

基于 PCA 的人脸识别

学号:16340300

姓名:赵佳乐

1. 算法描述

主要步骤如下：

1. 首先在人脸数据库中每类取出 7 张图片用于训练，剩下 3 张图片用来测试。
2. 将待训练的每个图像作为一个行向量放入矩阵里，最后组成 $280 \times n$ 的矩阵 T ， n 为每个图像像素大小。
3. 对矩阵 T 每列求均值，得到一个行向量， T 中的每行减去这个向量从而中心化。
4. 矩阵 T 乘 T 的转置，在对其求特征向量和特征矩阵，这里用的是 `eigs` 函数，可以求出指定的 K 个最大的特征值组成的特征向量和对应的特征矩阵。因为特征值较大的部分是不同图像之间差异最大的部分，而特征值较小的部分差异较小，对图像检测的影响很小，我们可以舍去。
5. 再将 T 的转置乘上第四步得到的特征矩阵，将其转置后得到 $K \times n$ 的矩阵 E ，其每一行向量就作为后续映射关系的一组基向量，共有 K 个基向量。
6. 在进行人脸识别时，我们先把之前的矩阵 T 用上一步得到的矩阵 E 映射到新的特征空间，即 $E \cdot T'$ 。之后我们对待测图像做同样的操作映射到新的特征空间，再对比待测图像与我们训练图像在新的特征空间中的坐标，求出二范数(向量间的距离)，最小的那个就是最接近待测图像的训练样本图像，并找出其类别即可完成人脸识别。(待测图像也要进行中心化处理)

2. 代码功能

`CreateTrainingSet` 获得由所有训练图像组成的矩阵 T

`EigenfaceCore` 训练函数，得到对新的特征向量的映射矩阵

`Recognition` 单张图像的测试函数，参数 `flag=1` 时会输出结果

`test` 测试所有图像，返回正确率

`testK` 测试不同的 K 值，返回正确率

详细代码见 `code` 文件夹。

运行步骤:

```
T=CreateTrainingSet('att_faces');
```

```
[MeanFace, MeanNormFaces, EigenFaces] = EigenfaceCore(T,70); %这里的 70 是 K 值,也可以换成其他值
```

```
Recognition('att_faces\s35\8.pgm', MeanFace, MeanNormFaces, EigenFaces,1); %第一个参数是测试图像的路径
```

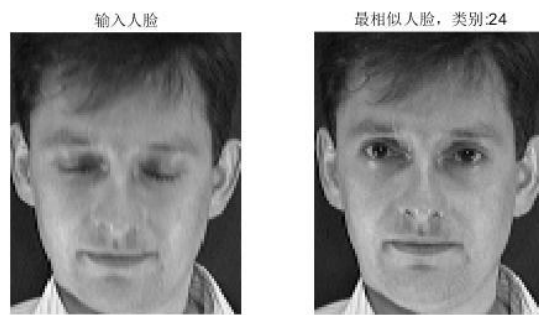
程序运行可能花费较长时间，请耐心等待。

3. 测试性能表格

对于单张图像的测试:

K 值取 70

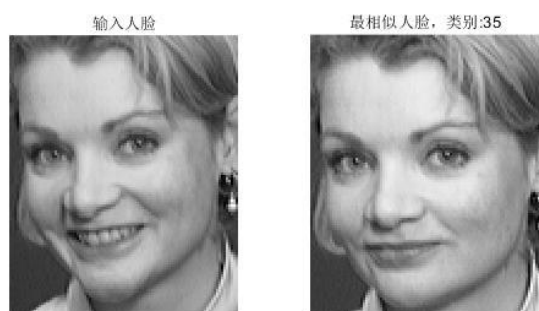
```
Recognition('att_faces\s24\8.pgm', MeanFace, MeanNormFaces, EigenFaces,1);
```



Recognition('att_faces\s15\9.pgm', MeanFace, MeanNormFaces, EigenFaces,1);

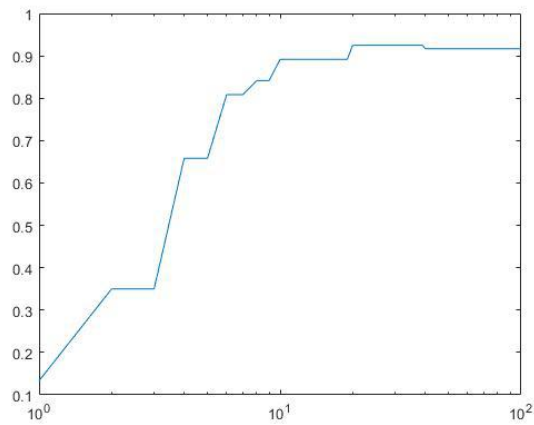


Recognition('att_faces\s35\9.pgm', MeanFace, MeanNormFaces, EigenFaces,1);



最后的正确率约为:91.67%

取不同 K 值测试，因为数据太多，为了节省时间只取了部分 K 值:

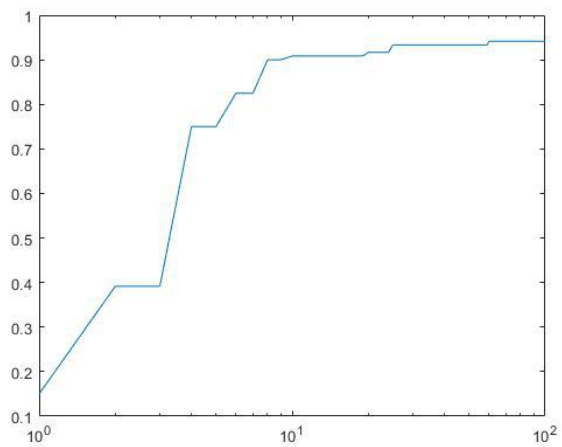


可以看到 K 取较小值的时候正确率变化较大，K 大于 50 之后正确率几乎保持 92% 左右不会变化，这与算法相符，特征值较大的部分影响图像检测，而较小的部分几乎不影响图像检测。

另外，不同的样本图像也会对结果产生一点影响，本次实验取的是前 7 张图片作为样本，若取后七张图作为样本，前三张图测试，结果为：

K 值取 70 时，正确率：94.17%

取不同 K 值时：



曲线与前面基本符合。

