**南京邮电大学**

**自动化学院、人工智能学院**

**实 验 报 告**

**课程名称：** 模式识别（双语）A

**实验名称：** 特征选择和降维实验

**指导教师：** 谢九成

**所在专业：**  人工智能

**学生姓名：** 单家俊

**班级学号：** B210416 B21080526

**2023 /2024 学年第 一 学期**

**实验名称：特征选择和降维实验**

**一、实验目的**

1. 理解主成分分析（PCA）原理
2. 实现基于PCA的人脸识别系统
3. 分析主成分数目的选取对识别结果的影响

**二、实验任务**

实现基于 PCA 的人脸识别，采用 yale 人脸库同时实验分析主成分数目的选取对识别结果的影响。

**三、实验内容**

import os  
from PIL import Image  
import numpy as np  
import cv2  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif'] = ['SimHei']  
def gif2png():  
 path = "D:/QQ/1413679561/FileRecv/exp/exp/yale/train"  
 savepath = "D:/QQ/1413679561/FileRecv/exp/exp/yale/train\_png"  
  
 testpath = "D:/QQ/1413679561/FileRecv/exp/exp/yale/eval"  
 testsavepath = "D:/QQ/1413679561/FileRecv/exp/exp/yale/eval\_png"  
  
 filelist = os.listdir(path)  
 for file in filelist:  
 im = Image.open(os.path.join(path, file))  
 filename = os.path.splitext(file)[0]  
 im.save(os.path.join(savepath, filename + '.png'))  
  
 filelist = os.listdir(testpath)  
 for file in filelist:  
 im = Image.open(os.path.join(testpath, file))  
 filename = os.path.splitext(file)[0]  
 im.save(os.path.join(testsavepath, filename + '.png'))  
  
  
def data\_process():  
 folder\_path = "D:/QQ/1413679561/FileRecv/exp/exp/yale/train\_png"  
 image\_array\_list = []  
  
 for filename in os.listdir(folder\_path):  
 if filename.endswith(".png"):  
 file\_path = os.path.join(folder\_path, filename)  
 image = Image.open(file\_path)  
 image\_array = np.array(image)  
 image\_array\_list.append(image\_array)  
  
  
 images\_np\_array = np.array(image\_array\_list)  
 *# print(images\_np\_array.shape)  
 # 每个人有7张图像，总共15个人* num\_images\_per\_person = 7  
 num\_people = 15  
 label\_np\_array = np.array([i for i in range(1, num\_people + 1) for \_ in range(num\_images\_per\_person)])  
  
 *# 计算平均脸* average\_face = np.mean(images\_np\_array, axis=0).astype(np.uint8)  
 average\_face\_image = Image.fromarray(average\_face)  
 average\_face\_image.save("D:/QQ/1413679561/FileRecv/exp/exp/yale/average\_face.png")  
  
 return images\_np\_array, label\_np\_array, average\_face  
  
  
def load\_test\_images(test\_folder\_path):  
 image\_array\_list = []  
 image\_label\_list = []  
 filename = os.listdir(test\_folder\_path)  
 for file in filename:  
 label = file[0:2]  
 if '0' in label:  
 label.rsplit('0')  
 image\_label\_list.append(int(label))  
  
 true\_labels\_np\_array = np.array(image\_label\_list)  
 for filename in os.listdir(test\_folder\_path):  
 if filename.endswith(".png"):  
 file\_path = os.path.join(test\_folder\_path, filename)  
 image = Image.open(file\_path)  
 image\_array = np.array(image)  
 image\_array\_list.append(image\_array)  
  
 test\_images\_np\_array = np.array(image\_array\_list)  
 return test\_images\_np\_array, true\_labels\_np\_array  
  
  
def evaluate\_model(recognizer, test\_images\_np\_array, true\_labels\_np\_array):  
 correct\_count = 0  
 total\_count = len(true\_labels\_np\_array)  
  
 for i in range(len(test\_images\_np\_array)):  
 label, confidence = recognizer.predict(test\_images\_np\_array[i])  
 if label == true\_labels\_np\_array[i]:  
 correct\_count += 1  
  
 accuracy = correct\_count / total\_count  
 return accuracy  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 gif2png()  
 images\_np\_array, label\_np\_array, average\_face = data\_process()  
 *# print("数组维度:", images\_np\_array.shape)  
 # print(images\_np\_array)  
  
 # 显示生成的平均脸* plt.imshow(average\_face, cmap='gray')  
 plt.title('平均脸')  
 plt.axis('off')  
 plt.show()  
  
 *# 准备验证集数据* test\_folder\_path = "D:/QQ/1413679561/FileRecv/exp/exp/yale/eval\_png"  
 test\_images\_np\_array, true\_labels\_np\_array = load\_test\_images(test\_folder\_path)  
 *# print(len(true\_labels\_np\_array))  
  
 # 评估模型并记录准确率* pca\_dimensions = range(1, 200)  
 accuracies = []  
  
 for pca\_dimension in pca\_dimensions:  
 i = 0;  
 recognizer = cv2.face.EigenFaceRecognizer\_create(num\_components=pca\_dimension)  
 recognizer.train(images\_np\_array, label\_np\_array)  
  
 accuracy = evaluate\_model(recognizer, test\_images\_np\_array, true\_labels\_np\_array)  
 accuracies.append(accuracy)  
 i += 1  
  
 *# 绘制PCA维数与准确率的关系图* plt.plot(pca\_dimensions, accuracies)  
 plt.title('PCA维数与准确率的关系')  
 plt.xlabel('PCA维数')  
 plt.ylabel('准确率')  
 plt.show()

**四、实验过程描述**

**实验结果分析：**

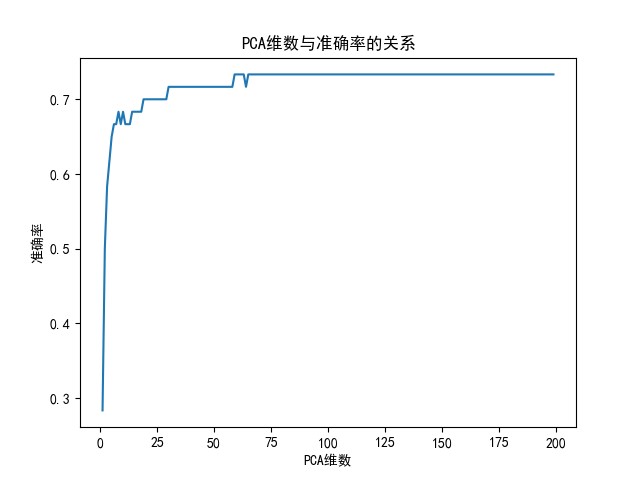


图1 主成分数目的选取对识别结果的影响



图2 生成的平均脸

通过循环尝试不同的PCA维数，你、训练了多个EigenFace Recognizer模型，并记录了它们在测试集上的准确率。通过绘制PCA维数与准确率的关系图，可以直观地观察到随着PCA维数的升高，人脸识别的准确率越来越高。

在展示生成的平均脸之前，通过计算平均脸的方式提供了对训练集整体特征的可视化。平均脸的生成是对数据集整体结构的一种概括，有助于了解共性特征。

**实验过程遇到的问题：**

* 问题描述： 在实验中，需要将图像转换为NumPy数组以便进行后续的数据处理和模型训练。
* 解决方案： 使用Python图像处理库PIL（Pillow）的Image.open函数打开图像文件，然后通过np.array()将图像转换为NumPy数组。
* 问题描述： 在模型训练之后，如何评估模型的准确性，如何判断模型是否对测试集进行了正确的预测。
* 解决方案： 使用训练好的模型对测试集进行预测，然后与真实标签进行比较。可以通过计算准确率等指标来评估模型性能。

**体会：**

PCA在人脸识别等领域的应用具有重要的意义，它不仅能够简化数据，提高计算效率，还能够提取关键特征，增强模型的稳定性和泛化能力。通过这次实验让我熟悉了PCA的原理以及主成分数目的选取对识别结果的影响。