**人脸识别实验报告**

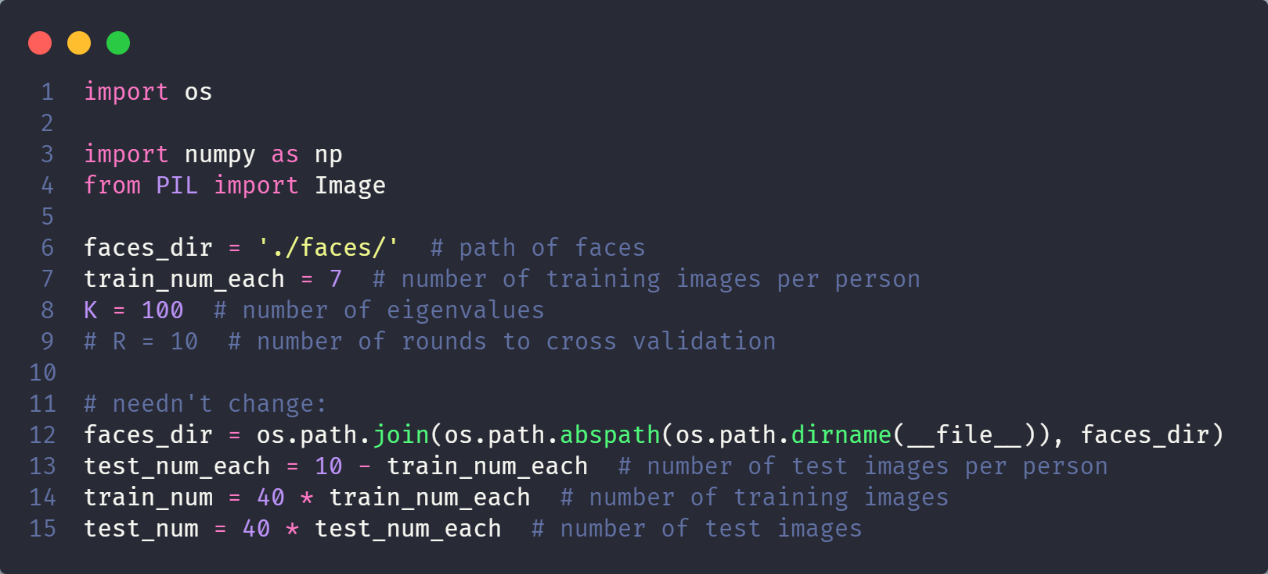
1. **问题描述**

基于ORL数据库，利用K-L变换以及PCA方法，求得特征脸，并对输入的人脸进行识别，输出特征最相似的人脸。

1. **算法描述 & 程序实现**

**算法的具体步骤如下：**

1. 库的导入、初始值的设定（包括数据集路径、训练样本数（每组人脸）、特征数、交叉验证轮数）：



并根据所设定的值计算其他一些需要的值。

1. 读取所有人脸图像，转为向量，并划分为训练集和测试集，加入到不同的矩阵中：



这里在1-10范围内随机取7（所设定的训练样本数）个整数作为训练样本标号，其他的样本划分到测试集。

返回训练集和测试集，矩阵大小见注释。

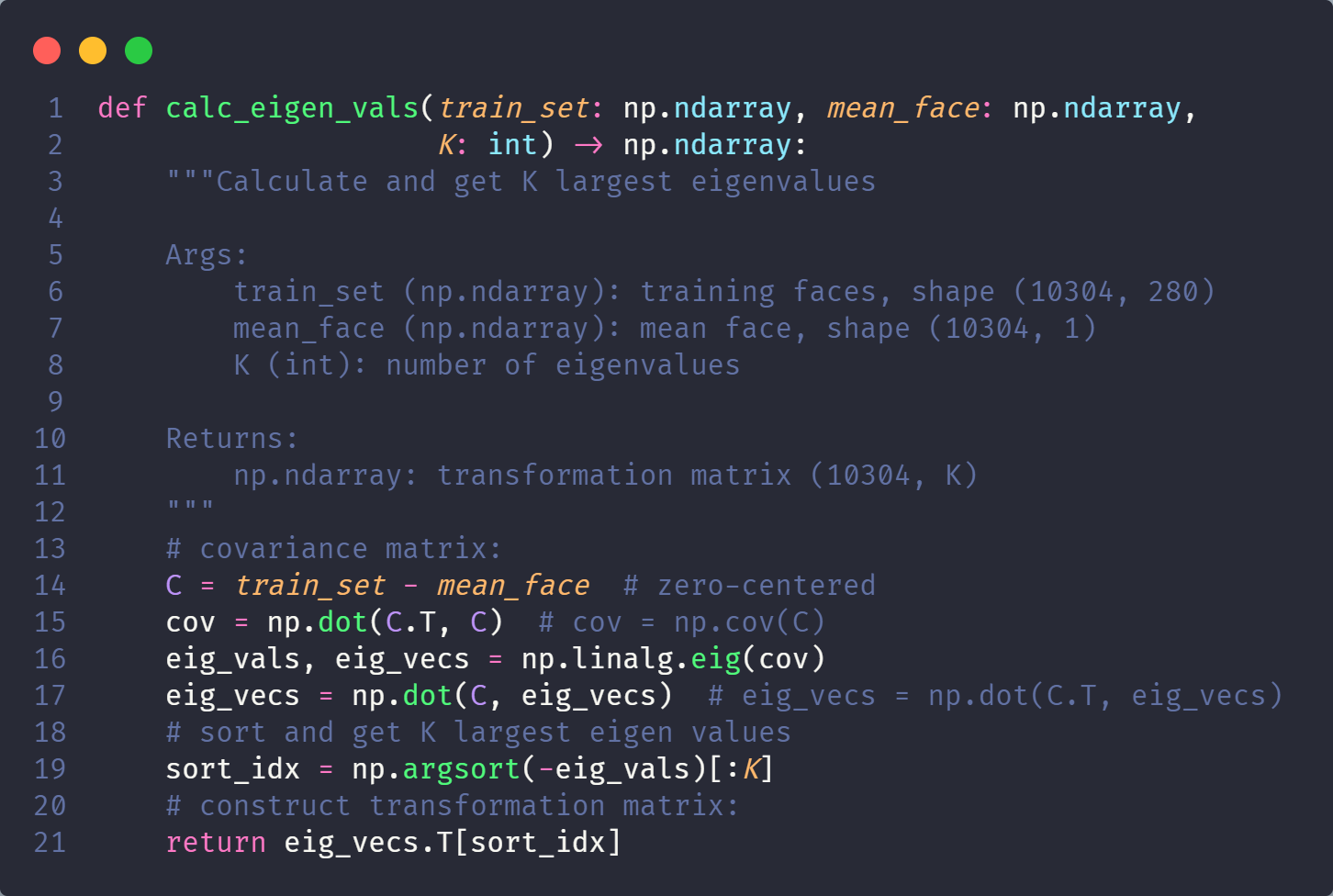
1. 对训练集的每一行计算平均值，得到平均脸：



例如，在一次运行中，得到的平均脸为：

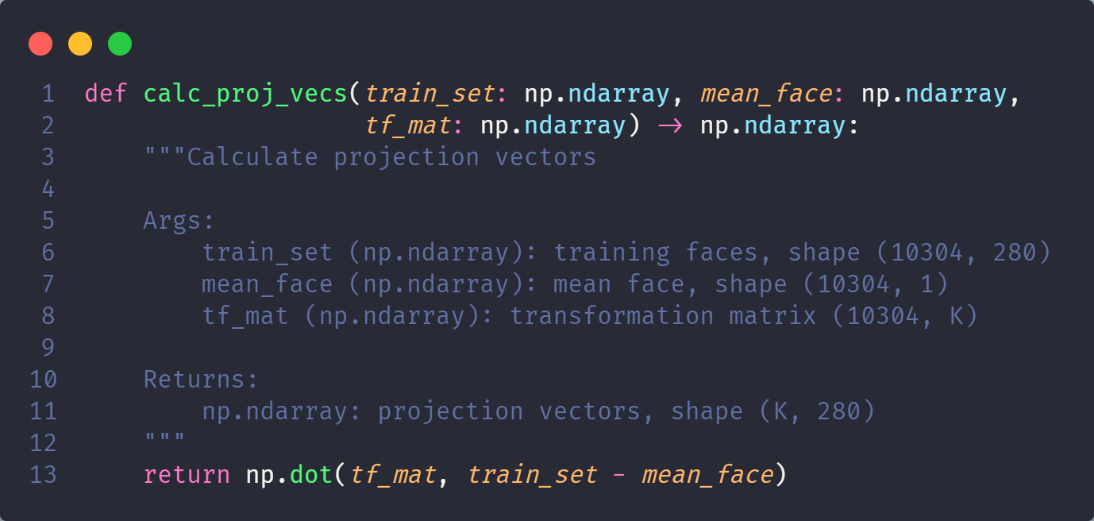


1. 计算训练集的协方差矩阵，计算特征值、特征向量，排序后取K个最大的特征向量，构成变换矩阵：



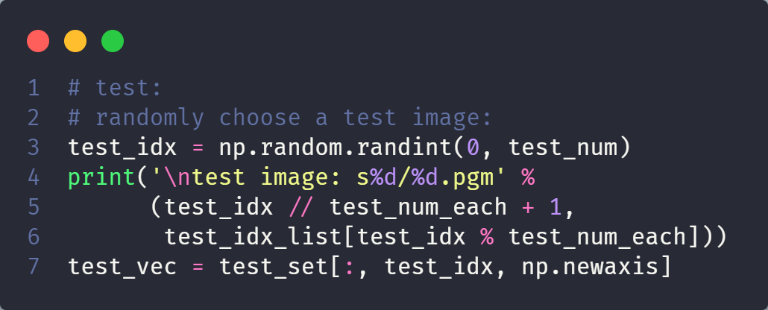
协方差矩阵的计算应为，但过于耗时，采用可以快很多，在后面恢复特征向量时，也要相应地变为用C来乘。

1. 计算训练集中各样本在以上特征向量下的投影，得到查找集：

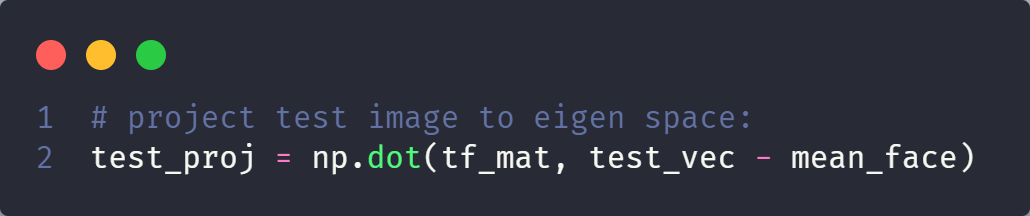


至此，训练已经完成，下面进行测试。

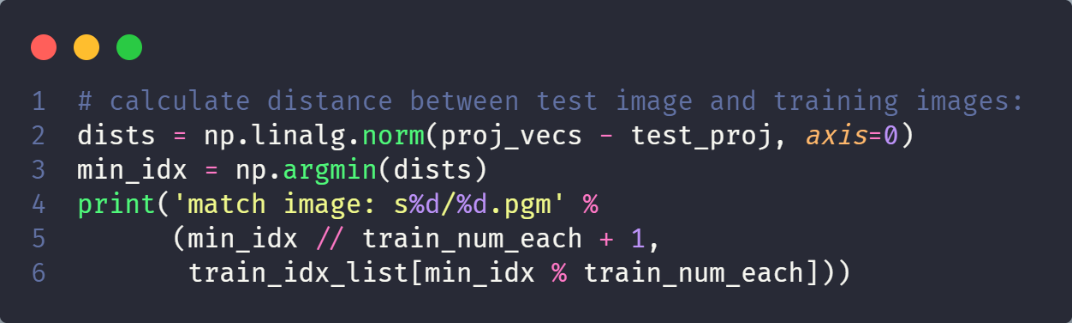
1. 测试：
2. 从前面得到的测试集中随机取一张图片（向量形式）：



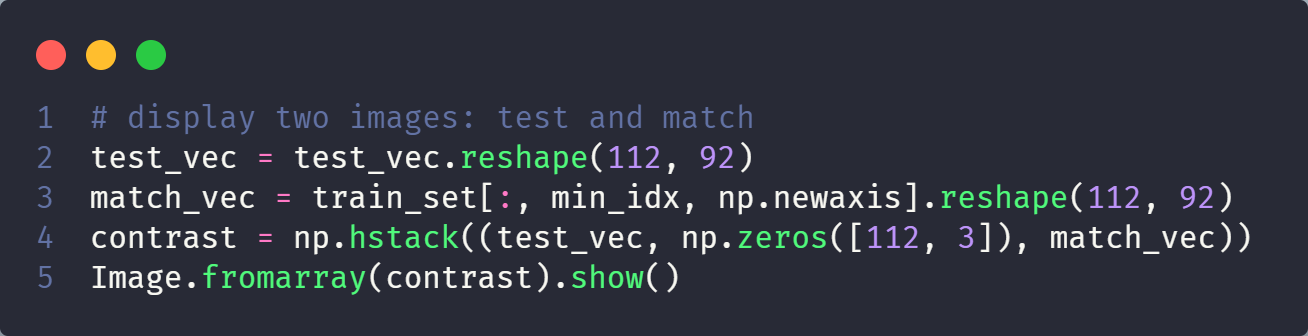
1. 通过变换矩阵，计算在K个特征向量下的投影：



1. 与查找集中的各个向量进行距离计算，找到距离最小的一个，其所属的类（人）即为匹配结果：



1. 将测试图像和匹配图像放在一起打印出来：



1. **计算过程**

以下是其中一次的运行，控制台输出和图像输出：

可以看到是将每个人的第1、3、4、6、7、8、9张图片作为训练集；

测试时取了第23位的第2张图像；

匹配的结果是：第23位的第8张是最接近的，即识别结果是第23位。

training images: [1 3 4 6 7 8 9]

test images: [2 5 10]

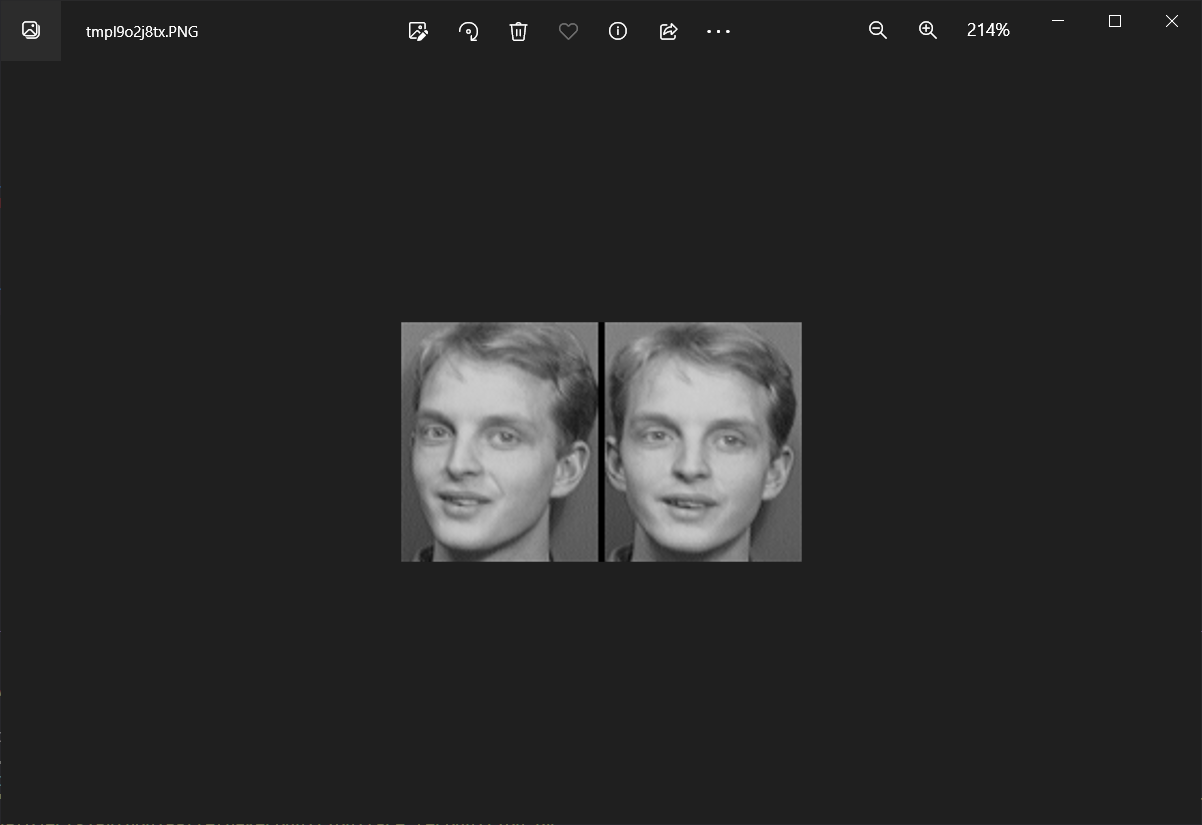
test image: s23/2.pgm

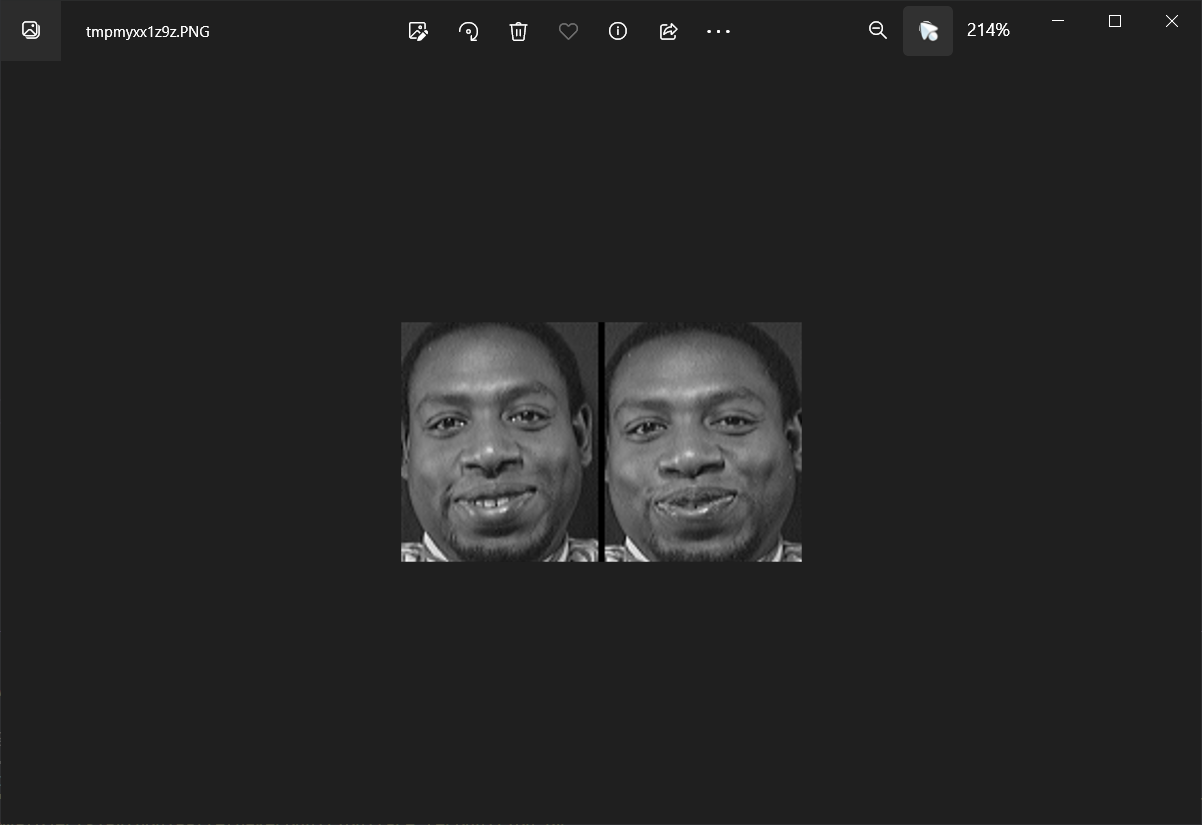
match image: s23/8.pgm



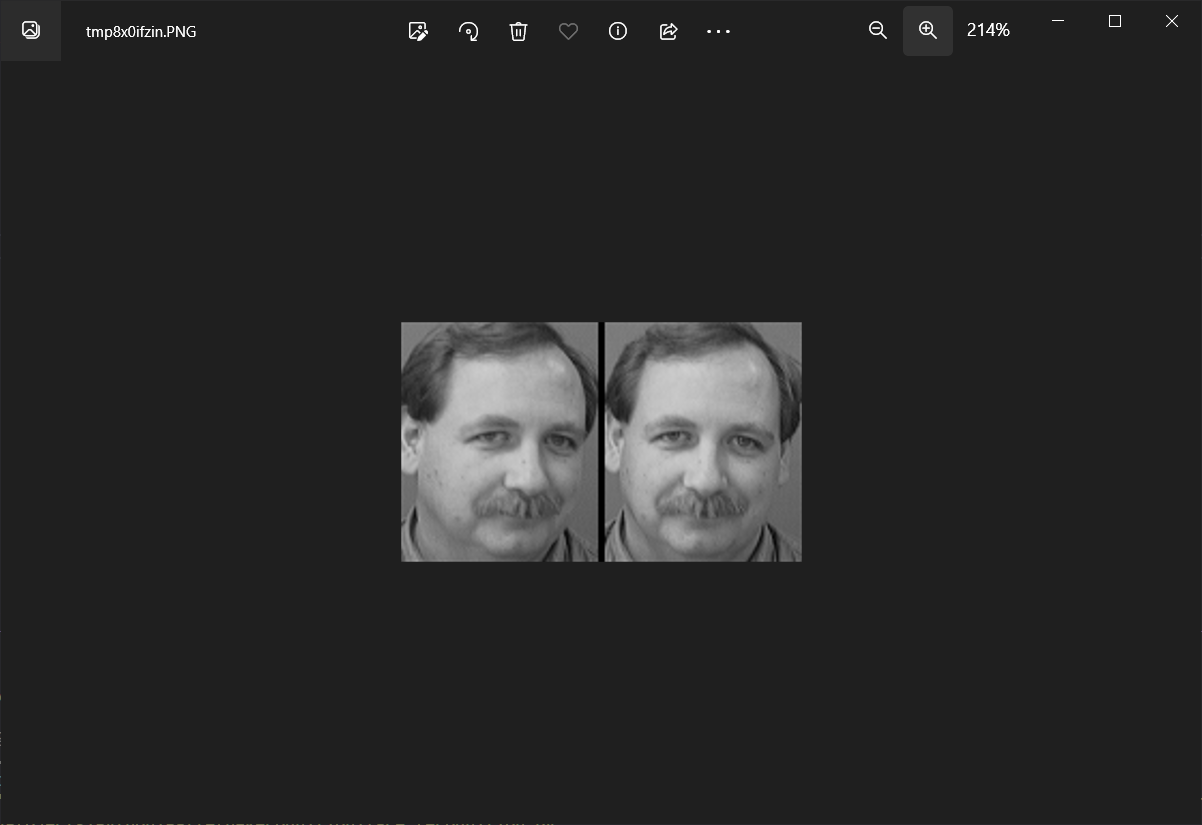
1. **结果分析**

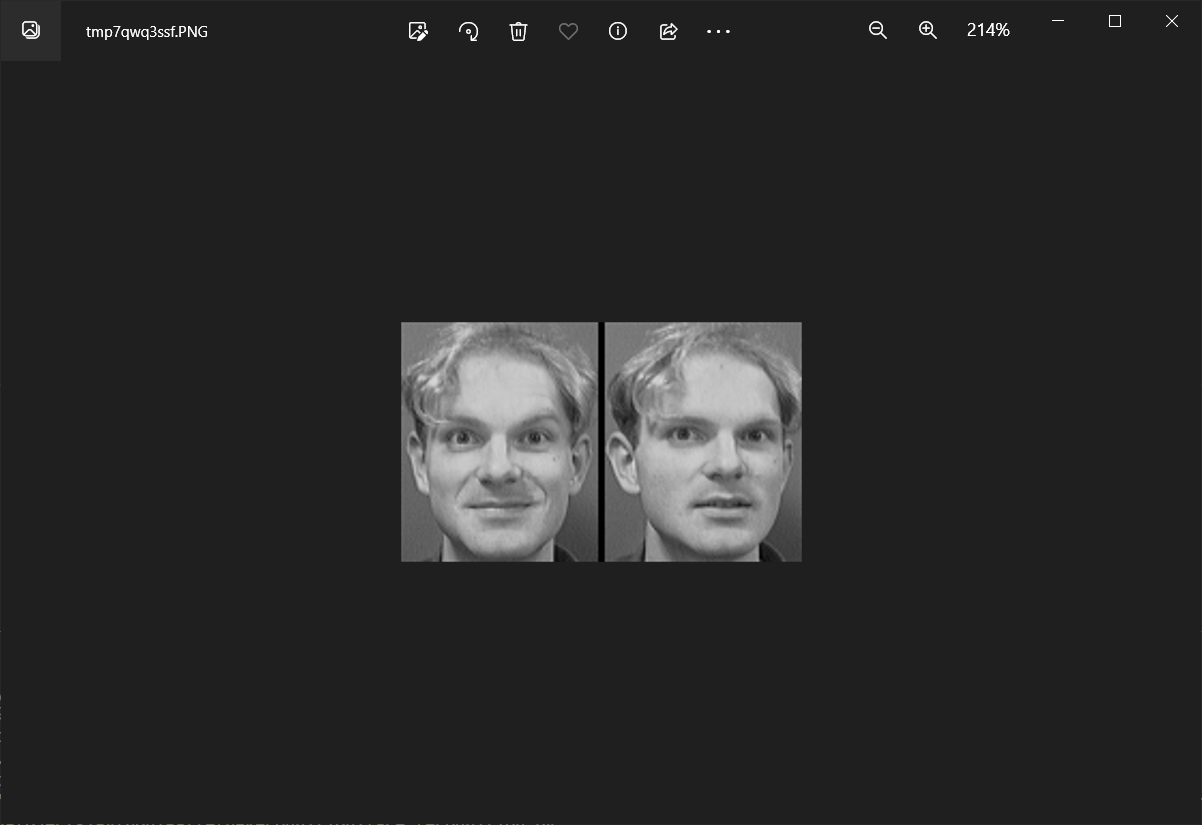
在几组测试中可以看到均能够得到较为相似的匹配结果，且结果基本正确





接下来尝试把特征数量减少，将K设为50，观察匹配的准确率：

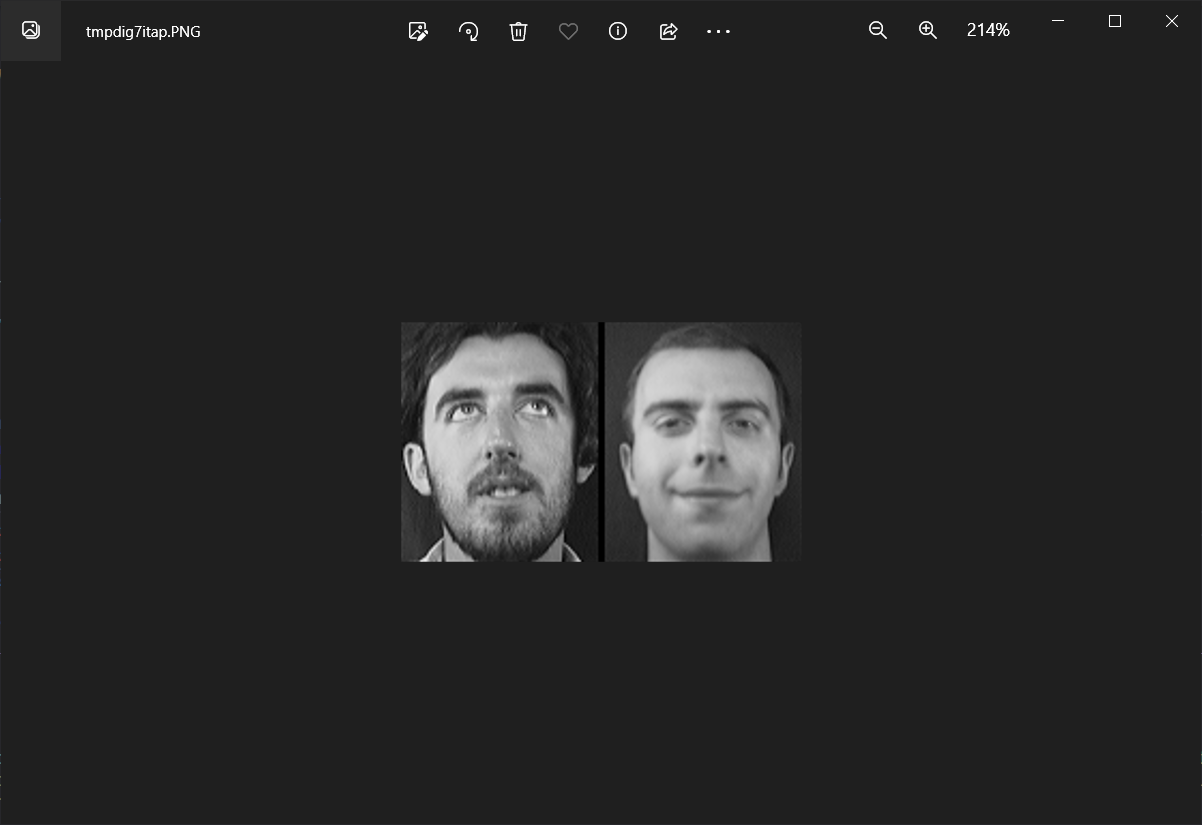




仍能够保持较高的准确率

进一步降低到K=30时：

偶尔会出现错判，比如下面这一结果：



由以上结果可以得到，特征数的取值很大程度影响识别的准确率，较少的特征数会使得误判概率升高；但同时，特征数增多又会带来更大的计算、存储开销（尽管在本实验中，样本数较少影响不大，测试中特征数30与100均在0.31-0.32秒完成）。所以需要多轮交叉验证，评估不同特征数与相应地准确率，以达到最好的效果。