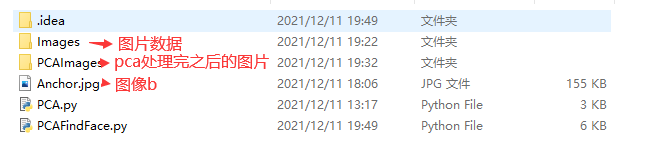
PCA作业代码说明

# 1.代码结构



Images 存放人脸样本集

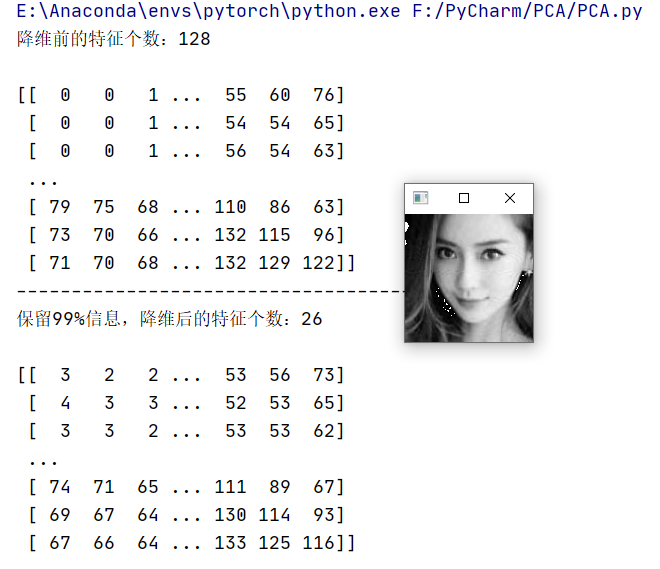
PCAImages 存放PCA处理完之后的图片

Anchor.jpg 是人脸图像b

PCA.py是用于查看pca的降维效果

PCAFindFace.py 是在人脸样本集中找到距离最近的两张人脸

# 2.PCA降维效果



在代码PCA.py中的文件可以看到PCA方法降维效果，

**降维前的特征个数：128，保留99%信息，降维后的特征个数：26**

PCAImages文件夹中保存PCA降维后的图片



# 3.寻找距离最近的两张人脸

采用face\_recognition库，这是一个可以提取人脸特征向量并比较的库。

我们将所有的图片提取特征向量，再与我们的图片b作比较，重写face\_recognition的方法（库中方法不满足我们需求），会得到list距离数组（最大值为1）

[0.7078869182296682, 0.5714152933180368, 0.5371096105029584, 0.5866594852688799, 0.5836994866031222, 0.5694570235851021, 0.6244297863306105, 0.8124810427968264, 0.6473735376678095, 0.6498998624267733, 0.6524312662188423, 0.6380955224933879, 0.7088677025391611, 0.6016642122944855, 0.7782523657906301, 0.7253884226678528, 0.6431344071232882, 0.6067376426020884, 0.7297929619174705, 0.709472965625184, 0.6568574965452935, 0.6295006258957168, 0.6180526584713102, 0.6198983296942037, 0.567107887562961, 0.6648319448296622, 0.5941856296770884, 0.6995222223863892, 0.5489622089916928, 0.6422238052154244, 0.5369004236856058, 0.6073029096365177, 0.5066910919336096, 0.6638114677107612, 0.6863644170023746, 0.6014466961825728, 0.6576655802678792, 0.6545980087634113, 0.7028240232787908, 0.582145580924138, 0.6405926520160143, 0.6093654942746449, 0.6430010069819747, 0.673809509292481, 0.6308839519916428, 0.6482696885526424, 0.7029934713043173, 0.6488135242056052, 0.6892293971628664, 0.704335908628576, 0.5826805775123959, 0.6981388662233363, 0.5579999610115058, 0.5905217160863343, 0.6204663551458423, 0.651224973459566, 0.5341784227953864, 0.5982473480975855, 0.5485094464969584, 0.6312753990183497, 0.6091100517569094, 0.4763351689401784, 0.7424119874331697, 0.520919020958796, 0.35083905922556946]

这是一个65个元素的一维数组，分别代表图像b与其他65张图片的距离。我们遍历两边找出第一小和第二小的下标就能找到本数据集中距离最近的两张人脸。

在后面提取人脸特征的时候，采用特征提取速度也加快

原图特征提取速度为39.4s：



使用PCA降维后特征提取速度37.9s



结果如下：

