Tranditional Face Recognition Methods

1: 理解 PCA (Pricipal Component Analysis)

数据 x 为 10 个样本的二维特征，使用 PCA 将其降维至一维

x = [[2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2.0, 1.0, 1.5, 1.1],
[2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9]]

去中心化：

x =[[0.69 -1.31 0.39 0.09 1.29 0.49 0.19 -0.81 -0.31 -0.71]
[0.49 -1.21 0.99 0.29 1.09 0.79 -0.31 -0.81 -0.31 -1.01]]

计算原始数据的协方差矩阵：

[[0.61655556 0.61544444]
[0.61544444 0.71655556]]

求协方差矩阵的特征值与特征向量

特征值：[0.0490834 1.28402771]

特征向量：[[-0.73517866 -0.6778734]
[0.6778734 -0.73517866]]

降为 1 维，取最大特征值对应特征向量组成矩阵

CY = [[1.28402771 0.]
[0. 1.28402771]]

W = [[-0.6778734 -0.73517866]]

P = W.transpose()

计算投影后的数据

Y = [-0.82797019 1.77758033 -0.99219749 -0.27421042 -1.67580142 -0.9129491
0.09910944 1.14457216 0.43804614 1.22382056]

2. ORL 人脸数据集共包含 40 个不同人的 400 张图像，是由英国剑桥的 Olivetti 研究实验室创建，请代码实现 PCA 在 ORL 数据库上的人脸识别。

s1/10.pgm is the most similar to s1/5.pgm

s2/10.pgm is the most similar to s2/7.pgm

s3/10.pgm is the most similar to s3/9.pgm

s4/10.pgm is the most similar to s4/5.pgm

s5/10.pgm is the most similar to s40/5.pgm

s6/10.pgm is the most similar to s6/4.pgm

s7/10.pgm is the most similar to s7/5.pgm

s8/10.pgm is the most similar to s8/3.pgm
s9/10.pgm is the most similar to s9/5.pgm
s10/10.pgm is the most similar to s8/3.pgm
s11/10.pgm is the most similar to s11/1.pgm
s12/10.pgm is the most similar to s12/9.pgm
s13/10.pgm is the most similar to s13/5.pgm
s14/10.pgm is the most similar to s14/9.pgm
s15/10.pgm is the most similar to s15/2.pgm
s16/10.pgm is the most similar to s16/3.pgm
s17/10.pgm is the most similar to s17/7.pgm
s18/10.pgm is the most similar to s18/8.pgm
s19/10.pgm is the most similar to s19/1.pgm
s20/10.pgm is the most similar to s20/2.pgm
s21/10.pgm is the most similar to s21/8.pgm
s22/10.pgm is the most similar to s22/2.pgm
s23/10.pgm is the most similar to s23/1.pgm
s24/10.pgm is the most similar to s24/9.pgm
s25/10.pgm is the most similar to s25/3.pgm
s26/10.pgm is the most similar to s26/8.pgm
s27/10.pgm is the most similar to s27/1.pgm
s28/10.pgm is the most similar to s28/3.pgm
s29/10.pgm is the most similar to s29/3.pgm
s30/10.pgm is the most similar to s30/1.pgm
s31/10.pgm is the most similar to s31/5.pgm
s32/10.pgm is the most similar to s32/6.pgm
s33/10.pgm is the most similar to s33/2.pgm
s34/10.pgm is the most similar to s34/9.pgm
s35/10.pgm is the most similar to s35/5.pgm
s36/10.pgm is the most similar to s36/6.pgm
s37/10.pgm is the most similar to s37/9.pgm
s38/10.pgm is the most similar to s38/5.pgm
s39/10.pgm is the most similar to s39/6.pgm
s40/10.pgm is the most similar to s40/4.pgm
accuracy: 0.950000

3. LBP (Local Binary Pattern)

手动计算以下图像像素点（红色）所对应的LBP值。

51	72	58
66	68	69
64	76	59

LBP = (0 1 0 1 0 1 0 0) = 84

```
In [1]: import numpy as np

In [21]: wind = np.array([
          [ 51, 72, 58],
          [ 66, 68, 69],
          [ 64, 76, 59]])
         wind

Out[21]: array([[51, 72, 58],
                [66, 68, 69],
                [64, 76, 59]])

In [22]: wind = wind - wind[1,1]

In [23]: clip = np.clip(wind, 0, 1)

In [24]: pattern = np.array([
             [128, 64, 32],
             [1,   0,  16],
             [2,   4,  8]
         ])

In [26]: lbp_mat = clip * pattern

In [27]: lbp = lbp_mat.sum()
         lbp

Out[27]: 84
```

4. 编程实现人脸的 LBP 值

5. SVM