

项目介绍:

该实验通过 PCA 降维+BP 神经网络的算法实现对面脸数据集中人脸数据的识别。

1、图片预处理

首先将测试集和训练集图片转化为灰度图，为了减少背景对实验结果的影响，使用 OpenCV 中继承好的 **CascadeClassifier** 级联分类器从原始的灰度图中识别并将人脸切取出来并保存。由于人脸的不规则性所以切取出来的人脸数据图片的大小并不一致，所以将图片 reshape 为 200×200 的像素大小。

2、PCA 降维

经过预处理后的图片已经有统一的格式，都有 200×200 个像素，即 40000 个特征，我们的目标是通过 PCA 算法建立映射空间将原始数据映射到低维空间中从而实现对初始数据的降维，并保存降维后的训练数据和测试数据，便于后期神经网络的分类运算，步骤如下：

- 1、设有 n 训练集中的图片，每张训练集图片的像素为 $c=a \times b(200 \times 200)$
- 2、将每张图片的 ab 矩阵转换 c 维列向量的形式构成 n 列的矩阵 $X(c \times n)$
- 3、对矩阵 X 进行均值、中心化操作、并求得协方差矩阵
- 4、求取协方差矩阵的特征值并取 k 个特征值(k :取值于使累计贡献率 $>90\%$ 的特征值数量)所对应的特征向量 V (特征脸)。
- 5、将 K 个特征向量合并成特征空间 $T(c \times k)$
- 6、将原始数据矩阵投射到该特征空间得到降维矩阵。

3、BP 神经网络实现对图像的分类

使用 pytorch 框架构建神经网络，并进行训练分类，步骤如下：

- 1、重写 Getloader 类，使用 Dataloader 迭代器处理训练数据和测试数据，构建用于模型训练和测试的数据结构，便于后期的使用和扩展。
- 2、构建一个 $k \times 200 \times 350 \times 500 \times 625$ 的神经网络。(k 为降维后的图片的特征数，即输入神经元的个数，200、300、500 为隐藏层的神经元个数，625 为输出神经元的个数)
- 3、用训练数据训练模型。
- 4、用测试训练集的数据进行分类并计算分类的准确性。

运行

依次执行 main.py, pcatest.py, BPTest.py 程序

运行结果

```
epoch: 24, Train Loss: 0.162084, Train Acc: 0.985511, Eval Loss: 0.350189, Eval Acc: 0.941468
epoch: 25, Train Loss: 0.052091, Train Acc: 0.997159, Eval Loss: 0.297434, Eval Acc: 0.941468
epoch: 26, Train Loss: 0.021640, Train Acc: 0.999716, Eval Loss: 0.286307, Eval Acc: 0.948413
epoch: 27, Train Loss: 0.017152, Train Acc: 1.000000, Eval Loss: 0.301742, Eval Acc: 0.931052
epoch: 28, Train Loss: 0.015740, Train Acc: 1.000000, Eval Loss: 0.309322, Eval Acc: 0.924107
epoch: 29, Train Loss: 0.013492, Train Acc: 1.000000, Eval Loss: 0.278421, Eval Acc: 0.944940
程序运行时间是: 37.09584283828735
```

避坑

文件地址尽量纯英文，避免读取出错

最后，代码太烂了，可读性太差。。。见谅。。。。。