# 程序报告

学号：2211313　　姓名：贾景顺

## 一、问题重述

采用特征脸（Eigenface）算法来进行有效的人脸识别，其核心原理是对向量化后的人脸图像进行主成分分析（PCA）。实验过程需要计算平均脸和特征脸，并利用前 K 个特征脸进行测试人脸的识别。另外还需实现对给定的图片，利用前 K 个特征脸重构图片的功能。

## 二、设计思想

在对原 ORL 人脸库进行读取以及训练/测试集的划分后，正式进入特征脸算法的部分。

首先需要在 eigen_train 函数中计算平均人脸、特征人脸以及中心化人脸。平均人脸即为对训练数据按列取平均值，计算所有样本人脸在各像素的灰度平均值，使用 np.mean（np 即为 numpy 库，下同）实现。将原训练集人脸各维度数据对应减去平均值，所得即为中心化人脸。计算特征脸可采用 np.linalg.svd 函数来进行奇异值分解并保留右奇异矩阵，矩阵的列向量则对于各特征脸向量，随后按参数 k 进行特征脸的保留。

计算特征脸后即可利用测试集进行人脸识别，该功能由 rep_face 函数实现。首先利用上一步计算的平均脸对输入的待识别人脸进行中心化，随后确定特征脸使用数量（numEigenface），即为输入 numComponents 和特征脸向量的行数的较小值。最后利用 np 实现矩阵乘法，计算中心化后的人脸在选定特征脸上的投影（即矩阵的点乘）。识别过程即计算测试人脸与训练人脸在特征投影下的欧氏距离，分类为距离最小的样本，并与测试集样本自身标签进行匹配。在设置主特征数=200 时，识别正确率为 91.25%，识别正确率较高。

最后进行人脸重建功能的实现，事实上人脸重建模型即为人脸识别模型的逆过程。首先和上一部分相同，选取特征脸向量的 numComponents（输入和特征脸向量行数的较小值）行，并进行归一化。随后根据选定的主特征数，计算输入人脸图像归一化后在特征向量上的投影，随后再进行去归一化，即重新加上对应的平均脸，并将维度还原为原始图像尺寸，即可完成对人脸图像的重建。

## 三、代码内容

```
#特征人脸算法
def eigen_train(trainset, k=20):
    avg_img = np.mean(trainset, axis=0)
    norm_img = trainset - avg_img
    _, _, Vt = np.linalg.svd(norm_img, full_matrices=False)
    feature = Vt[:k]
    return avg_img, feature, norm_img
================================================================================
#人脸识别模型
def rep_face(image, avg_img, eigenface_vects, numComponents = 0):
    max_available = eigenface_vects.shape[0]
    if numComponents == 0:
```

```
        numEigenFaces = max_available
    else:
        numEigenFaces = min(numComponents, max_available)
    normalized = image - avg_img
    representation = np.dot(eigenface_vects[:numEigenFaces], normalized)
    return representation, numEigenFaces
==========================================================================
#人脸重建模型
def recFace(representations, avg_img, eigenVectors, numComponents, sz=(112,92)):
    max_available = eigenVectors.shape[0]
    if numComponents <= 0:
        numEigenFaces = max_available
    else:
        numEigenFaces = min(numComponents, max_available)
    eigen_subset = eigenVectors[:numEigenFaces]
    coefficients = representations[:numEigenFaces]
    reconstructed = avg_img + np.dot(coefficients, eigen_subset)
    face = reconstructed.reshape(sz)
    return face, 'numEigenFaces_{}'.format(numComponents)
```

## 四、实验结果

```
print("人脸识别准确率: {}%".format(num / 80 * 100))
```

```
人脸识别准确率: 91.25%
```

| 测试点 | 状态 | 时长 | 结果 |
| --- | --- | --- | --- |
| 测试结果 | ✓ | 3s | 测试成功! |

模型对人脸正确识别的准确率为 91.25%，且能通过测试点。

## 五、总结

在本次实验中，通过对特征脸算法的学习，我复习了主成分分析（PCA）算法的主要实现过程，也增进了阅读英文科学论文的经验。对于特征脸算法的实现过程以及使用方式也有了较为全面的理解。同时，在对特征脸算法的复现过程中，通过 numpy 库向量操作对矩阵的 SVD 分解，点乘等计算实现运用的较为熟练。通过数学公式的推导和代码编写成功解决了问题，也使得我倍感收获。在本次实验中，遇到了一些因为不熟悉矩阵操作和题目要求的输入输出导致矩阵规模不匹配的情况，但都根据.shape 的方式增加对数据的了解完成化解。本次实验的不足之处包括没有使用更大的人脸训练数据集，若使用更大的数据集或许可以获得更好的算法实现。