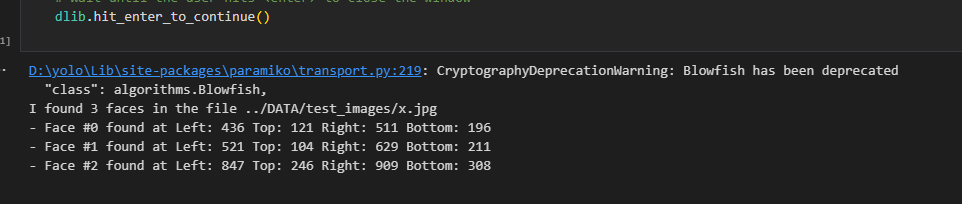
**任务1：根据课上演示的Demo，使用与演示时不同的图片完成以下任务，截图运行成功的代码和所有实验结果记录在本报告中，并保存实验结果到指定的文件夹。该任务占大作业成绩的30%。**

1. Face detection using HOG，将实验结果图像和生成的实验数据（.csv格式）保存在以“1- Face detection using HOG”的新建文件夹中；



图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 表格, Excel

描述已自动生成

1. Face landmark detection using HOG，将实验结果图像和生成的实验数据（.csv格式）保存在以“2- Face landmark detection using HOG”的新建文件夹中；

屏幕上有字

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

截图里有照片

描述已自动生成

1. 使用OpenFace程序处理自选3张图像，其中第1张图像（文件名1face.xxx）包含1张人脸，第2张图像（文件名2face.xxx）包含2张人脸，第3张图像（文件名multi-face.xxx）包含超过10张可识别人脸。要求额外设置程序保存HOG和对齐（aligned）的人脸数据。分别用图像名新建文件夹，将生成的实验结果分别放在对应的文件夹中。将三个文件夹保存在以“3- OpenFaceApp”的新建文件夹中；

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

1. Face detection using Face Recognition Algorithm，将实验结果图像和生成的实验数据（.csv格式）保存在以“4- Face detection using Face Recognition Algorithm”的新建文件夹中；

图形用户界面, 应用程序

描述已自动生成

表格

描述已自动生成

1. Face feature marking using Face Recognition Algorithm，将2种实验结果图像和生成的实验数据（.csv格式）分别保存在以“5- Face feature marking using Face Recognition Algorithm”的新建文件夹中；

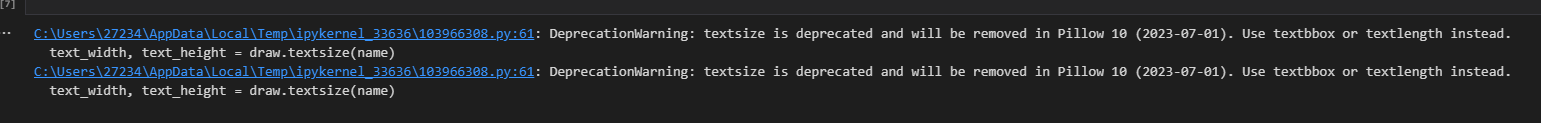
文本

描述已自动生成

图形用户界面, 应用程序, Teams

描述已自动生成

1. Identifying people and drawing name boxes on faces using Face Recognition Algorithm，将实验结果图像保存在以“6- Identifying people and drawing name boxes on faces using Face Recognition Algorithm”的新建文件夹中；



图形用户界面, 应用程序

描述已自动生成

1. Face makeup using Face Recognition Algorithm，将实验结果图像保存在以“7-Face makeup using Face Recognition Algorithm”的新建文件夹中；

图形用户界面, 应用程序

描述已自动生成

1. Face matching using Face Recognition Algorithm，将实验结果数据（.csv格式）保存在以“8- Face matching using Face Recognition Algorithm”的新建文件夹中；

文本, 日历

中度可信度描述已自动生成 文本

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Face distance calculation using Face Recognition Algorithm，将实验结果数据（.csv格式）保存在以“9- Face distance calculation using Face Recognition Algorithm”的新建文件夹中；

文本

描述已自动生成

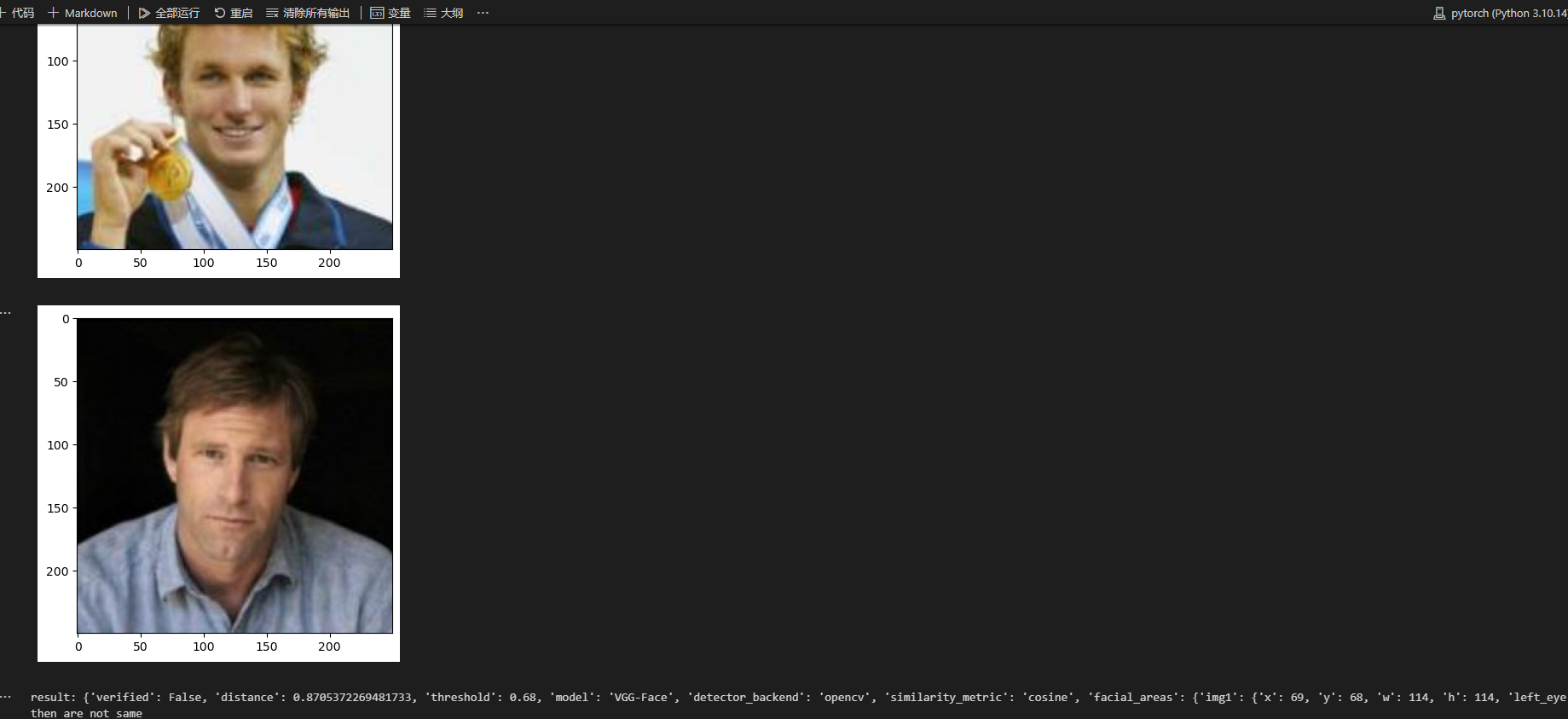
图形用户界面, 文本, 应用程序

描述已自动生成

1. Face distance calculation using DeepFace Algorithm，将2种实验模式（仅显示结果数据信息和显示图像及结果数据信息）的2种匹配形式（匹配和不匹配）的数据（.csv格式）分别保存在以“10-Face distance calculation using DeepFace Algorithm”的新建文件夹中；

文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

文本

描述已自动生成

1. Finding a target face from a dataset using DeepFace Algorithm，将2种实验模式（同一个人及不同人）的结果数据（.csv格式）保存在以“11-Finding a target face from a dataset using DeepFace Algorithm”的新建文件夹中；

文本

描述已自动生成 

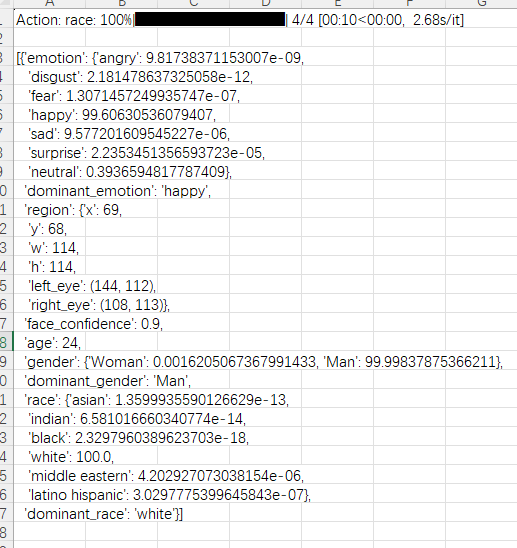
1. Analysing face attributes using DeepFace Algorithm，挑选4张代表不同表情、年龄、性别、种族的人脸图像进行实验，将4次实验的使用图像和结果数据（.csv格式）保存在以“12-Analysing face attributes using DeepFace Algorithm”的新建文件夹中。

文本

描述已自动生成

图形用户界面, 文本, 应用程序

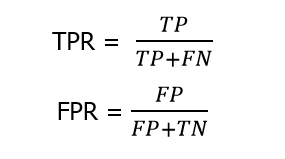
描述已自动生成



**任务2：分别使用Face Recognition和DeepFace两个算法对Datasets中的5个数据库分别进行All2All的人脸识别匹配并计算匹配分（distance）。将识别匹配分保存成.csv文件（例如：1-Yale-face-dataset.csv），其中每行数据格式是“人脸路径（名称）1，人脸路径（名称）2，匹配分（distance）”。对于生成的匹配分文件进行Genuine比较和Imposter比较的得分划分，生成2个新的.csv文件（例如：1-Yale-face-dataset\_genuine.csv和1-Yale-face-dataset\_imposter.csv）。最后将上述两个.csv文件里的前两列图像路径删除仅保留一列得分，生成新的两个.csv文件（例如：1-Yale-face-dataset\_genuine\_scores.csv和1-Yale-face-dataset\_imposter\_scores.csv）以便使用pyeer进行人脸识别算法性能测试。以数据库文件夹同名的方式新建文件夹存放利用pyeer生成的不同数据库的结果数据。最后将不同数据库的文件夹保存在以“13-Performance Evaluation”的新建文件夹中。根据生成的使用两个人脸识别算法在不同数据库上的性能评估结果，在本报告中进行总结性报告，简要总结两个算法在不同数据库上的性能。**

1. **结果分析原理：**

1.根据学习器的预测值所产生的排序的好坏，体现了综合考虑学习器在不同任务下的“期望泛化性能”的好坏。ROC曲线，全称为Receiver Operating Characteristic （受试者工作特征）曲线 ，它的纵轴为真正例率TPR(True Positive Rate),横轴为假正例率FPR(False Positive Rate)，分别定义为：



2.真正例率就是学习器认为的正例占所有正例的比率，也就是我们通常所说的查全率或者召回率。假正例率就是在所有的负例中学习器认为是正例的比率。此外，再引入两个常用的错误率度量，分别是错误接受率FAR(False Acceptance Rate）和错误拒绝率FRR(False Reject Rate),分别定义为：

文本

描述已自动生成

3.错误接受率就是所有的负例里面被学习器误认为是正例的比率 ，错误拒绝率就是所有的正例里面被学习器误认为是负例的比率。我们可以很容易观察到FAR = FPR, FRR = 1 - TPR。

如果没有特别指明FMR和FNMR，则：FMR = FAR ，FNMR = FRR4.从AUC判断分类器（预测模型）优劣的标准： · AUC = 1，是完美分类器，采用这个预测模型时，存在至少一个阈值能得出完美预测。绝大多数预测的场合，不存在完美分类器。 · 0.5 < AUC < 1，优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。· AUC = 0.5，跟随机猜测一样（例：丢铜板），模型没有预测价值。· AUC < 0.5，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。总的来说，AUC值越大，模型的分类效果越好，人脸检测越准确；不过两个模型AUC值相等并不代表模型效果相同。

AUC(area under thecurve)，也就是ROC曲线的下夹面积，越大说明分类器越好，最大值是1。

5. EER（平均错误概率）是一种生物识别安全系统算法，用于预先确定其错误接受率及其错误拒绝率的阈值。当速率相等时，公共值称为相等错误率。该值表明错误接受的比例等于错误拒绝的比例。等错误率值越低，生物识别系统的准确度越高。我们在ROC曲线上将坐标点（0，1）和（1，0）连接起来，与ROC曲线的交点所对应的FPR值就是EER 。通过上述EER的定义我们可以发现：EER是FPR = 1 - TPR时的FPR值，FPR其实就是FAR,而FRR= 1 - TPR。所以EER又是FRR与FAR相等时的值，即错误接受率与错误拒绝率相等时的值，所以称之为等错误率即Equal Error Rate。

**参考资料：**

[深入理解ROC曲线和EER及其之间的关系\_eer与orr的区别-CSDN博客](https://blog.csdn.net/lcdxshengpeng/article/details/88950587)

[[深度概念]·评估指标EER(Equal Error Rate)介绍-CSDN博客](https://blog.csdn.net/xiaosongshine/article/details/89037068)

[什么是ROC曲线？为什么要使用ROC?以及 AUC的计算-腾讯云开发者社区-腾讯云 (tencent.com)](https://cloud.tencent.com/developer/article/1747389)

[西瓜书-性能度量-腾讯云开发者社区-腾讯云 (tencent.com)](https://cloud.tencent.com/developer/article/1532296)

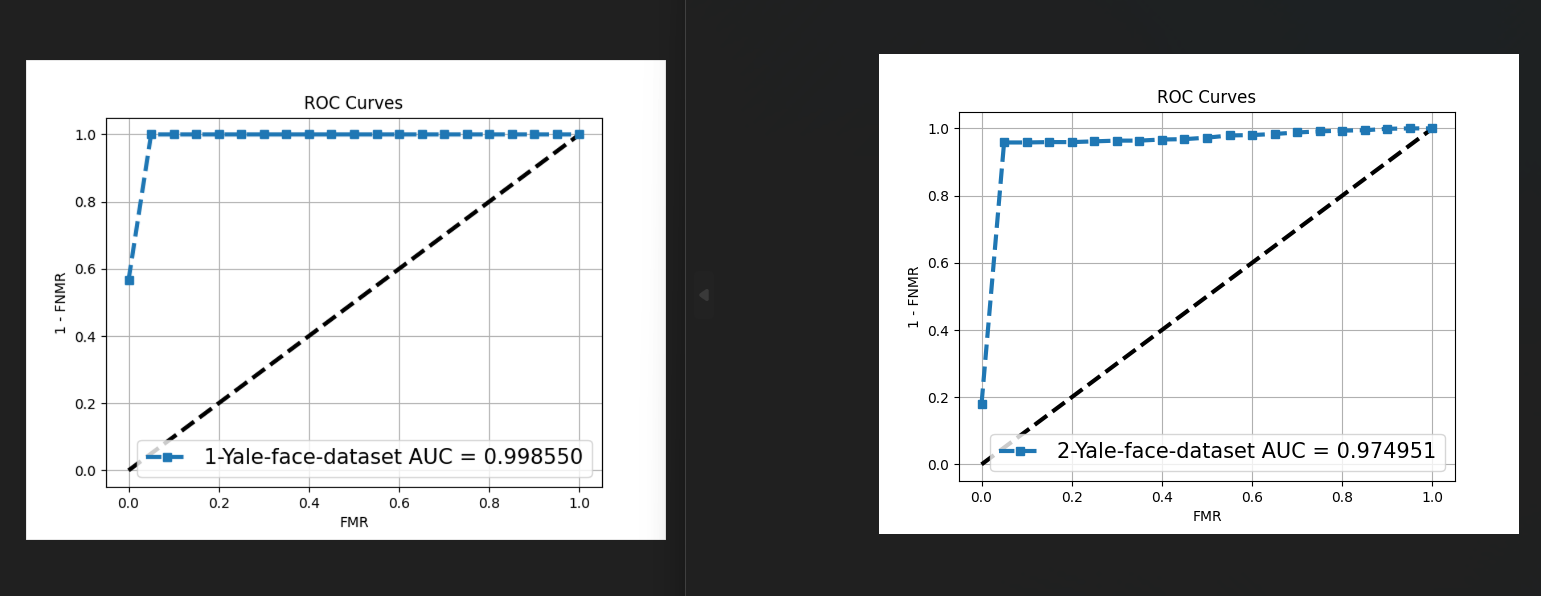
[模型评估指标AUC和ROC，这是我看到的最透彻的讲解-腾讯云开发者社区-腾讯云 (tencent.com)](https://cloud.tencent.com/developer/article/1508882)

[【数据库基础】EER图（Enhanced Entity-Relationship Modelling）-CSDN博客](https://blog.csdn.net/YFY20020109/article/details/125331603)

1. **结果分析实例：（1开头的是face recognition算法，2是deep face算法）**

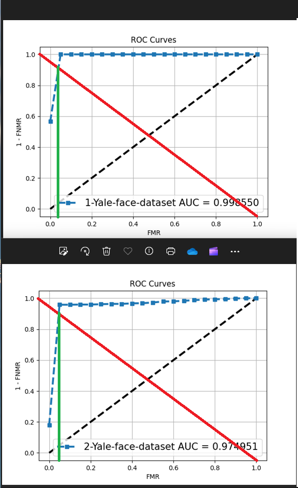
接下来，针对两个算法的评估主要从ROC曲线和EER得分这两个评估角度出发

1.Yale-face-dataset



Face Recognition ROC曲线 Deepface ROC曲线

可观察到，Face Recognition的ROC曲线图中的AUC大于Deep Face的，性能评估结果更优。



再通过EER观察，可看到Face Recognition（0.020345）略低于Deep Face（0.041873），人脸识别算法的准确度略高。

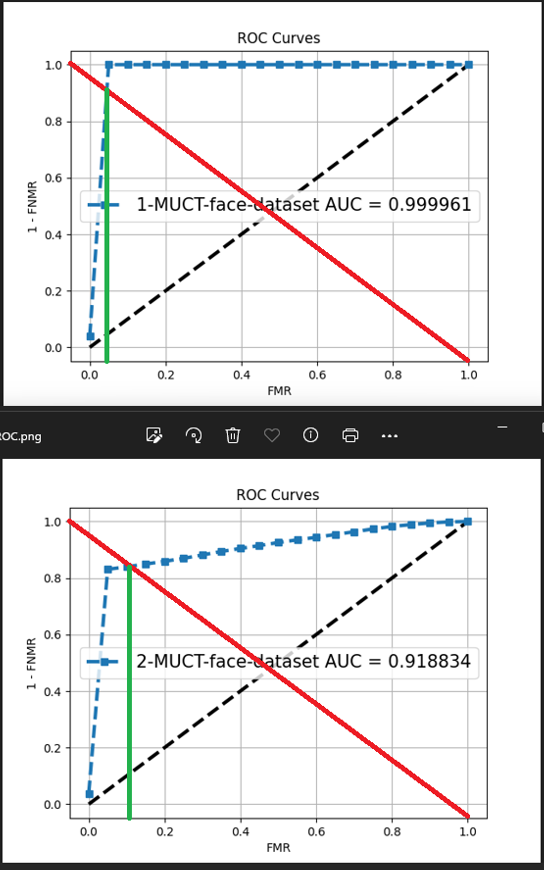
2.MUCT-face-dataset

图表, 折线图

描述已自动生成

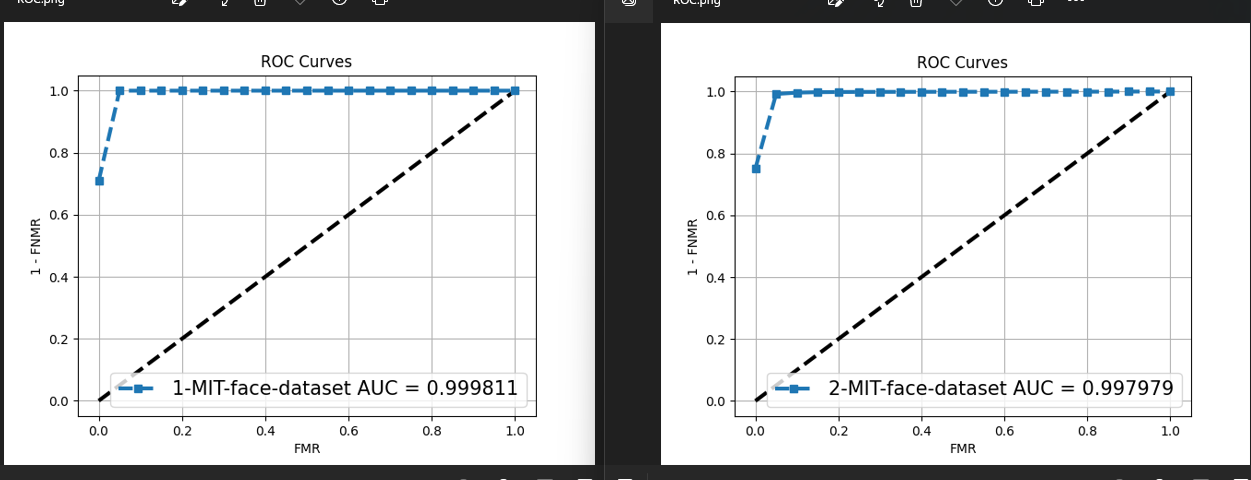
Face Recognition ROC曲线 Deepface ROC曲线

可观察到，Face Recognition的ROC曲线图中的AUC大于Deep Face的，性能评估结果更优。



再通过EER观察，可看到Face Recognition（0.002029）低于Deep Face（0.150491），人脸识别算法的准确度更高。

3. MIT-face-dataset

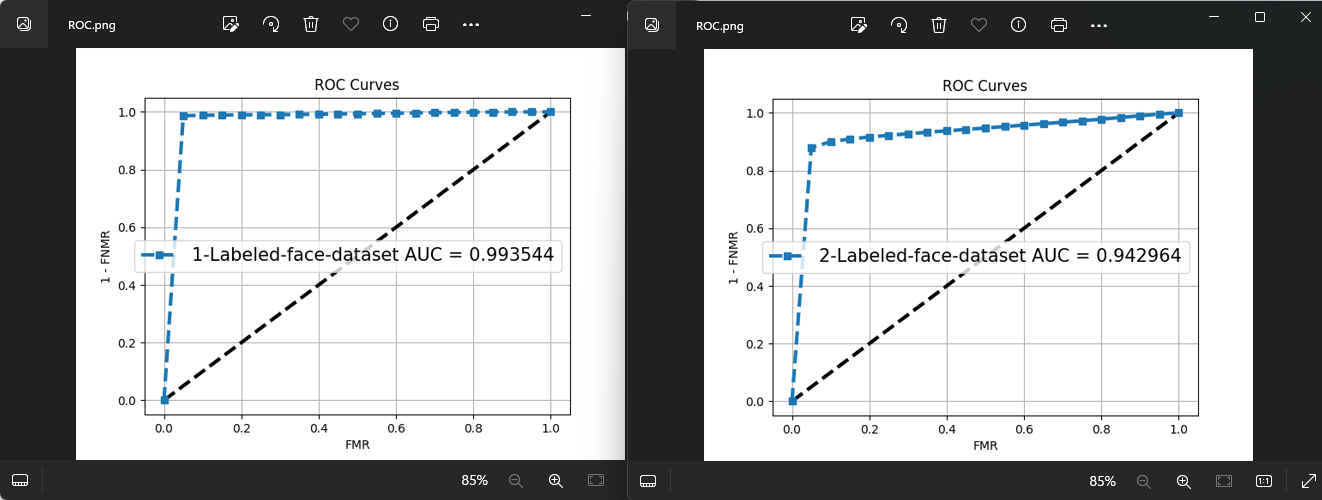


Face Recognition ROC曲线 Deepface ROC曲线

可观察到，Face Recognition的ROC曲线图中的AUC大于Deep Face的，性能评估结果更优。

EER同上，Face Recognition（0.006813）略优于Deep Face（0.017196），不作过多赘述。

4. Labeled-face-dataset

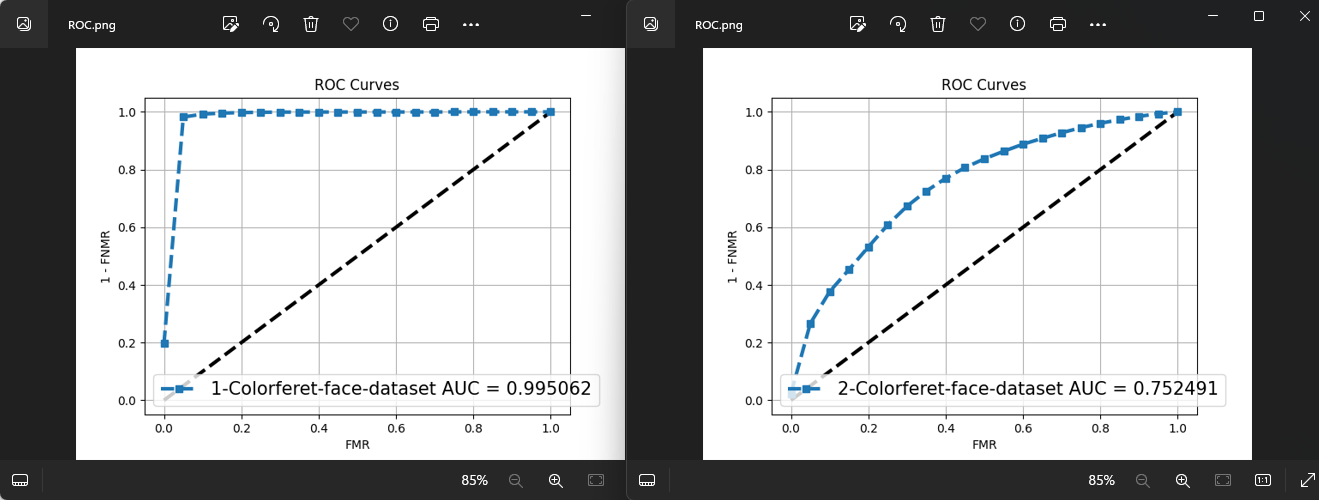


Face Recognition ROC曲线 Deepface ROC曲线

可观察到，Face Recognition的ROC曲线图中的AUC大于Deep Face的，性能评估结果更优。

EER同上，Face Recognition（0.015578）略优于Deep Face（0.100164），不作过多赘述。

5. Colorferet-face-dataset

 Face Recognition ROC曲线 Deepface ROC曲线

同理观察5数据库可视化结果，Face Recognition算法性能都优于Deep Face。

总结可得出，Face recognition算法在以上五个数据库上的性能比DeepFace更准确。

同时，在程序运行中。Face recognition 算法的时间开销要远小于Deep Face

结合AUC和EER得分，综合评价出在这五个数据集中，Face recognition算法的准确率与效率要优于Deepface

总结：

在程序编写过程中，遇到的了许多的问题。首先是对数据集原始文件如何处理的问题。在YALE数据集中，数据集为特殊后缀的GIF文件，但是对于deepface算法，无法直接读取gif，如果以其他图像文件格式读取会损坏图片数据。解决办法是读取图片后进行通道转换（强制转换成三通道格式）。同时，在对处理较大数据集时，串行运行时间为58小时（13000张待处理文件）。由于python的GIL锁限制，python3.9 难以实现真正的多线程。只能利用多进程multiprocessing 来实现‘虚假的’多进程。但是，在利用多线程进行运算之后，显存和内存的需求超过了现有的硬件（12G显存 48G内存），只能减少线程数，最后经过测试，四个线程能达到设备的最大负载。最后，在导出eer结果时，同样遇到了内存不足的情况，由于Labeled 数据集的 genuine 和 imposter 结果文件都达到了 21GB,在实际运行过程中，会因内存耗尽而终止运行，解决办法为优化代码，将读取结果文件与eer运算分开为两个自定义函数，及时销毁无用的数据，避免设置不必要的全局变量。同时，减少结果distance 的数据精度也可以减少文件占用。

由AUC图像与EER得分，我们可以看到face recognition 算法的精度要由于 deepface 。但是，后者算法出现时间要晚于前者，运算时间也远比前者长，但是却得到了差于face recognition 算法的识别准确度，这一点令人疑惑。