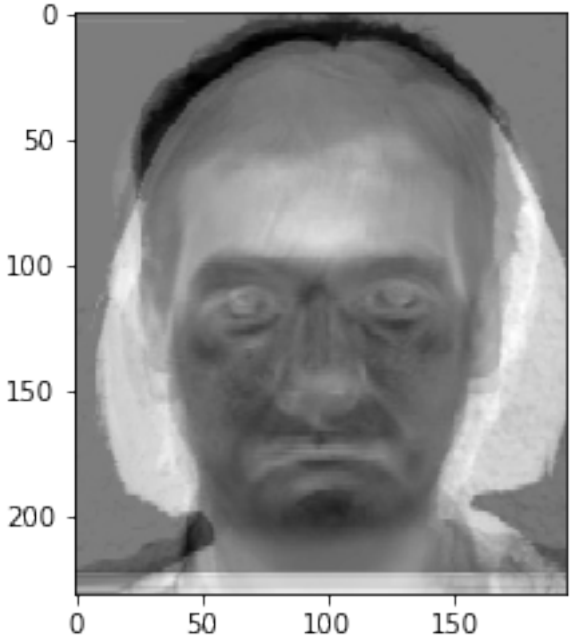
```
def face_detect(img,U,Omega_training,T_0,T_1):
    '''
    Returns the predicted subject if recognised
    else returns 0 for a non-face, -1 for unknown face
    '''
    I   = img.flatten()
    I = I - m
    Omega_I = np.dot(U.T,I)
    I_R = np.dot(U,Omega_I)
    d_0 = dist(I_R,I)
    print('d_o is %d' % d_0)
    dist_array = [dist(Omega_I,Omega) for Omega in
Omega_training]
    res = min(dist_array)
    print('d_1 is %d' %res)
    index = Img_dict[dist_array.index(res)]
    if d_0 > T_0:
        return 0
    else:
        if res > T_1:
            return -1
        return index



Predicted = [face_detect(img,U,Omega_training,T_0,T_1) for img
in comp_image_list]



def print_results(act,pred):
    total = len(act)
    correct = 0
    for i in range(len(act)):
        print("Subject %d is identified as Subject %d"%
(act[i],pred[i]))
        if act[i] == pred[i]:
            correct += 1
    print("Got %d right out of %d" % (correct,total))



#Predictions
print_results(Actual,Predicted)
```

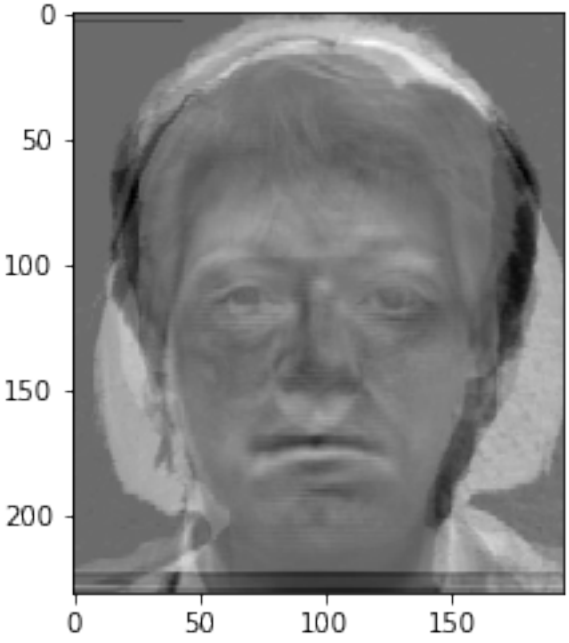
```
#Eigenfaces:

for i in range(8):
    eigen_face = U[:,i]
    eigen_face = eigen_face.reshape((231,195))
    fig, ax = plt.subplots(1,1)
    im2 = ax.imshow(eigen_face,plt.cm.gray)
    plt.show()



#Other output for test images:

test_image_list =
[img1c,img1h,img7c,img7h,img11c,img11h,img12n,img14h,img14s,img_
apple]

 #Helper function to print output
def output_func(img,U,Omega_training,T_0,T_1):
    sh = img.shape
    I   = img.flatten()
    I = I - m
    I_res = I.reshape(sh)
    fig, (ax1, ax2) = plt.subplots(1,2)
    ax1.set_title('I - m face')
    im1 = ax1.imshow(I_res,plt.cm.gray)
    Omega_I = np.dot(U.T,I)
    print('PCA coefficients are \n')
    print(Omega_I)
    I_R = np.dot(U,Omega_I)
    Ir_res = I_R.reshape(sh)
    ax2.set_title('Reconstructed face I_r')
    im2 = ax2.imshow(Ir_res,plt.cm.gray)
    dist_array = [dist(Omega_I,Omega) for Omega in
Omega_training]
    print('The distances are \n')
    print(dist_array)
    plt.show()
    print('\n\n\n\n\n\n')


for img in test_image_list:
    print('For %r we have the following results \n\n' %img)
    output_func(img,U,Omega_training,T_0,T_1)
```

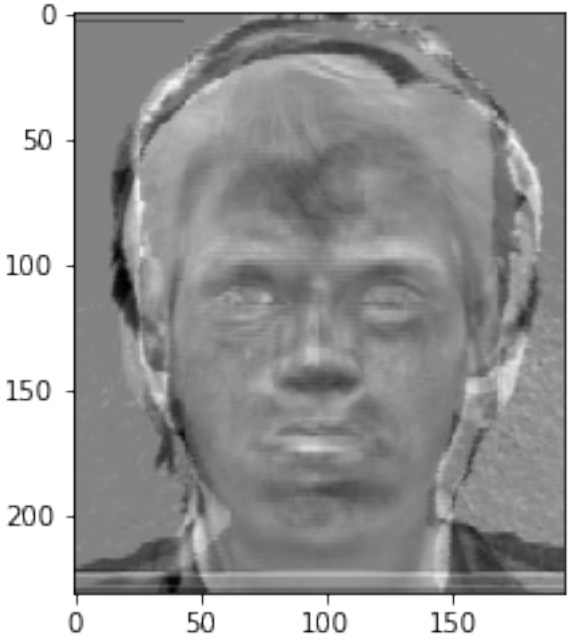
Mean Face:

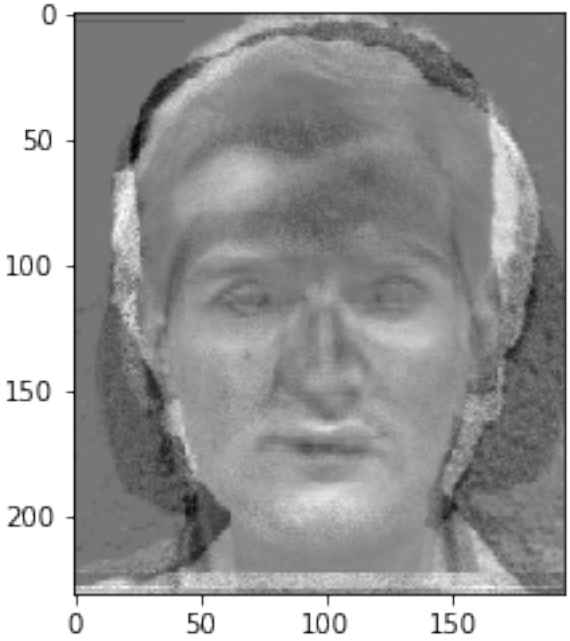


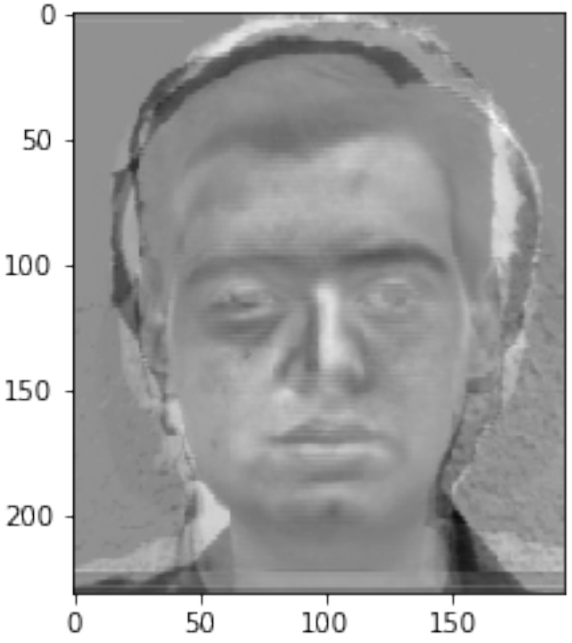
Eigen Faces:
In order

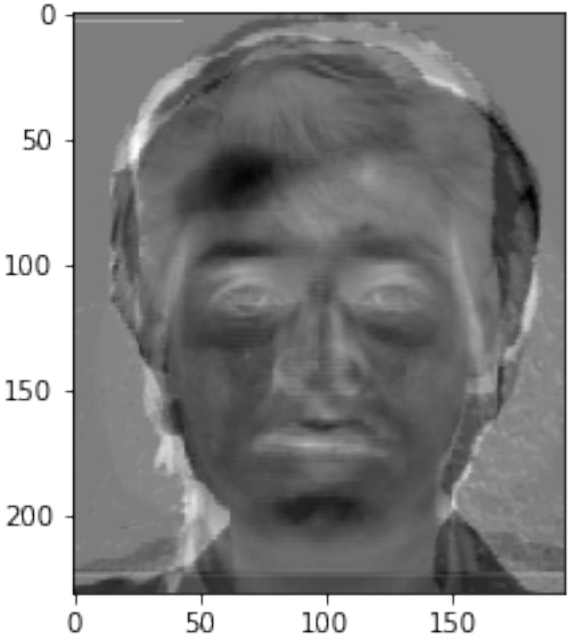


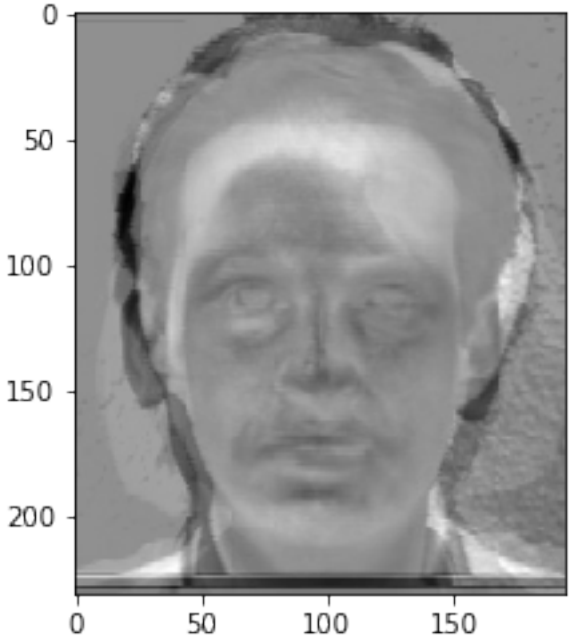


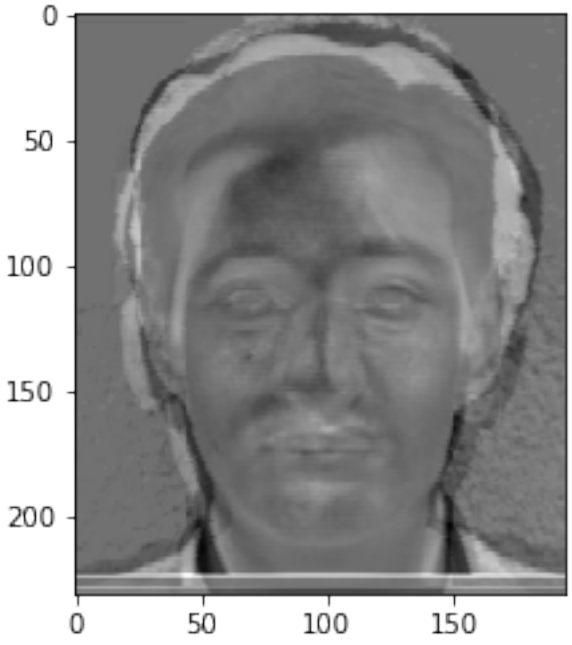

0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150

(2) The PCA coefficients (Ωi) for each training image:

Ans:

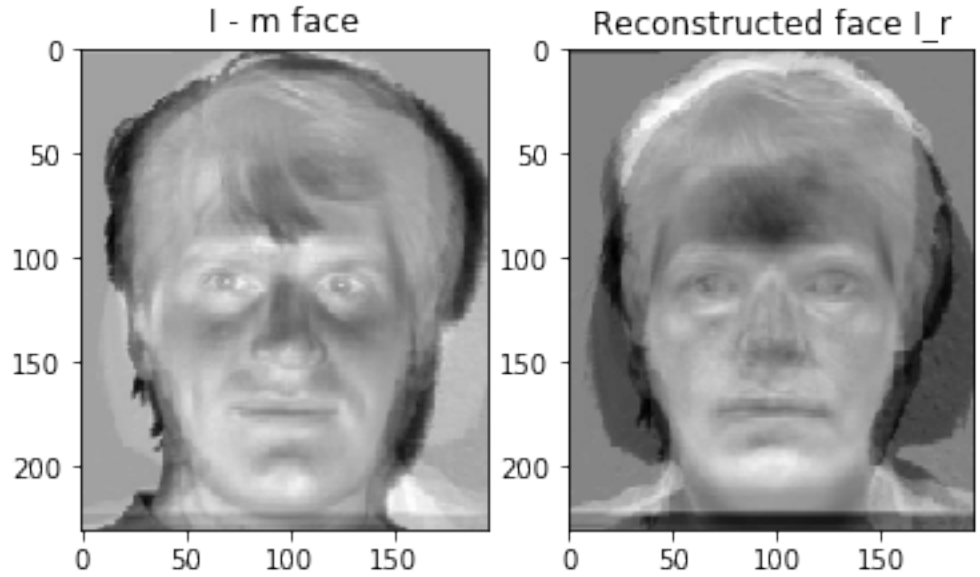
```
[array([  7.45283569e+07,   3.00355322e+08,   6.08775242e-08,
         -1.86486103e+08,   2.43272892e+07,  -2.81183673e+07,
         -4.79970158e+06,   1.99954727e+07]),
 array([  2.35959144e+08,   1.48419206e+08,   2.23977520e-08,
         -1.12998840e+08,  -6.30304611e+06,   2.37321739e+07,
         -1.49498856e+07,  -1.05723954e+07]),
 array([  2.18277162e+08,   2.30305748e+08,   4.30747679e-08,
         -1.34152903e+08,  -5.02106649e+06,  -1.10471303e+08,
          1.20529290e+07,  -2.82610338e+07]),
 array([  3.09733059e+08,   1.41798513e+08,   2.37310788e-08,
         -7.67160288e+07,   2.79303377e+07,  -7.89909114e+07,
         -2.24966723e+07,   3.00546705e+07]),
 array([  1.24666594e+08,   2.54799768e+08,   8.24925081e-09,
         -3.62157505e+07,   1.53539668e+07,  -2.76577806e+07,
          5.65848466e+07,   6.64733438e+06]),
 array([ -1.59597833e+08,   1.00126907e+08,   5.66612114e-08,
         -1.20188087e+08,   4.33176243e+06,  -6.28407792e+07,
          1.26881806e+07,   1.57079311e+07]),
 array([  9.55641073e+07,   2.77313854e+08,   3.29310945e-08,
         -7.21654833e+07,  -2.12127268e+07,  -5.89966615e+07,
         -2.12443601e+07,   4.18677846e+07]),
 array([  3.03810810e+08,   1.60449738e+08,   5.82784758e-08,
         -1.64718867e+08,  -7.50063905e+06,  -4.88507975e+07,
          5.15186459e+07,   4.48709067e+07])]
```

(3) For each test image: the image after subtracting the mean face (I),
its PCA coefficients (ΩI), the reconstructed face image (IR), distances di for $i = 0\ to\ M$,
and classification result (non-face, unknown face, or identify of face.)

Ans:

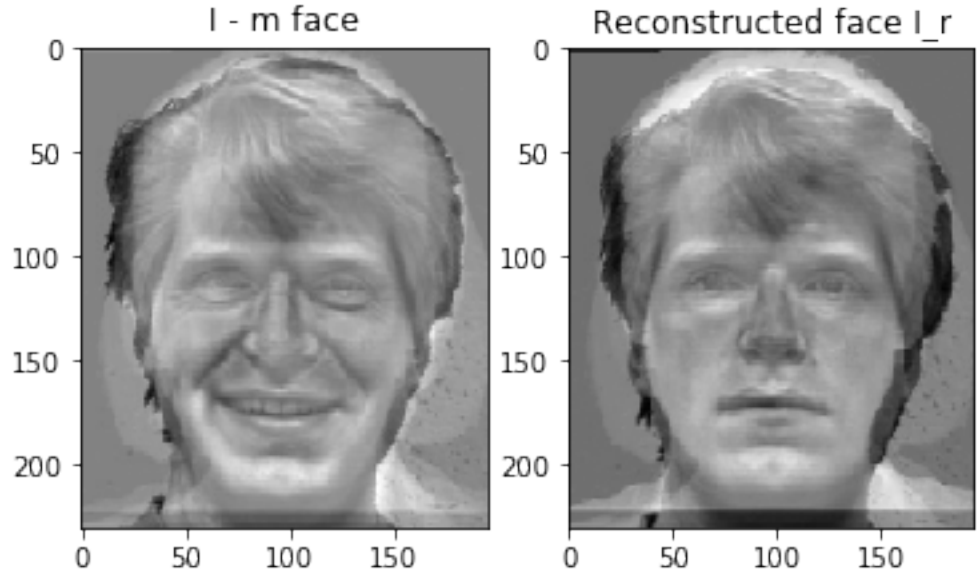
1 Centerlight

```
PCA coefficients are

[ -4.19265944e+07   9.33435765e+06   4.46553506e-09
-1.96894052e+07
   3.51175100e+06   2.33364673e+07  -4.53879945e+06
-2.45326164e+06]
The distances are

[111586822.41848011, 155242389.34017056,
146676511.3660945, 229591657.82398075,
119707328.1334251, 292870266.5065856,
107945627.81560327, 214023418.31487235]
```



1 happy

```
PCA coefficients are

[ -2.84226819e+07   6.21657618e+07   1.44756247e-08
-5.58402336e+07
   1.39551196e+07   2.39919521e+07  -1.56038461e+07
-4.88745009e+05]
The distances are

[63067848.62498437, 181936137.17792708,
147059539.1608555, 249271519.0750657,
147572653.64382246, 332324142.1820322,
117694790.26319712, 222118699.9686467]
```



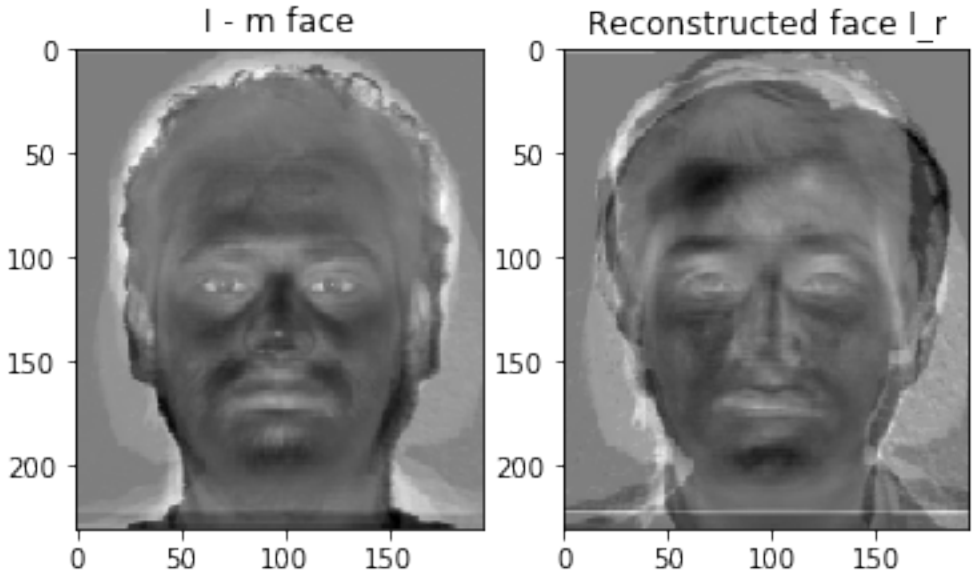
7 Centerlight

PCA coefficients are

```
[  5.94280720e+06   1.09073119e+07  -2.00761307e-08
6.44391875e+07
  -4.06118038e+06  -1.88646091e+06  -1.60012838e+07
1.25351205e+07]
```

The distances are

```
[186019216.48344228, 147462236.8534809,
136206359.70409366, 176510231.03979555,
90747702.06187524, 343932044.6814206,
96679844.91103761, 204454021.45323846]
```



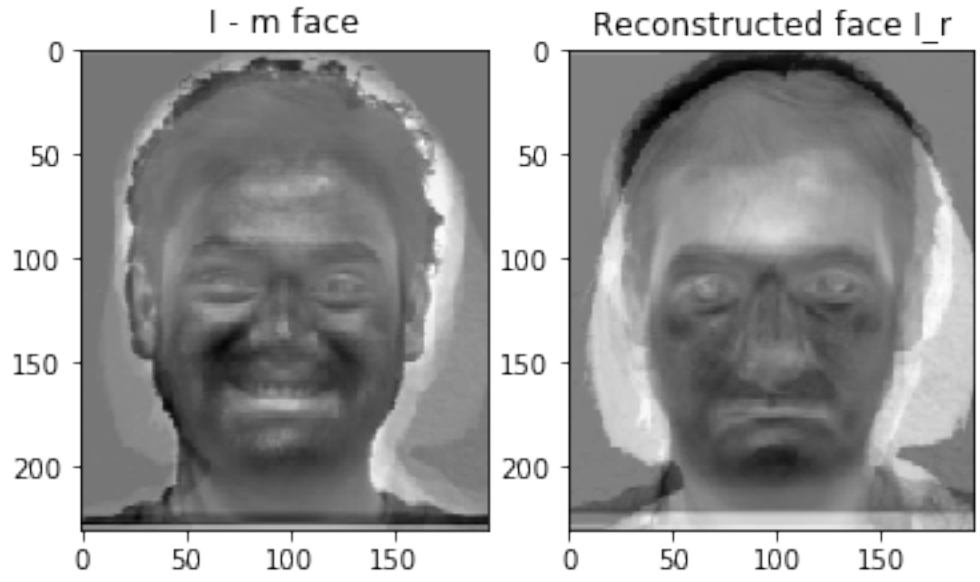
7 Happy

PCA coefficients are

```
[  1.03893380e+08  -9.15597753e+06  -1.07347312e-08
2.31395182e+07
   9.78874352e+05  -3.10262255e+06  -4.00701970e+06
6.01476791e+06]
```

The distances are

```
[233055760.2403581, 100426750.6450444,
103330781.053852, 88831686.49484487,
164784911.90361091, 425382239.23979646,
185795025.3343768, 109655903.29353152]
```

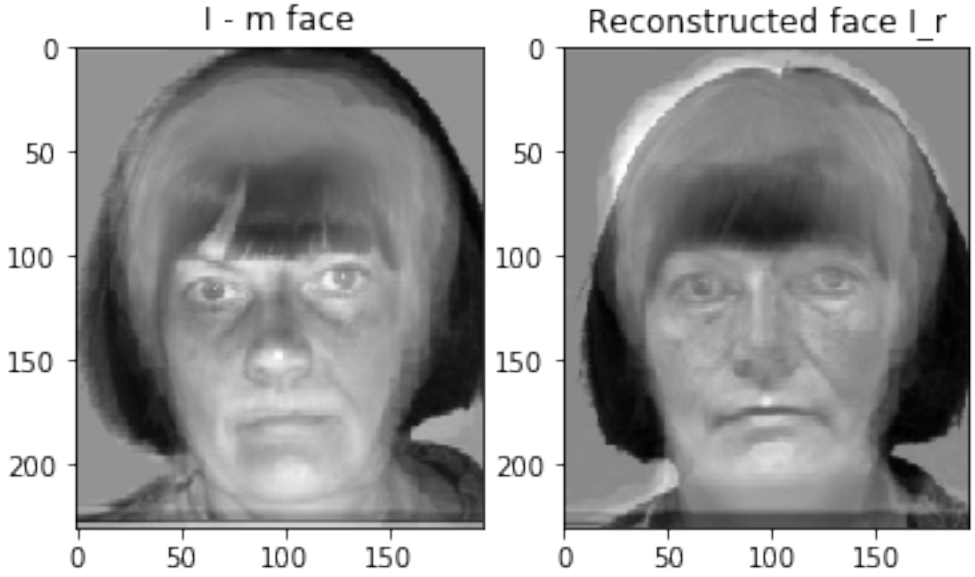


11 Centerlight

PCA coefficients are

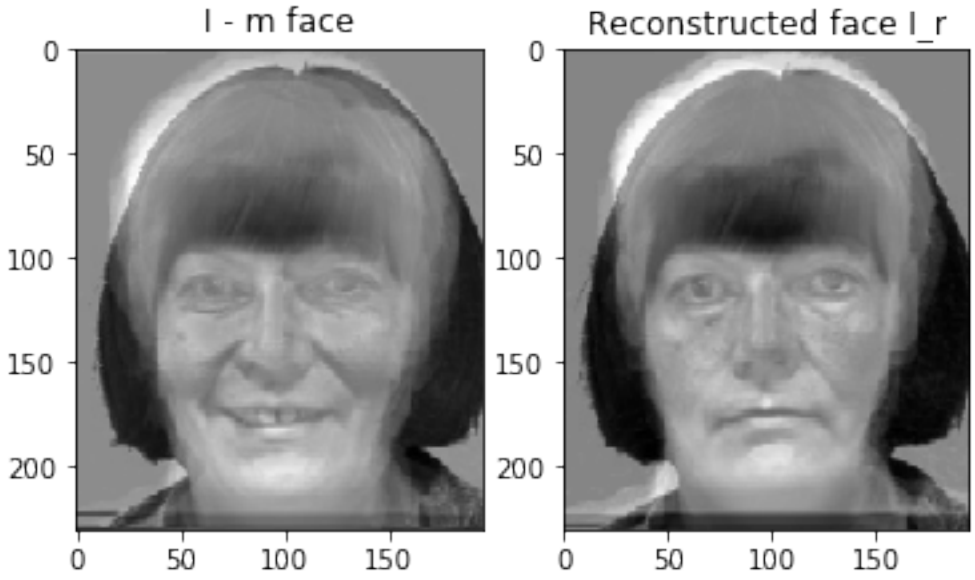
[-1.84458324e+08 -1.27514264e+08   9.22289043e-09
-3.94098492e+06
  -4.82546087e+06   6.57524499e+06 -1.18073793e+07
-4.07760603e+06]
The distances are

[262141694.81047183, 288905361.76333904,
307856548.3060301, 356796012.7954678,
261538544.37455803, 130964036.02232766,
249090014.3271997, 357951333.85845053]



11 Happy

```
PCA coefficients are

[ -2.91280480e+08  -1.07981032e+08   1.76253199e-08
-8.07984970e+06
  -3.16856693e+06  -8.84253231e+06  -2.77433359e+06
1.23549801e+06]
The distances are

[308163446.49942905, 391093302.1635501,
390729034.1051311, 457750024.72404075,
327812434.4398219, 21783094.363421325,
306425098.0061767, 455182031.45493424]
```

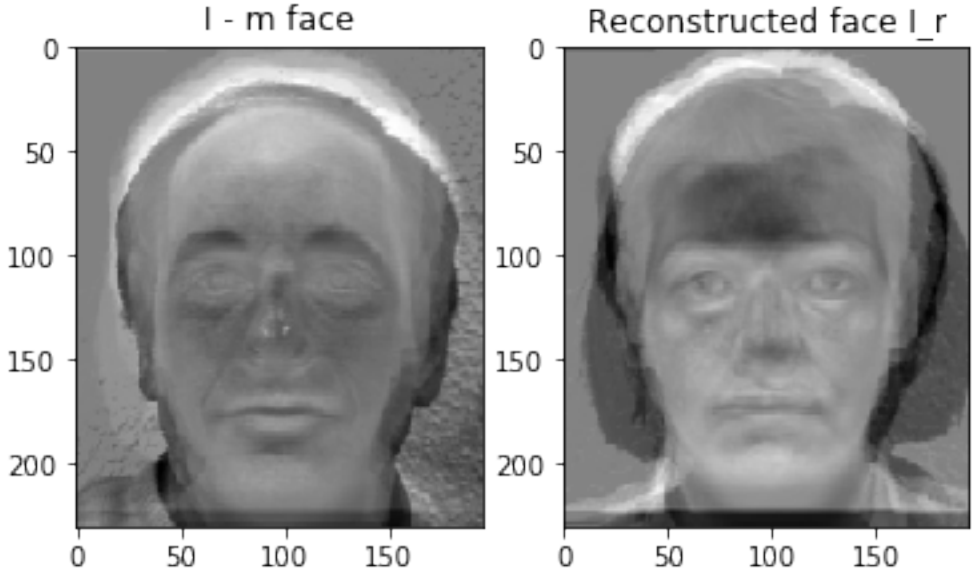


12 Normal

PCA coefficients are

[-8.48151488e+07 2.39983394e+07 -2.25923212e-09
9.24693940e+06
  1.09479841e+06 3.03539436e+07 -1.06420075e+07
1.14064585e+07]
The distances are

[114088420.68360247, 196373441.62815523,
189715468.36281228, 268302208.3295624,
113596022.11108254, 262703834.78576162,
86528212.38591179, 262709370.18400285]



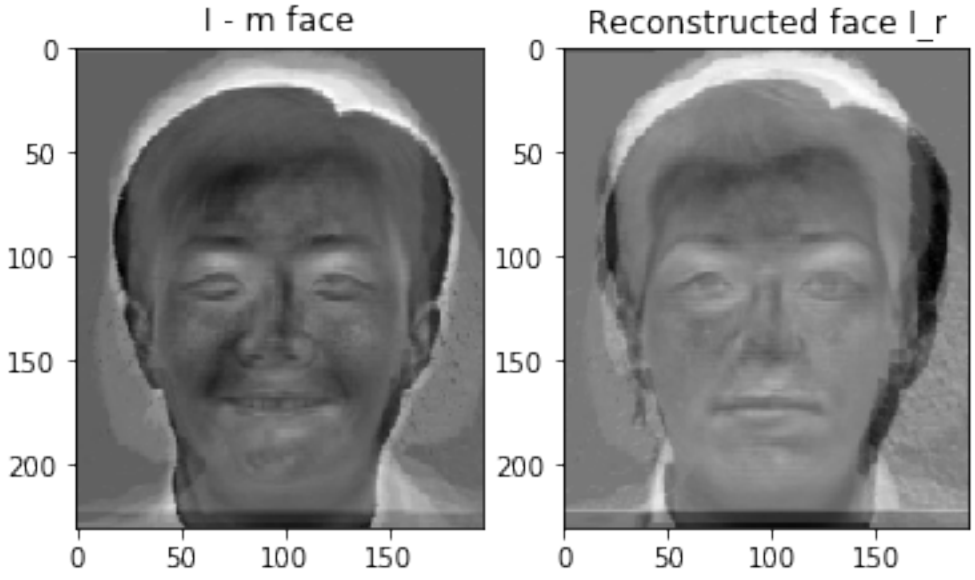
14 Happy

PCA coefficients are

```
[ -3.72003742e+07   7.23552596e+07  -1.41108835e-09
2.01072900e+07
  -1.10391535e+07  -1.16749219e+07  -3.12504845e+07
1.93984877e+07]
```

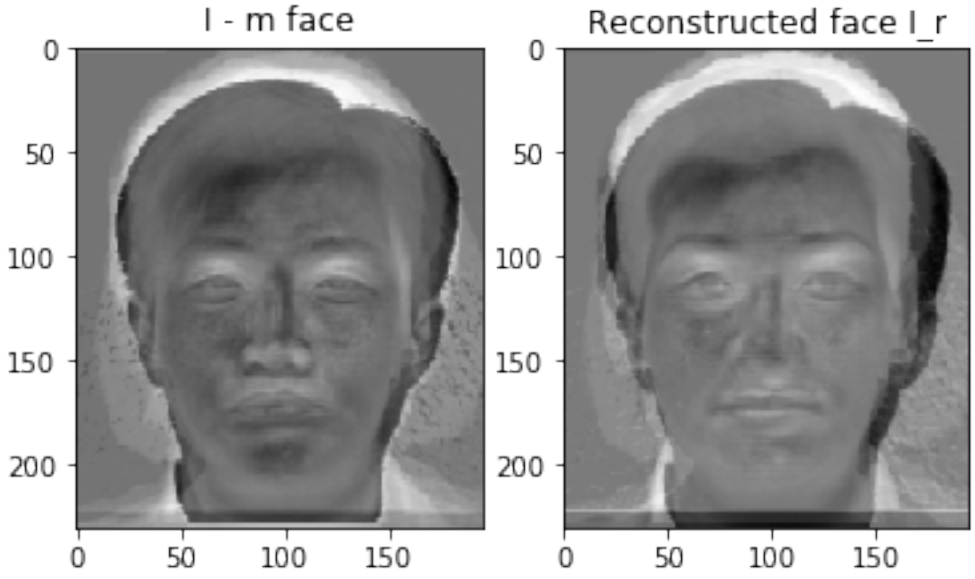
The distances are

```
[116291957.18283445, 201198094.2003047,
149264366.4588196, 240762641.3781882,
111100562.01474124, 327302460.0456842,
31760601.24447625, 245267097.18086782]
```



14 SAD

```
PCA coefficients are

[ -3.09557722e+07   6.21419290e+07  -8.93708875e-09
4.11282105e+07
  -1.89530184e+07  -9.12193903e+06  -3.02355547e+07
1.57609187e+07]
The distances are

[139061618.92516568, 192702933.1697089,
149125741.22847518, 231118169.49102747,
99585049.07145986, 329821305.0914483,
30212478.883972634, 242904340.5481909]
```



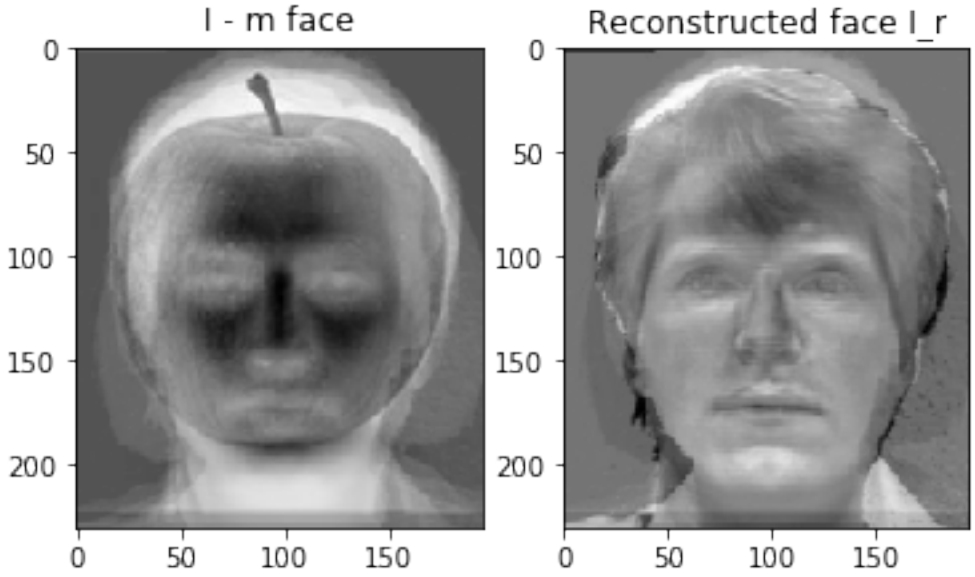
APPLE

PCA coefficients are

```
[ -2.40683176e+07   2.33966524e+07   1.91194543e-08
-5.18978138e+07
   1.54361939e+07  -2.59415798e+06  -3.62456870e+07
1.56605404e+07]
```

The distances are

```
[100111536.8448275, 169692008.72081405,
136153029.87689295, 221814828.11369932,
160265519.12945694, 318666513.7422043,
118883680.68799105, 207071343.06062484]
```



Results

```
Subject 1 center light is identified as Subject 14
Subject 1 happy is identified as Subject 1
Subject 7 center light is identified as Subject 10
Subject 7 happy is identified as Subject 7
Subject 11 center light is identified as Subject 11
Subject 11 happy is identified as Subject 11
Subject -1(12) Normal is identified as Subject 14
Subject 14 Happy is identified as Subject 14
Subject 14 Sad is identified as Subject 14
Subject 0(Apple) is identified as Subject 1
```