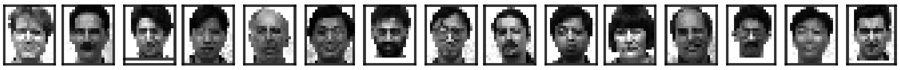
PRO1: Face Recognition and Detection

吴瑞林 电计 1701 201785072

(1) Data: project1-data-Recognition.zip. This is the normalized Yale face database, containing the manually cropped and aligned images of the original Yale face database (credit goes to Prof. Deb Roy, MIT Media Lab). This dataset has 165 images of 15 subjects (subject01, subject02, etc.). There are 11 images per subject, one for each of the following facial expressions or configurations: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. **Let N be the number of images randomly selected from each subject (N < 11). For each experiment, take 15N images from the database as training samples, and the rest as testing samples.**



(2) Let N = 3. Implement Eigenfaces algorithm to the training samples, and then perform face recognition to the testing samples using nearest neighbor classifier. Plot the average misclassification rate vs. the dimensions of Eigenspace (e.g. the dimension K = 10, 20, 30, ...).

K=10

```
N: 3 K: 10 nn_K: 1 m_rate:0.1750
N: 3 K: 10 nn_K: 1 m_rate:0.2417
N: 3 K: 10 nn_K: 1 m_rate:0.2250
N: 3 K: 10 nn_K: 1 m_rate:0.1583
N: 3 K: 10 nn_K: 1 m_rate:0.2083
N: 3 K: 10 nn_K: 1 m_rate:0.1833
N: 3 K: 10 nn_K: 1 m_rate:0.2417
N: 3 K: 10 nn_K: 1 m_rate:0.2250
N: 3 K: 10 nn_K: 1 m_rate:0.1833
N: 3 K: 10 nn_K: 1 m_rate:0.2000
average miss of 10 times: 0.2042
```

K=20

```
N: 3 K: 20 nn_K: 1 m_rate:0.1417
N: 3 K: 20 nn_K: 1 m_rate:0.1917
N: 3 K: 20 nn_K: 1 m_rate:0.1667
N: 3 K: 20 nn_K: 1 m_rate:0.1500
N: 3 K: 20 nn_K: 1 m_rate:0.1167
N: 3 K: 20 nn_K: 1 m_rate:0.1167
N: 3 K: 20 nn_K: 1 m_rate:0.1750
N: 3 K: 20 nn_K: 1 m_rate:0.2167
N: 3 K: 20 nn_K: 1 m_rate:0.2417
N: 3 K: 20 nn_K: 1 m_rate:0.1917
average miss of 10 times: 0.1708
```

K=30

```
N: 3 K: 30 nn_K: 1 m_rate:0.1667
N: 3 K: 30 nn_K: 1 m_rate:0.1917
N: 3 K: 30 nn_K: 1 m_rate:0.2000
N: 3 K: 30 nn_K: 1 m_rate:0.1167
N: 3 K: 30 nn_K: 1 m_rate:0.1500
N: 3 K: 30 nn_K: 1 m_rate:0.1667
N: 3 K: 30 nn_K: 1 m_rate:0.1167
N: 3 K: 30 nn_K: 1 m_rate:0.1500
N: 3 K: 30 nn_K: 1 m_rate:0.1917
N: 3 K: 30 nn_K: 1 m_rate:0.1750
average miss of 10 times: 0.1625
```

K=40

```
N: 3 K: 40 nn_K: 1 m_rate:0.1583
N: 3 K: 40 nn_K: 1 m_rate:0.1417
N: 3 K: 40 nn_K: 1 m_rate:0.1500
N: 3 K: 40 nn_K: 1 m_rate:0.1583
N: 3 K: 40 nn_K: 1 m_rate:0.1417
N: 3 K: 40 nn_K: 1 m_rate:0.1417
N: 3 K: 40 nn_K: 1 m_rate:0.1917
N: 3 K: 40 nn_K: 1 m_rate:0.2000
N: 3 K: 40 nn_K: 1 m_rate:0.1833
N: 3 K: 40 nn_K: 1 m_rate:0.1750
average miss of 10 times: 0.1642
```

K=50

```
N: 3 K: 50 nn_K: 1 m_rate:0.1750
N: 3 K: 50 nn_K: 1 m_rate:0.1417
N: 3 K: 50 nn_K: 1 m_rate:0.1000
N: 3 K: 50 nn_K: 1 m_rate:0.2250
N: 3 K: 50 nn_K: 1 m_rate:0.1667
N: 3 K: 50 nn_K: 1 m_rate:0.1417
N: 3 K: 50 nn_K: 1 m_rate:0.1583
N: 3 K: 50 nn_K: 1 m_rate:0.1833
N: 3 K: 50 nn_K: 1 m_rate:0.1583
N: 3 K: 50 nn_K: 1 m_rate:0.0917
average miss of 10 times: 0.1542
```

K=60

```
N: 3 K: 60 nn_K: 1 m_rate:0.1167
N: 3 K: 60 nn_K: 1 m_rate:0.2083
N: 3 K: 60 nn_K: 1 m_rate:0.1417
N: 3 K: 60 nn_K: 1 m_rate:0.1417
N: 3 K: 60 nn_K: 1 m_rate:0.1500
N: 3 K: 60 nn_K: 1 m_rate:0.1500
N: 3 K: 60 nn_K: 1 m_rate:0.1750
N: 3 K: 60 nn_K: 1 m_rate:0.1417
N: 3 K: 60 nn_K: 1 m_rate:0.1917
N: 3 K: 60 nn_K: 1 m_rate:0.1917
average miss of 10 times: 0.1608
```

K=70

K=80

```
N: 3 K: 70 nn_K: 1 m_rate:0.2000
N: 3 K: 70 nn_K: 1 m_rate:0.1667
N: 3 K: 70 nn_K: 1 m_rate:0.1750
N: 3 K: 70 nn_K: 1 m_rate:0.2250
N: 3 K: 70 nn_K: 1 m_rate:0.2000
N: 3 K: 70 nn_K: 1 m_rate:0.2000
N: 3 K: 70 nn_K: 1 m_rate:0.1833
N: 3 K: 70 nn_K: 1 m_rate:0.1167
N: 3 K: 70 nn_K: 1 m_rate:0.1750
N: 3 K: 70 nn_K: 1 m_rate:0.1583
average miss of 10 times: 0.18
```

```
N: 3 K: 80 nn_K: 1 m_rate:0.1250
N: 3 K: 80 nn_K: 1 m_rate:0.1500
N: 3 K: 80 nn_K: 1 m_rate:0.1333
N: 3 K: 80 nn_K: 1 m_rate:0.1750
N: 3 K: 80 nn_K: 1 m_rate:0.1750
N: 3 K: 80 nn_K: 1 m_rate:0.1917
N: 3 K: 80 nn_K: 1 m_rate:0.1750
N: 3 K: 80 nn_K: 1 m_rate:0.1583
N: 3 K: 80 nn_K: 1 m_rate:0.1833
N: 3 K: 80 nn_K: 1 m_rate:0.1583
average miss of 10 times: 0.1625
```

(3) Let N = 5 and 7, respectively. Repeat (2) for each N.

When N=5:

K=10

```
N: 5 K: 10 nn_K: 1 m_rate:0.1222
N: 5 K: 10 nn_K: 1 m_rate:0.0889
N: 5 K: 10 nn_K: 1 m_rate:0.1778
N: 5 K: 10 nn_K: 1 m_rate:0.1889
N: 5 K: 10 nn_K: 1 m_rate:0.1333
N: 5 K: 10 nn_K: 1 m_rate:0.1111
N: 5 K: 10 nn_K: 1 m_rate:0.1000
N: 5 K: 10 nn_K: 1 m_rate:0.1333
N: 5 K: 10 nn_K: 1 m_rate:0.1667
N: 5 K: 10 nn_K: 1 m_rate:0.1778
average miss of 10 times: 0.14
```

K=20

```
N: 5 K: 20 nn_K: 1 m_rate:0.1778
N: 5 K: 20 nn_K: 1 m_rate:0.0444
N: 5 K: 20 nn_K: 1 m_rate:0.0556
N: 5 K: 20 nn_K: 1 m_rate:0.1889
N: 5 K: 20 nn_K: 1 m_rate:0.1222
N: 5 K: 20 nn_K: 1 m_rate:0.1444
N: 5 K: 20 nn_K: 1 m_rate:0.1000
N: 5 K: 20 nn_K: 1 m_rate:0.1444
N: 5 K: 20 nn_K: 1 m_rate:0.1889
N: 5 K: 20 nn_K: 1 m_rate:0.0889
average miss of 10 times: 0.1256
```

K=30

```
N: 5 K: 30 nn_K: 1 m_rate:0.2000
N: 5 K: 30 nn_K: 1 m_rate:0.1444
N: 5 K: 30 nn_K: 1 m_rate:0.1222
N: 5 K: 30 nn_K: 1 m_rate:0.1222
N: 5 K: 30 nn_K: 1 m_rate:0.0556
N: 5 K: 30 nn_K: 1 m_rate:0.0444
N: 5 K: 30 nn_K: 1 m_rate:0.1556
N: 5 K: 30 nn_K: 1 m_rate:0.1444
N: 5 K: 30 nn_K: 1 m_rate:0.1111
N: 5 K: 30 nn_K: 1 m_rate:0.1111
average miss of 10 times: 0.1211
```

K=40

```
N: 5 K: 40 nn_K: 1 m_rate:0.1556
N: 5 K: 40 nn_K: 1 m_rate:0.1222
N: 5 K: 40 nn_K: 1 m_rate:0.0444
N: 5 K: 40 nn_K: 1 m_rate:0.1889
N: 5 K: 40 nn_K: 1 m_rate:0.1111
N: 5 K: 40 nn_K: 1 m_rate:0.1000
N: 5 K: 40 nn_K: 1 m_rate:0.2000
N: 5 K: 40 nn_K: 1 m_rate:0.1111
N: 5 K: 40 nn_K: 1 m_rate:0.1111
N: 5 K: 40 nn_K: 1 m_rate:0.1333
average miss of 10 times: 0.1278
```

K=50

```
N: 5 K: 50 nn_K: 1 m_rate:0.2111
N: 5 K: 50 nn_K: 1 m_rate:0.1333
N: 5 K: 50 nn_K: 1 m_rate:0.1333
N: 5 K: 50 nn_K: 1 m_rate:0.2333
N: 5 K: 50 nn_K: 1 m_rate:0.1111
N: 5 K: 50 nn_K: 1 m_rate:0.2222
N: 5 K: 50 nn_K: 1 m_rate:0.1333
N: 5 K: 50 nn_K: 1 m_rate:0.0556
N: 5 K: 50 nn_K: 1 m_rate:0.1111
N: 5 K: 50 nn_K: 1 m_rate:0.1333
average miss of 10 times: 0.1478
```

K=60

```
N: 5 K: 60 nn_K: 1 m_rate:0.1111
N: 5 K: 60 nn_K: 1 m_rate:0.1667
N: 5 K: 60 nn_K: 1 m_rate:0.0778
N: 5 K: 60 nn_K: 1 m_rate:0.0778
N: 5 K: 60 nn_K: 1 m_rate:0.1222
N: 5 K: 60 nn_K: 1 m_rate:0.0556
N: 5 K: 60 nn_K: 1 m_rate:0.1444
N: 5 K: 60 nn_K: 1 m_rate:0.1222
N: 5 K: 60 nn_K: 1 m_rate:0.0556
N: 5 K: 60 nn_K: 1 m_rate:0.1778
average miss of 10 times: 0.1111
```

K=70

K=80

```
N: 5 K: 70 nn_K: 1 m_rate:0.1778
N: 5 K: 70 nn_K: 1 m_rate:0.1556
N: 5 K: 70 nn_K: 1 m_rate:0.2222
N: 5 K: 70 nn_K: 1 m_rate:0.1444
N: 5 K: 70 nn_K: 1 m_rate:0.2444
N: 5 K: 70 nn_K: 1 m_rate:0.1111
N: 5 K: 70 nn_K: 1 m_rate:0.1556
N: 5 K: 70 nn_K: 1 m_rate:0.2111
N: 5 K: 70 nn_K: 1 m_rate:0.1333
N: 5 K: 70 nn_K: 1 m_rate:0.1667
average miss of 10 times: 0.1722
```

```
N: 5 K: 80 nn_K: 1 m_rate:0.1444
N: 5 K: 80 nn_K: 1 m_rate:0.1000
N: 5 K: 80 nn_K: 1 m_rate:0.1222
N: 5 K: 80 nn_K: 1 m_rate:0.1667
N: 5 K: 80 nn_K: 1 m_rate:0.0556
N: 5 K: 80 nn_K: 1 m_rate:0.1778
N: 5 K: 80 nn_K: 1 m_rate:0.2111
N: 5 K: 80 nn_K: 1 m_rate:0.0778
N: 5 K: 80 nn_K: 1 m_rate:0.0889
N: 5 K: 80 nn_K: 1 m_rate:0.1111
average miss of 10 times: 0.1256
```

When N=7:

K=10

```
N: 7 K: 10 nn_K: 1 m_rate:0.0500
N: 7 K: 10 nn_K: 1 m_rate:0.2833
N: 7 K: 10 nn_K: 1 m_rate:0.2500
N: 7 K: 10 nn_K: 1 m_rate:0.2500
N: 7 K: 10 nn_K: 1 m_rate:0.1833
N: 7 K: 10 nn_K: 1 m_rate:0.0667
N: 7 K: 10 nn_K: 1 m_rate:0.1167
N: 7 K: 10 nn_K: 1 m_rate:0.2500
N: 7 K: 10 nn_K: 1 m_rate:0.1667
N: 7 K: 10 nn_K: 1 m_rate:0.2500
average miss of 10 times: 0.1867
```

K=20

```
N: 7 K: 20 nn_K: 1 m_rate:0.1167
N: 7 K: 20 nn_K: 1 m_rate:0.1500
N: 7 K: 20 nn_K: 1 m_rate:0.2167
N: 7 K: 20 nn_K: 1 m_rate:0.0667
N: 7 K: 20 nn_K: 1 m_rate:0.2000
N: 7 K: 20 nn_K: 1 m_rate:0.2667
N: 7 K: 20 nn_K: 1 m_rate:0.2000
N: 7 K: 20 nn_K: 1 m_rate:0.1167
N: 7 K: 20 nn_K: 1 m_rate:0.2333
N: 7 K: 20 nn_K: 1 m_rate:0.1167
average miss of 10 times: 0.1683
```

K=30

```
N: 7 K: 30 nn_K: 1 m_rate:0.0667
N: 7 K: 30 nn_K: 1 m_rate:0.1833
N: 7 K: 30 nn_K: 1 m_rate:0.0500
N: 7 K: 30 nn_K: 1 m_rate:0.1500
N: 7 K: 30 nn_K: 1 m_rate:0.0500
N: 7 K: 30 nn_K: 1 m_rate:0.1333
N: 7 K: 30 nn_K: 1 m_rate:0.0667
N: 7 K: 30 nn_K: 1 m_rate:0.0500
N: 7 K: 30 nn_K: 1 m_rate:0.0500
N: 7 K: 30 nn_K: 1 m_rate:0.0167
average miss of 10 times: 0.0817
```

K=40

```
N: 7 K: 40 nn_K: 1 m_rate:0.0333
N: 7 K: 40 nn_K: 1 m_rate:0.1167
N: 7 K: 40 nn_K: 1 m_rate:0.2167
N: 7 K: 40 nn_K: 1 m_rate:0.0667
N: 7 K: 40 nn_K: 1 m_rate:0.0333
N: 7 K: 40 nn_K: 1 m_rate:0.1833
N: 7 K: 40 nn_K: 1 m_rate:0.1500
N: 7 K: 40 nn_K: 1 m_rate:0.1333
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.1167
average miss of 10 times: 0.105
```

K=50

```
N: 7 K: 50 nn_K: 1 m_rate:0.0167
N: 7 K: 50 nn_K: 1 m_rate:0.1667
N: 7 K: 50 nn_K: 1 m_rate:0.1833
N: 7 K: 50 nn_K: 1 m_rate:0.0333
N: 7 K: 50 nn_K: 1 m_rate:0.1667
N: 7 K: 50 nn_K: 1 m_rate:0.0167
N: 7 K: 50 nn_K: 1 m_rate:0.0667
N: 7 K: 50 nn_K: 1 m_rate:0.1833
N: 7 K: 50 nn_K: 1 m_rate:0.1333
N: 7 K: 50 nn_K: 1 m_rate:0.1500
average miss of 10 times: 0.1117
```

K=60

```
N: 7 K: 60 nn_K: 1 m_rate:0.1500
N: 7 K: 60 nn_K: 1 m_rate:0.1500
N: 7 K: 60 nn_K: 1 m_rate:0.2667
N: 7 K: 60 nn_K: 1 m_rate:0.1333
N: 7 K: 60 nn_K: 1 m_rate:0.0500
N: 7 K: 60 nn_K: 1 m_rate:0.0667
N: 7 K: 60 nn_K: 1 m_rate:0.0167
N: 7 K: 60 nn_K: 1 m_rate:0.1000
N: 7 K: 60 nn_K: 1 m_rate:0.1000
N: 7 K: 60 nn_K: 1 m_rate:0.0000
average miss of 10 times: 0.1033
```

K=70

K=80

N: 7 K: 70 nn_K: 1 m_rate:0.0167	N: 7 K: 80 nn_K: 1 m_rate:0.0333
N: 7 K: 70 nn_K: 1 m_rate:0.1500	N: 7 K: 80 nn_K: 1 m_rate:0.0333
N: 7 K: 70 nn_K: 1 m_rate:0.0500	N: 7 K: 80 nn_K: 1 m_rate:0.1167
N: 7 K: 70 nn_K: 1 m_rate:0.1833	N: 7 K: 80 nn_K: 1 m_rate:0.0167
N: 7 K: 70 nn_K: 1 m_rate:0.0167	N: 7 K: 80 nn_K: 1 m_rate:0.0000
N: 7 K: 70 nn_K: 1 m_rate:0.1333	N: 7 K: 80 nn_K: 1 m_rate:0.1167
N: 7 K: 70 nn_K: 1 m_rate:0.1500	N: 7 K: 80 nn_K: 1 m_rate:0.0167
N: 7 K: 70 nn_K: 1 m_rate:0.0667	N: 7 K: 80 nn_K: 1 m_rate:0.0167
N: 7 K: 70 nn_K: 1 m_rate:0.0000	N: 7 K: 80 nn_K: 1 m_rate:0.0167
N: 7 K: 70 nn_K: 1 m_rate:0.0000	N: 7 K: 80 nn_K: 1 m_rate:0.1500
average miss of 10 times: 0.0767	average miss of 10 times: 0.0517

(4) Repeat (2) and (3) for Fisher faces. Compare the simulation results of Eigenfaces and Fisher faces, and describe your findings.

When N=3. While computing the eigenvectors of the matrix, we may do sqrt to negative num, so there are complex number in result, which are difficult to handle in subsequent steps. I change them into float. K=10:

```
N: 3 K: 10 nn_K: 1 m_rate:0.1167
C:\Users\19041\Desktop\机器学习与模式识别\Project-1\pro1_吴瑞林
_201785072\HW1.py:139: ComplexWarning: Casting complex values to
real discards the imaginary part
  feature=np.transpose(feature).astype(np.float)
N: 3 K: 10 nn_K: 1 m_rate:0.1000
N: 3 K: 10 nn_K: 1 m_rate:0.1417
N: 3 K: 10 nn_K: 1 m_rate:0.0667
N: 3 K: 10 nn_K: 1 m_rate:0.1167
N: 3 K: 10 nn_K: 1 m_rate:0.0583
N: 3 K: 10 nn_K: 1 m_rate:0.1500
N: 3 K: 10 nn_K: 1 m_rate:0.1417
N: 3 K: 10 nn_K: 1 m_rate:0.1917
N: 3 K: 10 nn_K: 1 m_rate:0.1750
average miss of 10 times: 0.1258
```

K=20	**K=30**
N: 3 K: 20 nn_K: 1 m_rate:0.0917	N: 3 K: 30 nn_K: 1 m_rate:0.1000
N: 3 K: 20 nn_K: 1 m_rate:0.1750	N: 3 K: 30 nn_K: 1 m_rate:0.1250
N: 3 K: 20 nn_K: 1 m_rate:0.0833	N: 3 K: 30 nn_K: 1 m_rate:0.0917
N: 3 K: 20 nn_K: 1 m_rate:0.0417	N: 3 K: 30 nn_K: 1 m_rate:0.0917
N: 3 K: 20 nn_K: 1 m_rate:0.2167	N: 3 K: 30 nn_K: 1 m_rate:0.1083
N: 3 K: 20 nn_K: 1 m_rate:0.0750	N: 3 K: 30 nn_K: 1 m_rate:0.1917
N: 3 K: 20 nn_K: 1 m_rate:0.0500	N: 3 K: 30 nn_K: 1 m_rate:0.0583
N: 3 K: 20 nn_K: 1 m_rate:0.1500	N: 3 K: 30 nn_K: 1 m_rate:0.1417
N: 3 K: 20 nn_K: 1 m_rate:0.1417	N: 3 K: 30 nn_K: 1 m_rate:0.1333
N: 3 K: 20 nn_K: 1 m_rate:0.1083	N: 3 K: 30 nn_K: 1 m_rate:0.0917
average miss of 10 times: 0.1133	average miss of 10 times: 0.1133

K=40

K=50

```
N: 3 K: 40 nn_K: 1 m_rate:0.0750
N: 3 K: 40 nn_K: 1 m_rate:0.1167
N: 3 K: 40 nn_K: 1 m_rate:0.1167
N: 3 K: 40 nn_K: 1 m_rate:0.0833
N: 3 K: 40 nn_K: 1 m_rate:0.1333
N: 3 K: 40 nn_K: 1 m_rate:0.0833
N: 3 K: 40 nn_K: 1 m_rate:0.0917
N: 3 K: 40 nn_K: 1 m_rate:0.0667
N: 3 K: 40 nn_K: 1 m_rate:0.1167
N: 3 K: 40 nn_K: 1 m_rate:0.2083
average miss of 10 times: 0.1092
```

```
N: 3 K: 50 nn_K: 1 m_rate:0.0917
N: 3 K: 50 nn_K: 1 m_rate:0.2167
N: 3 K: 50 nn_K: 1 m_rate:0.1000
N: 3 K: 50 nn_K: 1 m_rate:0.1167
N: 3 K: 50 nn_K: 1 m_rate:0.0667
N: 3 K: 50 nn_K: 1 m_rate:0.1083
N: 3 K: 50 nn_K: 1 m_rate:0.1083
N: 3 K: 50 nn_K: 1 m_rate:0.0583
N: 3 K: 50 nn_K: 1 m_rate:0.0500
N: 3 K: 50 nn_K: 1 m_rate:0.1917
average miss of 10 times: 0.1108
```

When N=5

K=10

```
N: 5 K: 10 nn_K: 1 m_rate:0.0444
N: 5 K: 10 nn_K: 1 m_rate:0.0889
N: 5 K: 10 nn_K: 1 m_rate:0.0778
N: 5 K: 10 nn_K: 1 m_rate:0.1222
N: 5 K: 10 nn_K: 1 m_rate:0.0333
N: 5 K: 10 nn_K: 1 m_rate:0.1778
N: 5 K: 10 nn_K: 1 m_rate:0.0778
N: 5 K: 10 nn_K: 1 m_rate:0.0556
N: 5 K: 10 nn_K: 1 m_rate:0.0556
N: 5 K: 10 nn_K: 1 m_rate:0.0889
average miss of 10 times: 0.0822
```

K=20

```
N: 5 K: 20 nn_K: 1 m_rate:0.0889
N: 5 K: 20 nn_K: 1 m_rate:0.0333
N: 5 K: 20 nn_K: 1 m_rate:0.1333
N: 5 K: 20 nn_K: 1 m_rate:0.0444
N: 5 K: 20 nn_K: 1 m_rate:0.0778
N: 5 K: 20 nn_K: 1 m_rate:0.0333
N: 5 K: 20 nn_K: 1 m_rate:0.1333
N: 5 K: 20 nn_K: 1 m_rate:0.0556
N: 5 K: 20 nn_K: 1 m_rate:0.0667
N: 5 K: 20 nn_K: 1 m_rate:0.1111
average miss of 10 times: 0.0778
```

K=30

```
N: 5 K: 30 nn_K: 1 m_rate:0.1111
N: 5 K: 30 nn_K: 1 m_rate:0.0111
N: 5 K: 30 nn_K: 1 m_rate:0.0667
N: 5 K: 30 nn_K: 1 m_rate:0.0556
N: 5 K: 30 nn_K: 1 m_rate:0.2000
N: 5 K: 30 nn_K: 1 m_rate:0.0333
N: 5 K: 30 nn_K: 1 m_rate:0.1333
N: 5 K: 30 nn_K: 1 m_rate:0.0444
N: 5 K: 30 nn_K: 1 m_rate:0.0333
N: 5 K: 30 nn_K: 1 m_rate:0.0333
average miss of 10 times: 0.0722
```

K=40

```
N: 5 K: 40 nn_K: 1 m_rate:0.0444
N: 5 K: 40 nn_K: 1 m_rate:0.1556
N: 5 K: 40 nn_K: 1 m_rate:0.0556
N: 5 K: 40 nn_K: 1 m_rate:0.0333
N: 5 K: 40 nn_K: 1 m_rate:0.0333
N: 5 K: 40 nn_K: 1 m_rate:0.0667
N: 5 K: 40 nn_K: 1 m_rate:0.0444
N: 5 K: 40 nn_K: 1 m_rate:0.0778
N: 5 K: 40 nn_K: 1 m_rate:0.1667
N: 5 K: 40 nn_K: 1 m_rate:0.0333
average miss of 10 times: 0.0711
```

K=50

```
N: 5 K: 50 nn_K: 1 m_rate:0.1667
N: 5 K: 50 nn_K: 1 m_rate:0.0333
N: 5 K: 50 nn_K: 1 m_rate:0.0667
N: 5 K: 50 nn_K: 1 m_rate:0.0444
N: 5 K: 50 nn_K: 1 m_rate:0.0444
N: 5 K: 50 nn_K: 1 m_rate:0.1000
N: 5 K: 50 nn_K: 1 m_rate:0.2222
N: 5 K: 50 nn_K: 1 m_rate:0.0667
N: 5 K: 50 nn_K: 1 m_rate:0.0556
N: 5 K: 50 nn_K: 1 m_rate:0.0222
average miss of 10 times: 0.0822
```

K=60

```
N: 5 K: 60 nn_K: 1 m_rate:0.0000
N: 5 K: 60 nn_K: 1 m_rate:0.1333
N: 5 K: 60 nn_K: 1 m_rate:0.0667
N: 5 K: 60 nn_K: 1 m_rate:0.2000
N: 5 K: 60 nn_K: 1 m_rate:0.0222
N: 5 K: 60 nn_K: 1 m_rate:0.0556
N: 5 K: 60 nn_K: 1 m_rate:0.0667
N: 5 K: 60 nn_K: 1 m_rate:0.0111
N: 5 K: 60 nn_K: 1 m_rate:0.0111
N: 5 K: 60 nn_K: 1 m_rate:0.0556
average miss of 10 times: 0.0622
```

When N=7

K=10

```
N: 7 K: 10 nn_K: 1 m_rate:0.0500
N: 7 K: 10 nn_K: 1 m_rate:0.1333
N: 7 K: 10 nn_K: 1 m_rate:0.0667
N: 7 K: 10 nn_K: 1 m_rate:0.0167
N: 7 K: 10 nn_K: 1 m_rate:0.2500
N: 7 K: 10 nn_K: 1 m_rate:0.0000
N: 7 K: 10 nn_K: 1 m_rate:0.0333
N: 7 K: 10 nn_K: 1 m_rate:0.0333
N: 7 K: 10 nn_K: 1 m_rate:0.1333
N: 7 K: 10 nn_K: 1 m_rate:0.0500
average miss of 10 times: 0.0767
```

K=20

```
N: 7 K: 20 nn_K: 1 m_rate:0.0000
N: 7 K: 20 nn_K: 1 m_rate:0.0500
N: 7 K: 20 nn_K: 1 m_rate:0.3667
N: 7 K: 20 nn_K: 1 m_rate:0.0667
N: 7 K: 20 nn_K: 1 m_rate:0.0500
N: 7 K: 20 nn_K: 1 m_rate:0.0500
N: 7 K: 20 nn_K: 1 m_rate:0.1000
N: 7 K: 20 nn_K: 1 m_rate:0.0333
N: 7 K: 20 nn_K: 1 m_rate:0.0500
N: 7 K: 20 nn_K: 1 m_rate:0.1000
average miss of 10 times: 0.0867
```

K=30

```
N: 7 K: 30 nn_K: 1 m_rate:0.0833
N: 7 K: 30 nn_K: 1 m_rate:0.0000
N: 7 K: 30 nn_K: 1 m_rate:0.0833
N: 7 K: 30 nn_K: 1 m_rate:0.0500
N: 7 K: 30 nn_K: 1 m_rate:0.1833
N: 7 K: 30 nn_K: 1 m_rate:0.0333
N: 7 K: 30 nn_K: 1 m_rate:0.2000
N: 7 K: 30 nn_K: 1 m_rate:0.0000
N: 7 K: 30 nn_K: 1 m_rate:0.0000
N: 7 K: 30 nn_K: 1 m_rate:0.0667
average miss of 10 times: 0.07
```

K=40

```
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.0500
N: 7 K: 40 nn_K: 1 m_rate:0.0333
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.0333
N: 7 K: 40 nn_K: 1 m_rate:0.0000
N: 7 K: 40 nn_K: 1 m_rate:0.0833
average miss of 10 times: 0.02
```

K=50

```
N: 7 K: 50 nn_K: 1 m_rate:0.0667
N: 7 K: 50 nn_K: 1 m_rate:0.1167
N: 7 K: 50 nn_K: 1 m_rate:0.0167
N: 7 K: 50 nn_K: 1 m_rate:0.0000
N: 7 K: 50 nn_K: 1 m_rate:0.1333
N: 7 K: 50 nn_K: 1 m_rate:0.0000
N: 7 K: 50 nn_K: 1 m_rate:0.0000
N: 7 K: 50 nn_K: 1 m_rate:0.0167
N: 7 K: 50 nn_K: 1 m_rate:0.0667
N: 7 K: 50 nn_K: 1 m_rate:0.2000
average miss of 10 times: 0.0617
```

K=60

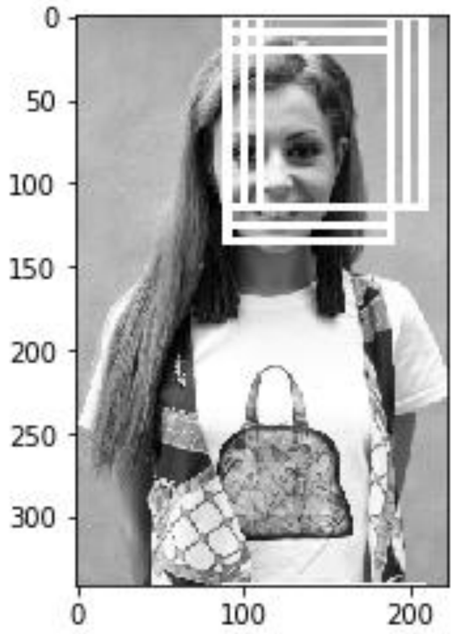
```
N: 7 K: 60 nn_K: 1 m_rate:0.0333
N: 7 K: 60 nn_K: 1 m_rate:0.0167
N: 7 K: 60 nn_K: 1 m_rate:0.0833
N: 7 K: 60 nn_K: 1 m_rate:0.0667
N: 7 K: 60 nn_K: 1 m_rate:0.0000
N: 7 K: 60 nn_K: 1 m_rate:0.0667
N: 7 K: 60 nn_K: 1 m_rate:0.0000
N: 7 K: 60 nn_K: 1 m_rate:0.1000
N: 7 K: 60 nn_K: 1 m_rate:0.0000
N: 7 K: 60 nn_K: 1 m_rate:0.2333
average miss of 10 times: 0.06
```

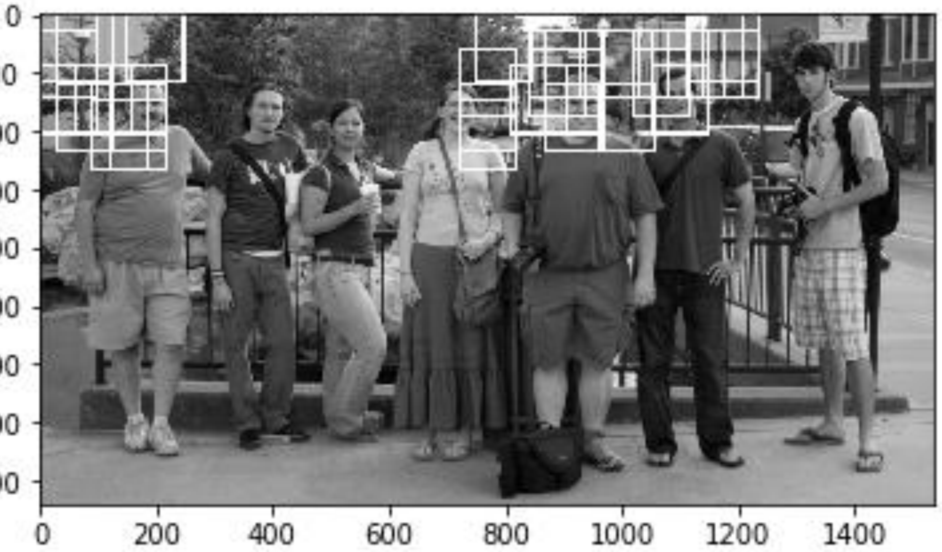
Analysis:

Warning： The problem of singular matrix appears when Fisherfaces algorithm is implemented. Personally, I think that's because for a small number of samples, there must be linearly dependent rows or columns. So I first use the PCA method to reduce the dimensions of the sample to NUM-category to ensure that Sw is non-exotic.

Conclusion 1: The result is better when N is promoted (the training sample used is increased). For PCA, when N=3, the Misclassification rate is 0.15-0.2.When N=5, it is 0.1-0.15.When N=7, it is around 0.1 and even has a misclassification rate of 0.05. This is because more samples are used, more information is available, more model training is available, so the effect is better. Because the sample size is small, until an overfitting is achieved, the more samples are used, the better effect we get.
Conclusion 2: When K is promoted, the main components retained become more and the Misclassification rate decreases. When a certain degree is exceeded, the newly retained dimensions no longer contain much major information, but are affected by noise points, thus the Misclassification rate increases. In PCA and LDA, the best dmension is about 40 to 50. For LDA with N=7 and K=40, 0.02 can appear, which is comparable to the misclassification rate of simple neural network.
Conclusion 3: LDA is better than PCA for labeled classification. It can be seen that lDA is better than PCA in terms of the lowest, highest and average value of all part of misclassification Rate. This is because the LDA uses tag information to separate the data to a greater extent.

(5) Data: project1-data-Detection.zip (Downloaded from Flickr). In this problem, you are asked to develop an algorithm for face detection based on the intermediate or final results of (2)-(4). Please use a rectangle (or square) to indicate the detected faces in the given 1 images. In the report, you should illustrate your algorithm clearly, discuss the challenges of face detection, and the strategies adopted in your algorithm to overcome these challenges.

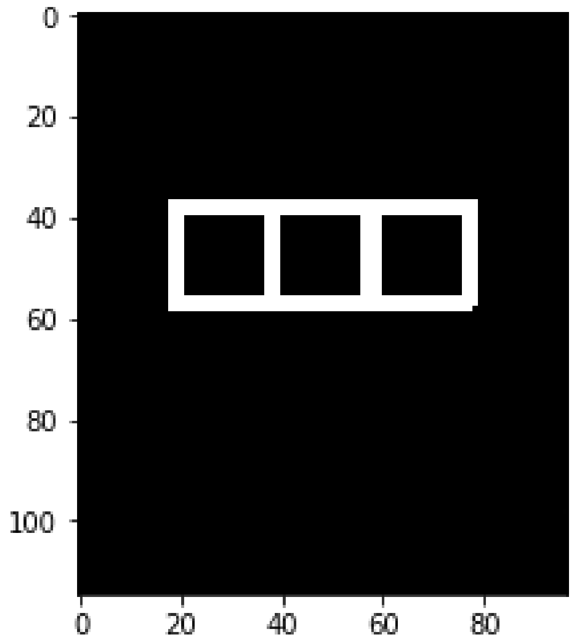




Challenges1:windows & time.

For avery window, if I use a complex mode to detect wether if the window is a face, the time is too long, and when sorting for the results, the result list is too long as well. So I designed a two-layer test, first I use a pretest to make a simple test, if the pretest consider the window is a face, use a complex method to detect the window and apend the window into result list.

When in pretest, I detect the eyes, if the sum of pixel of lefteye add the sum of pixel of righteye minus the sum of pixel of nose can below a threshold value, it detect the window might be a face.



Challenges2:Mean face and pca face.

At first, I trained a pca mode to detect face, soon I find that the pca transform and prediction takes too much time, which is unacceptable. I tried use mean face instead of train a pca mode, to my surprise, the results didn't change significantly, so finally I decide to use mean face.

Challenges3: scale space and muti-face

I manually adjusted and prompted the number of faces in the image. I have two solutions to the problem, but unfortunately I failed to code successfully. First, traverse in scale space to get the best scale space, and do next work on the scale space, but this will do take a lot of computing time. Second, it is obvious that a face can be contained in multiple windows. By using DBSCAN and other non-parametric clustering methods, similar windows can be grouped into one type, and similar Windows can be located to a face. The total number of categories is the number of faces.

Challenges4:position

For this problem, considering that the face is obviously above the clothes, the longitudinal detection limit is adjusted during detection to avoid detection of clothes.