《计算机视觉》实验报告

姓名：戴枫源 学号：19120199

实验四

一. 任务 1

a) 核心代码：

```
import numpy as np
from sklearn.decomposition import PCA
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from PIL import Image
import warnings

warnings.filterwarnings("ignore")



num = 10

def LoadData(): #载入数据集
    data = []
    label = []
    path_cwd = "ORL_dataset/"
    for j in range(1, 41):
        path = path_cwd + 's' + str(j)
```

```
            for number in range(1,num+1):
                path_full = path + '/' + str(number) +'.bmp'
                image = Image.open(path_full).convert('L')
                img = np.array(image)
                data.extend(img)
            label.extend(np.ones(num, dtype=np.int) * j)
        data = np.reshape(data, (num*j, 112*92))
        return np.matrix(data), np.matrix(label).T            #返回数据和标签



def knn(neighbor, traindata, trainlabel, testdata):
        neigh = KNeighborsClassifier(n_neighbors=neighbor)
        neigh.fit(traindata, trainlabel)
        return neigh.predict(testdata)

if __name__ == '__main__':

        # 设置 pca 保留数据方差值和 k
        var,k = 0.80, 5
        Data_train, Data_test, Label_train, Label_test = train_test_split(*LoadData())
        pca = PCA(var, True, True)  # 建立 pca 类，设置参数
        trainDataS = pca.fit_transform(Data_train)  # 拟合并降维训练数据
        # print(len(Data_test))

        acc = 0
        num_test = len(Data_test)

        for i in range(len(Data_test)):
```

```
        testDataS = pca.transform(Data_test[i].ravel())


        result = knn(k,trainDataS,Label_train,testDataS)
        # print("预测:",result[0])
        # print("实际:",int(Label_test[i]))
        if result[0] == int(Label_test[i]):
            acc += 1
    accuracy = float(acc/num_test*100)
    print("var={0},\tk={1}".format(var,k))
    print("accuracy:\t{0}%".format(accuracy))
```

b) 实验结果截图

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.8,    k=5
accuracy:   85.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.7,    k=5
accuracy:   82.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.75,   k=6
accuracy:   85.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.75,   k=5
accuracy:   81.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.85,   k=5
accuracy:   77.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.85,   k=3
accuracy:   82.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.8,    k=1
accuracy:   97.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.85,   k=1
accuracy:   92.0%
```

```
[Running] python -u "c:\Users\DAI FENGYUAN\Desktop\Computational Vision\exp4\exp3.py"
var=0.75,   k=1
accuracy:   92.0%
```

c) 实验小结

本实验用的数据集来自资料中的 ORL 人脸数据库，其中共有 40 个文件夹对应不同的人，每人有 10 张照片。因此总共有 400 张照片，我设置的训练集数和测试集数分别是 300 和 100，分类个数是 40。降维维度和 knn 参数可见上方实验结果。在反复实验过程中我发现保留 80%的数据方差，k=5 时取到的效果较好，准确率可以达到 85%。另外，我发现当 k=1 时，只要保留的数据方差设置的合理，准确率就会异常高，可以达到 90%以上，对此我猜测应该是发生了过拟合。