系统名称：

小组成员：段慧敏、杨语朦、许可、王豪琴

一、系统概述

面向用户什么功能：实现用户和哪位明星更相似的匹配功能。

用户打开网页，可以选择本地照片，或者现场拍摄照片，照片与数据库中明星照片进行匹配，展示出相似度最高的照片。

二、系统原型设计

原型设计工具：figma

界面，每个界面怎么交互（输入输出、交互反馈），什么约束，界面之间怎么链接的（什么操作）

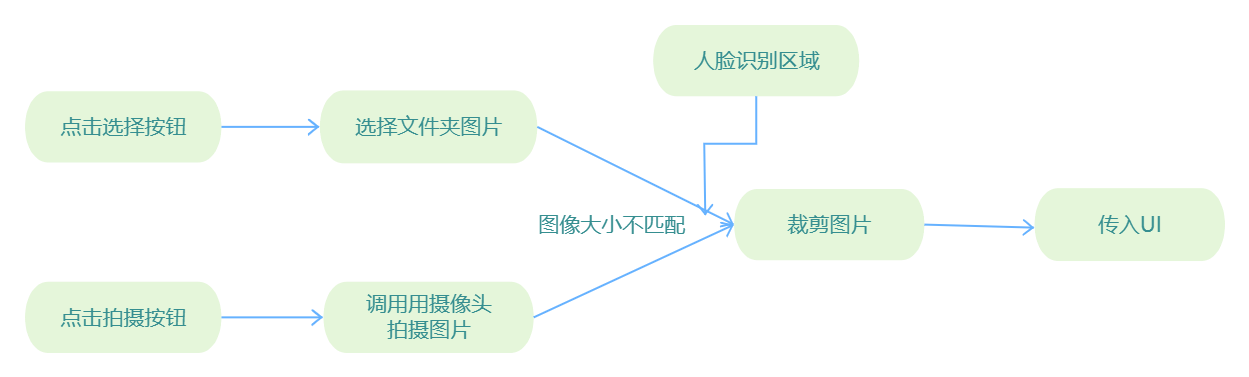
界面整体布局如下：



基本流程

1. 从③选择上传照片方式，

* 选择“拍摄照片”，系统调用摄像头，点击键盘‘s’键保存拍摄照片，点击‘p’键撤销摄像头；
* 选择“选择照片”，系统打开本地文件夹，选择‘.JPG’格式照片，



1. 上传后的照片经过人脸识别对齐，在②中显示
2. 点击**④**“开始匹配”，系统开始自动将用户选中照片与数据库中所有照片进行匹配
3. 相似度最高的照片在①中显示
4. 匹配完成的文本输出结果在**⑤**中显示

三、系统设计

*多少类（界面类、方法调包）、每个类的属性（名称、数据类型、含义、特别说明）、类的方法（名称、传入参数数据类型和含义、返回数据类型和含义）、方法内部处理的程序流程图（面向过程的那些）*

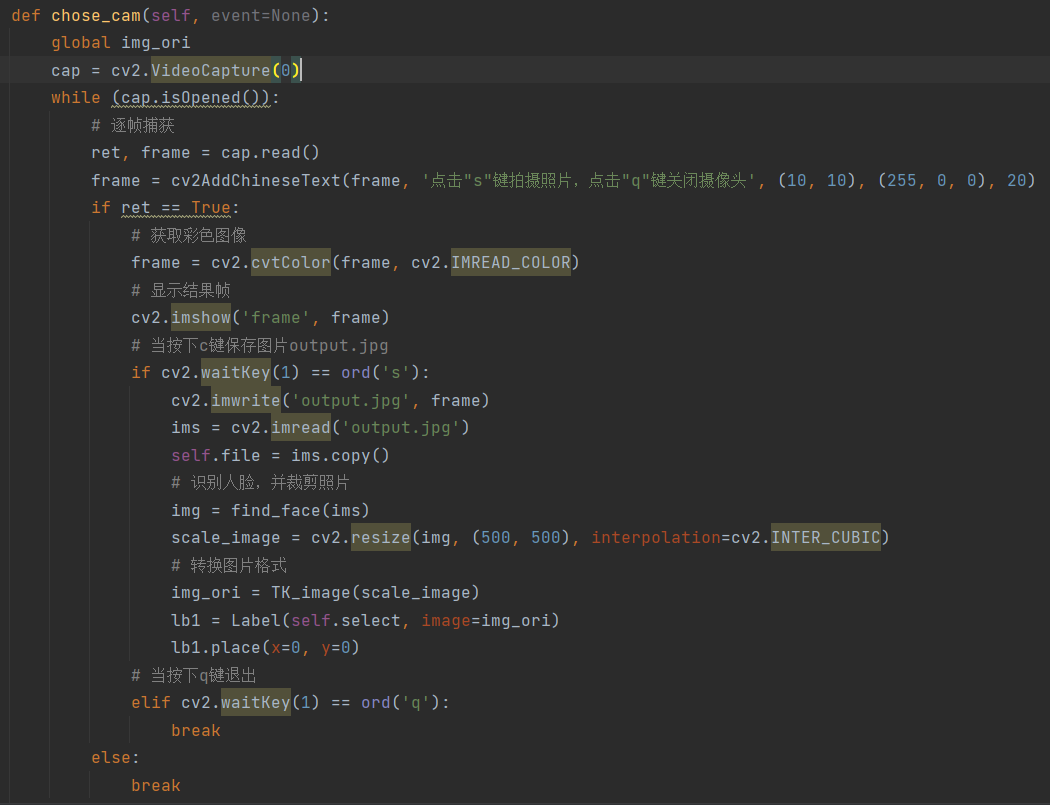
一个类：GUI，封装了整个UI布局，以及按钮控件调转的函数

* 类内方法：

方法1：点击选择按钮选择文件，调用chose\_file函数，将选择的照片放入UI中



方法2：点击拍照按钮，调用chose\_cam拍摄照片，将选择的照片放入UI中



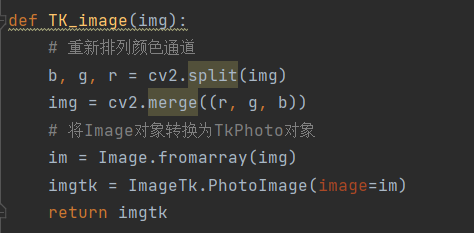
方法3：点击开始匹配，调用matching函数，将所选照片与数据库中进行匹配，输出最相似的照片

############# 截 def matching 函数 ###############

* 类外方法：

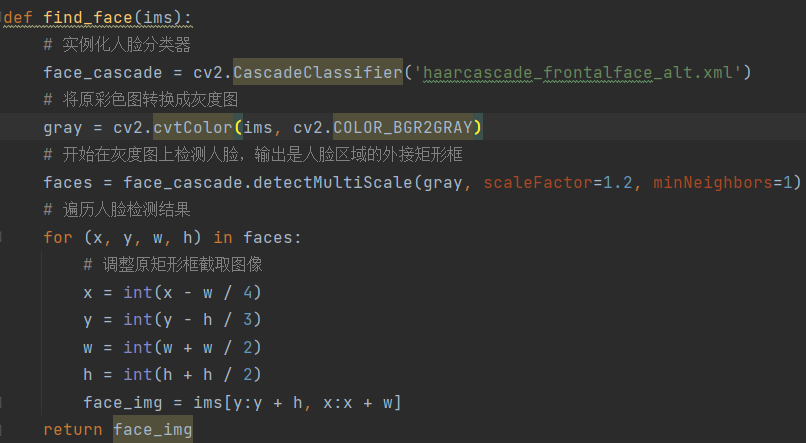
方法1：TK\_image（），转换opencv图片格式，让其可以被photoImage读取

输入opencv的（b，g，r）格式图片，输出（r,g,b）格式图片



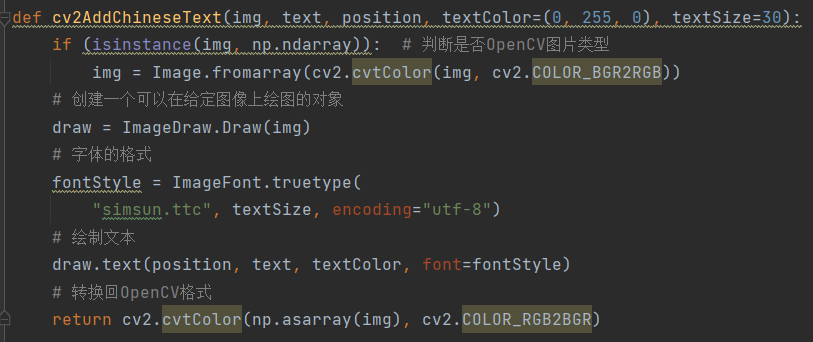
方法2：find\_face() 截取图片中的人脸部分

输入含有人脸的opencv格式图片，输出截取后图片中的人脸部分



方法3：cv2AddChineseText（），改写函数，使opencv格式图片可以写上汉字

输入原始图片和一段有汉字的文字，输出带有这些文字的图片

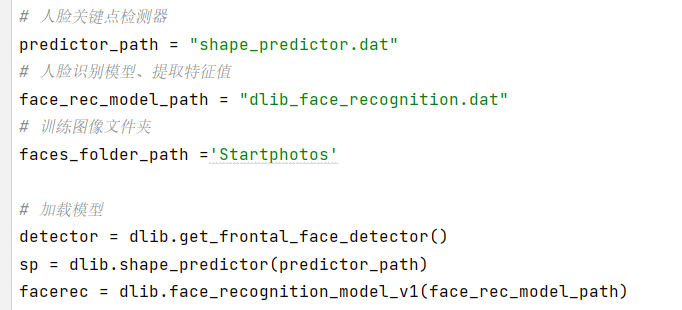


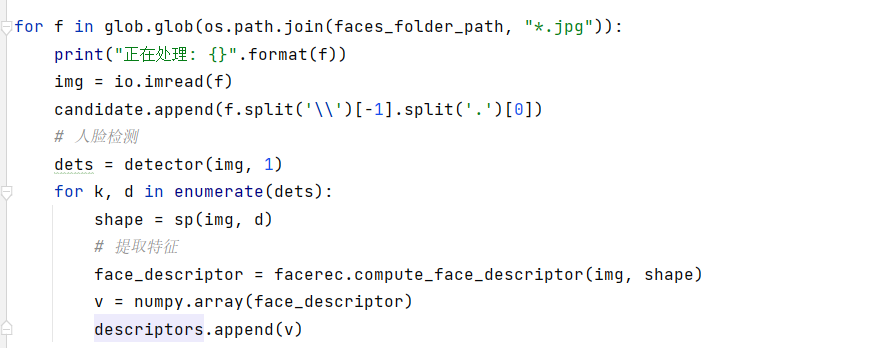
方法4：############你的函数说明，以及截图############

*测试误判率、实际运行结果样子*

############### 运行结果截图，可以从视频里面截 ################

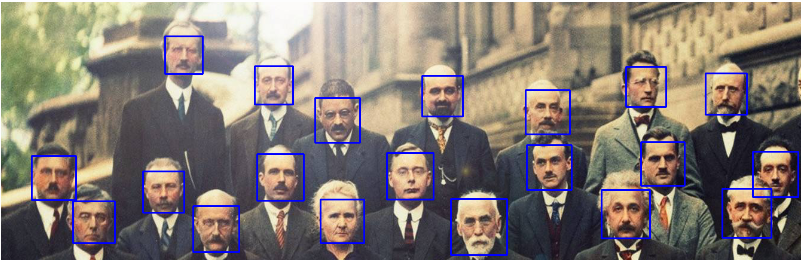
相似度检测：



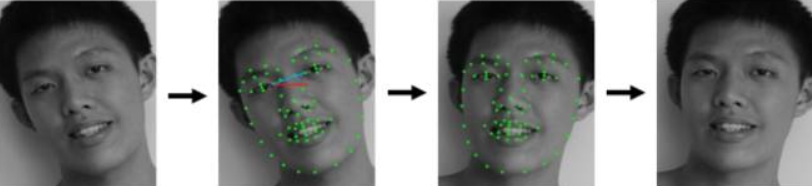
加载模型，对每张图进行人脸提取、关键点提取、特征检测，将得到的特征向量全部存入一个列表：

输入一张图片，做上述重复操作得到该图特征向量，然后与训练好的列表数据一一比较欧氏距离，得到欧氏距离最小的则为最相似的图。

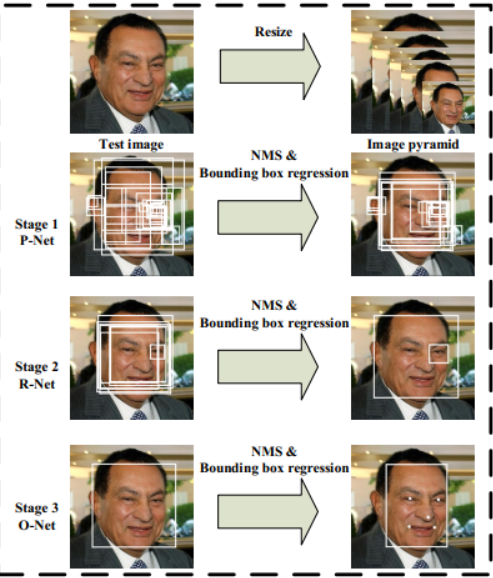
搭建人脸识别系统的第一步就是人脸检测，也就是在图片中找到人脸的位置。在这个过程中输入的是一张含有人脸的图像，输出的是所有人脸的矩形框。一般来说，人脸检测应该能够检测出图像中的所有人脸，不能有漏检，更不能有错检。



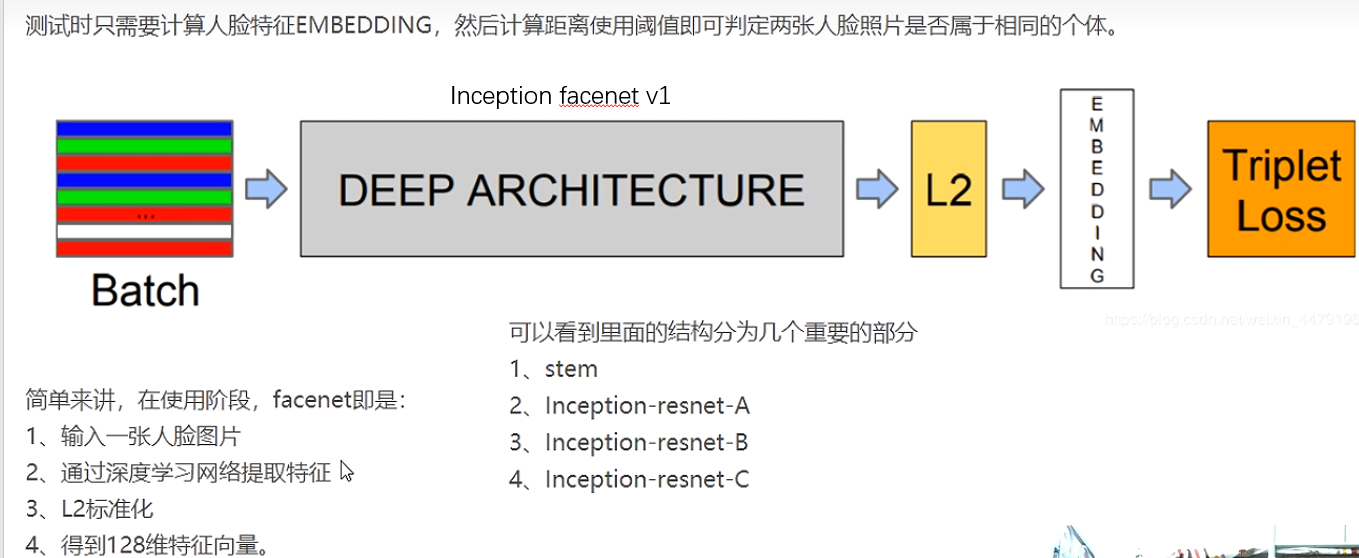
获得人脸之后，第二步我们要做的工作就是人脸对齐，由于原始图像中的人脸可能存在姿态、位置上的差异，为了之后的统一处理，我们要把人脸“摆正”。为此，需要检测人脸中的关键点，比如眼睛的位置、鼻子的位置、嘴巴的位置、脸的轮廓点等。根据这些关键点可以使用仿射变换将人脸统一校准，以消除姿势不同带来的误差。



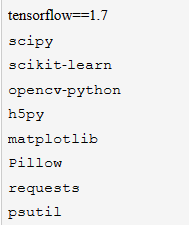
MTCNN：

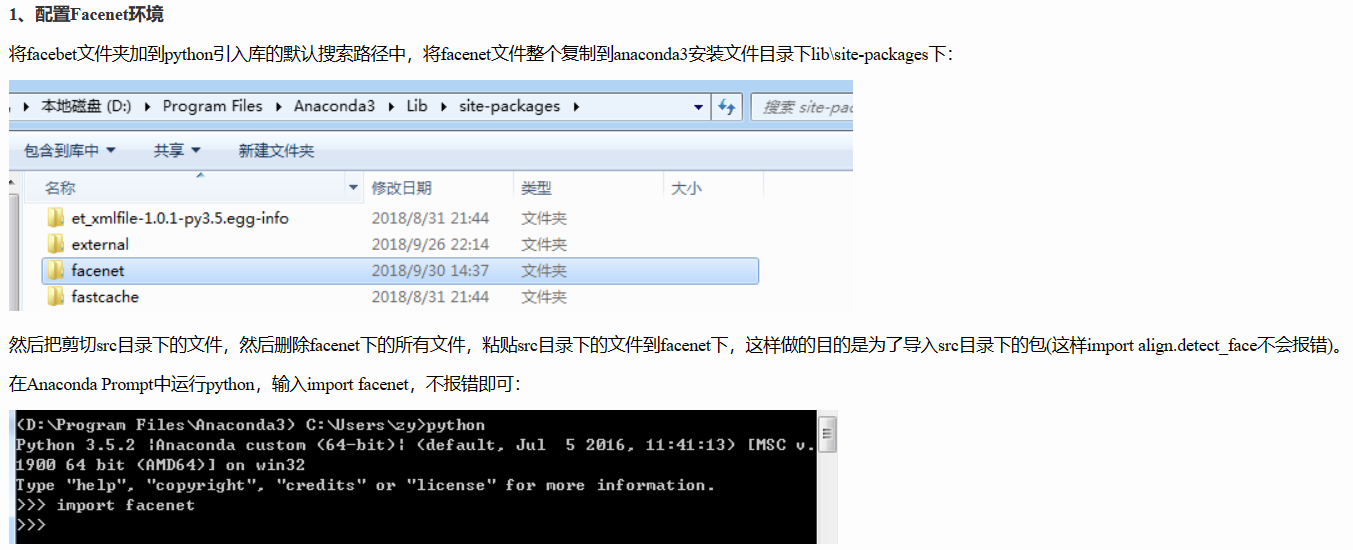


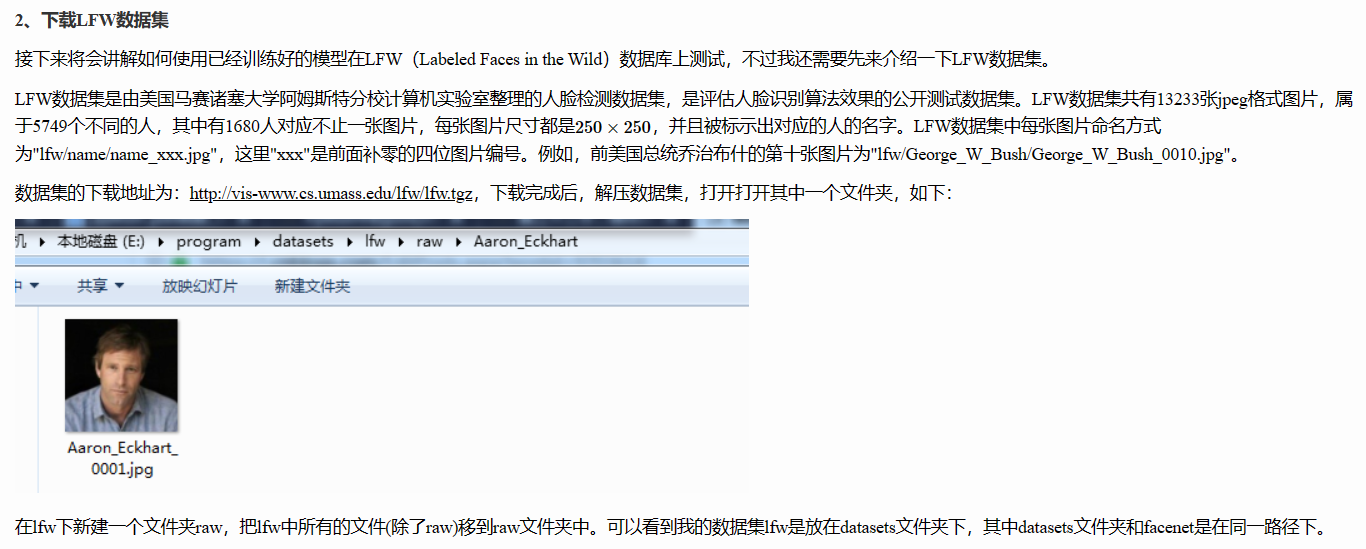
Facenet结构：



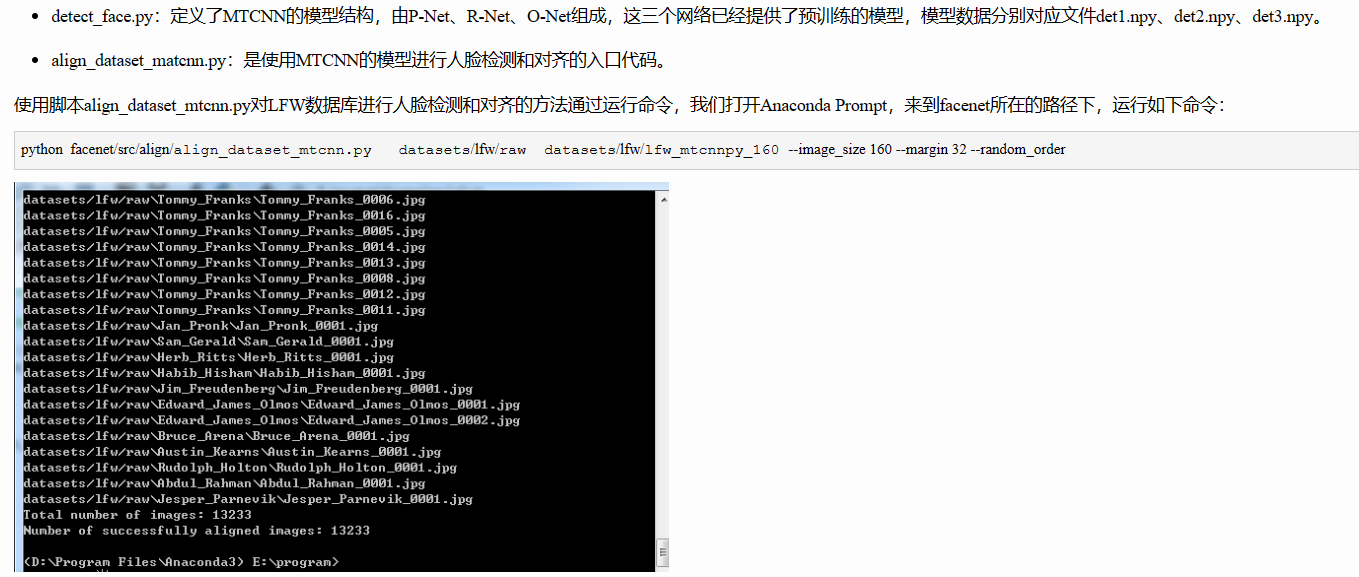
安装依赖库：

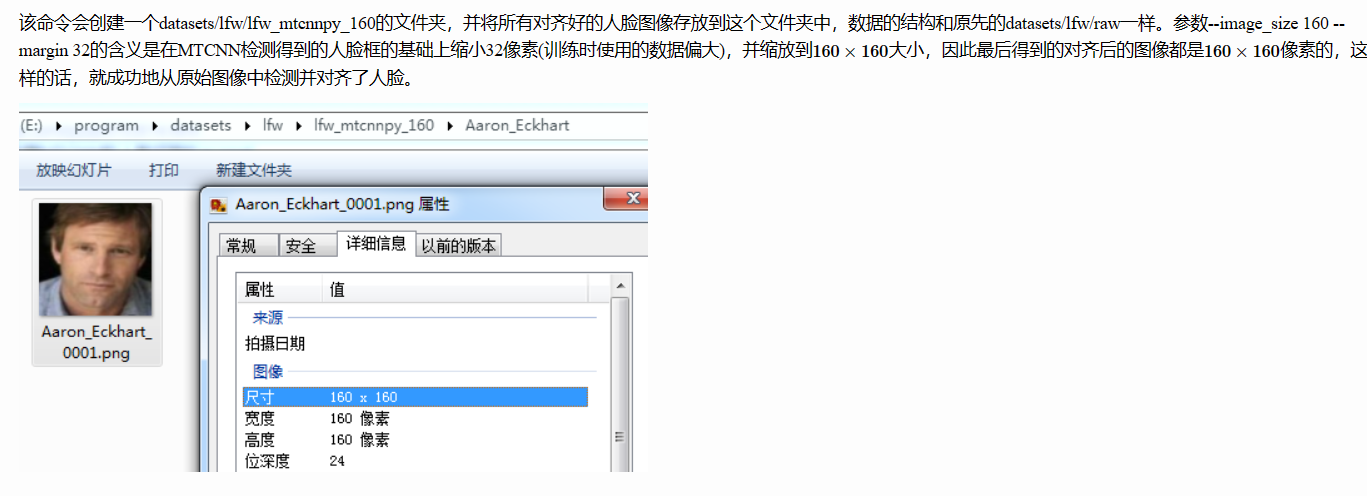


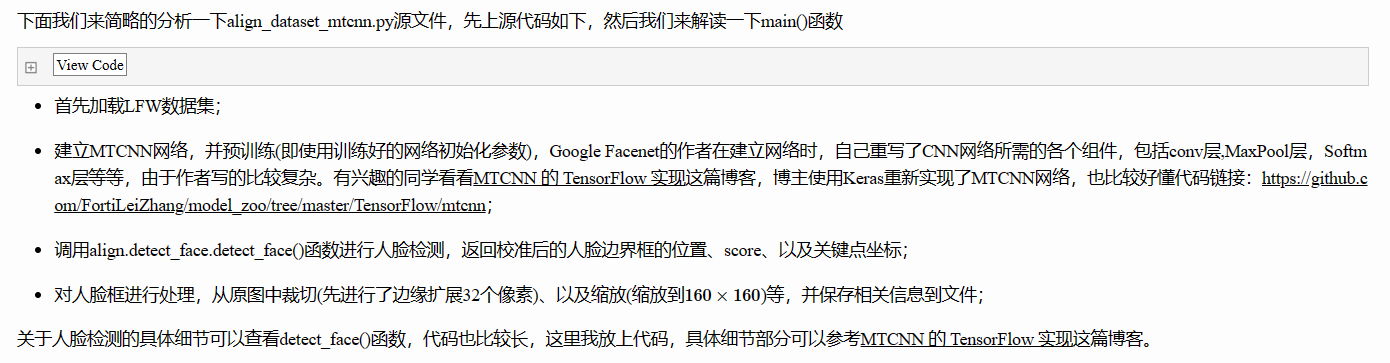




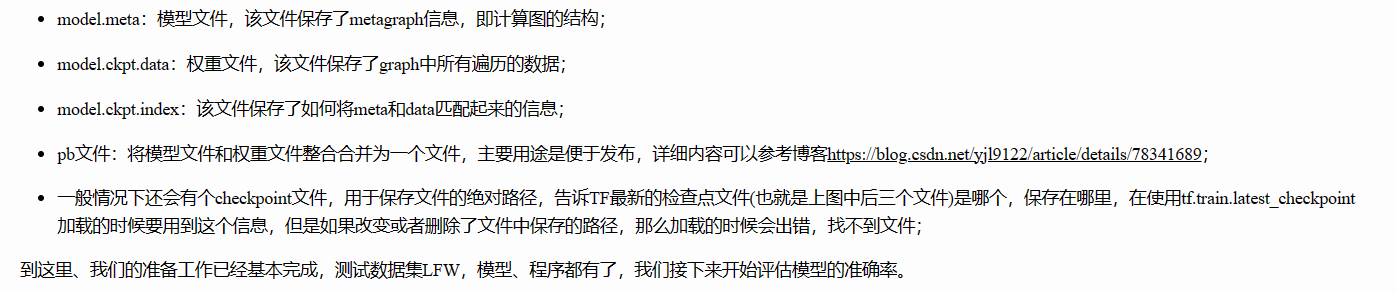




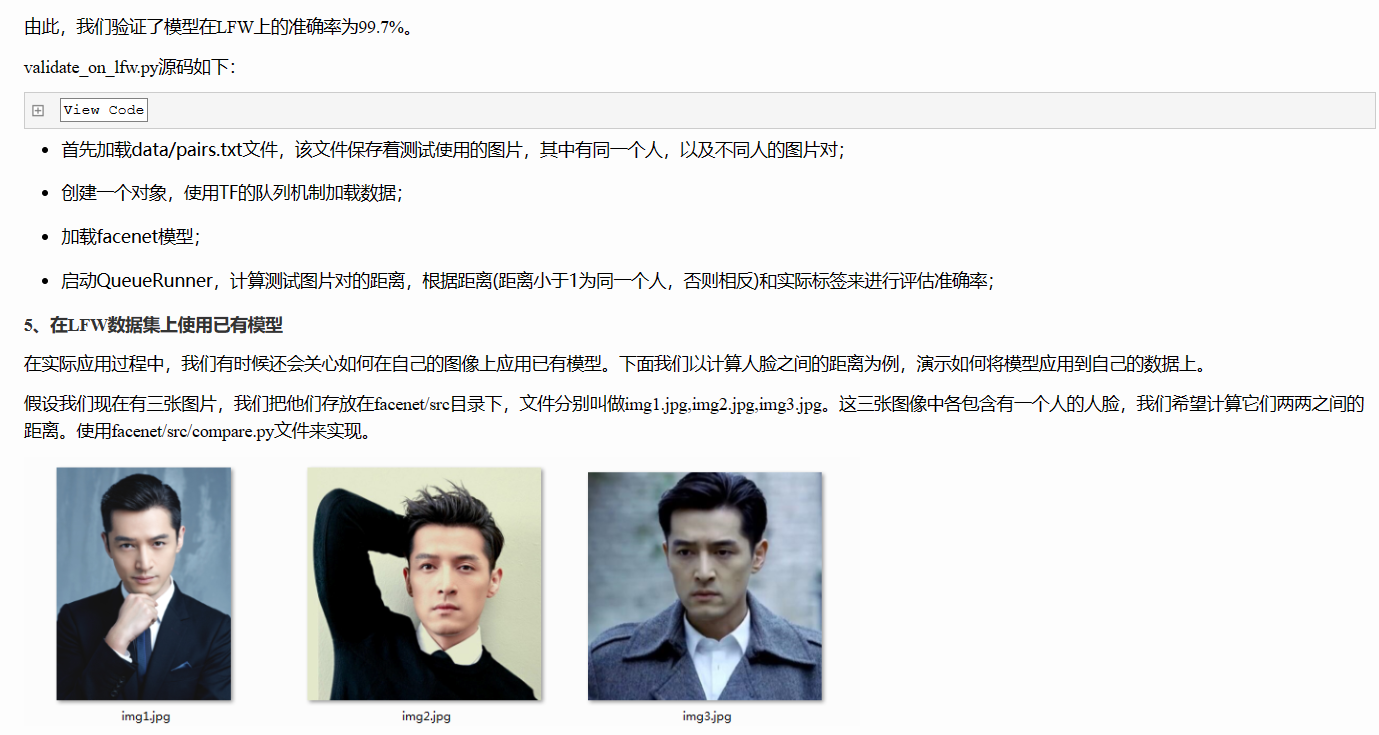


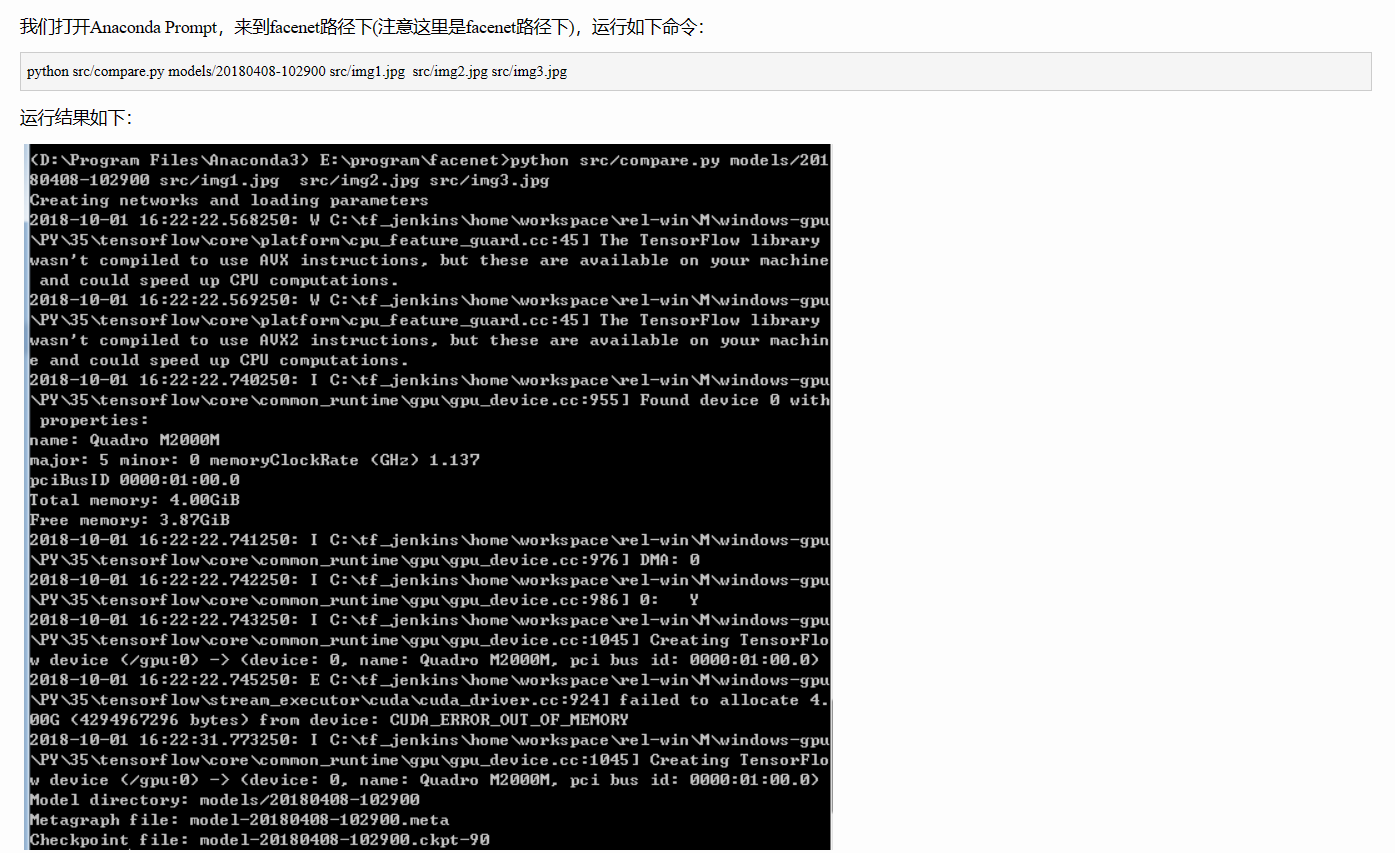




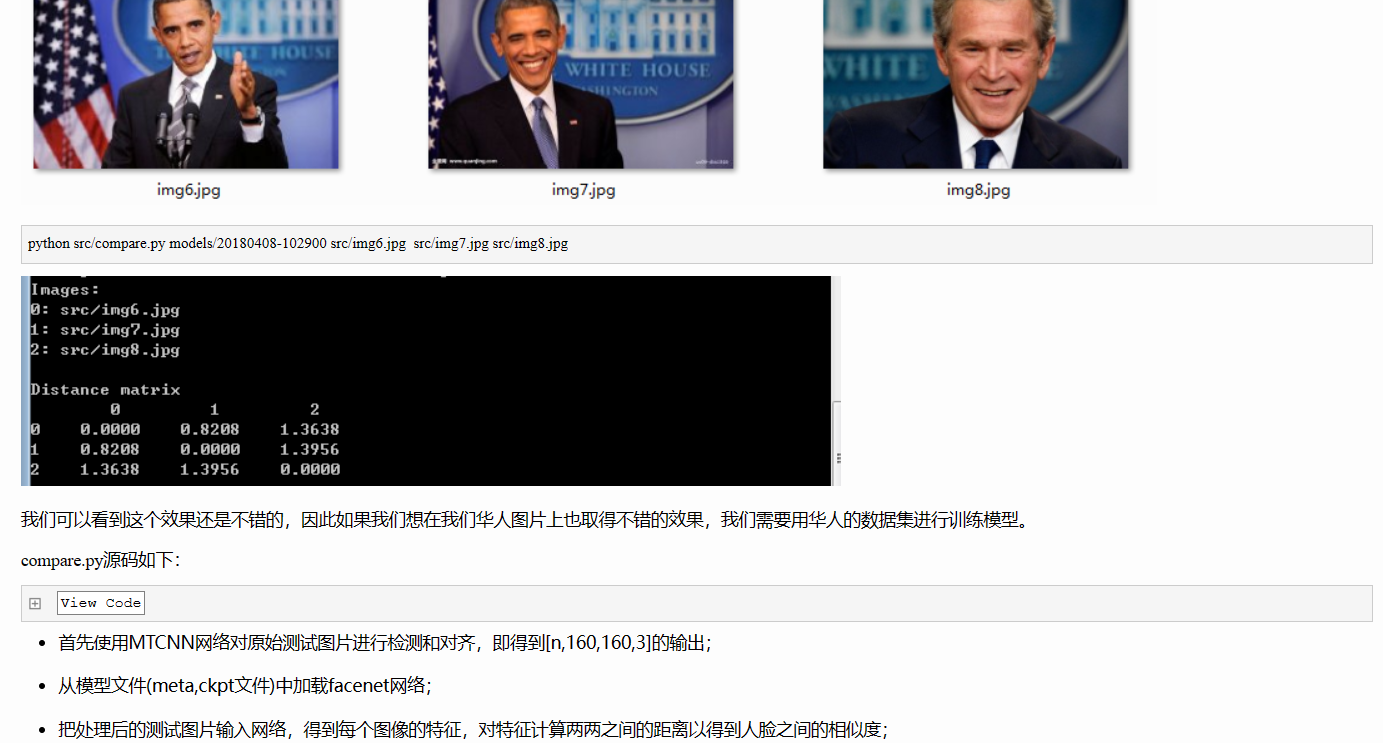


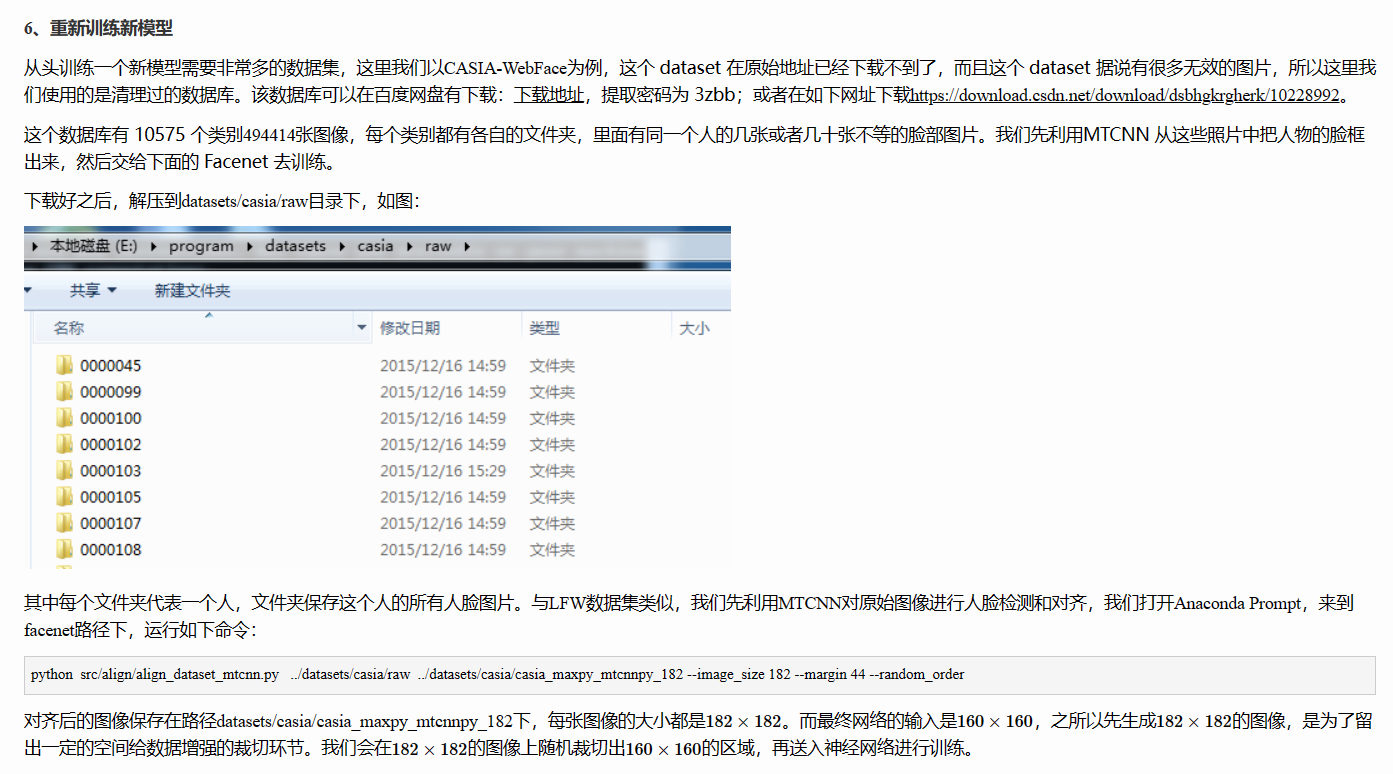








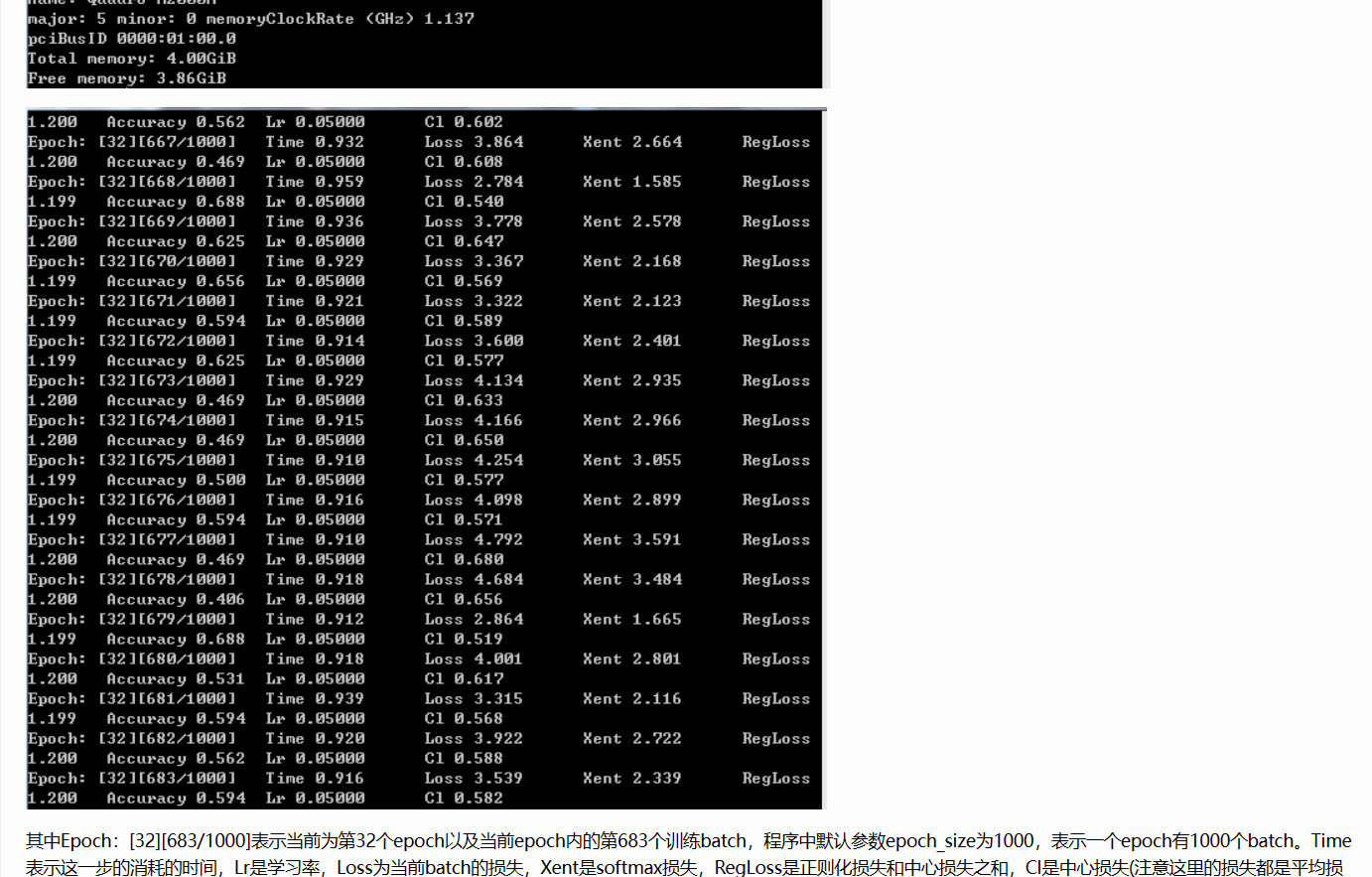




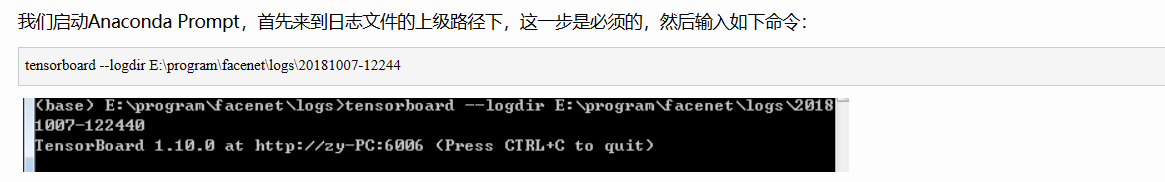


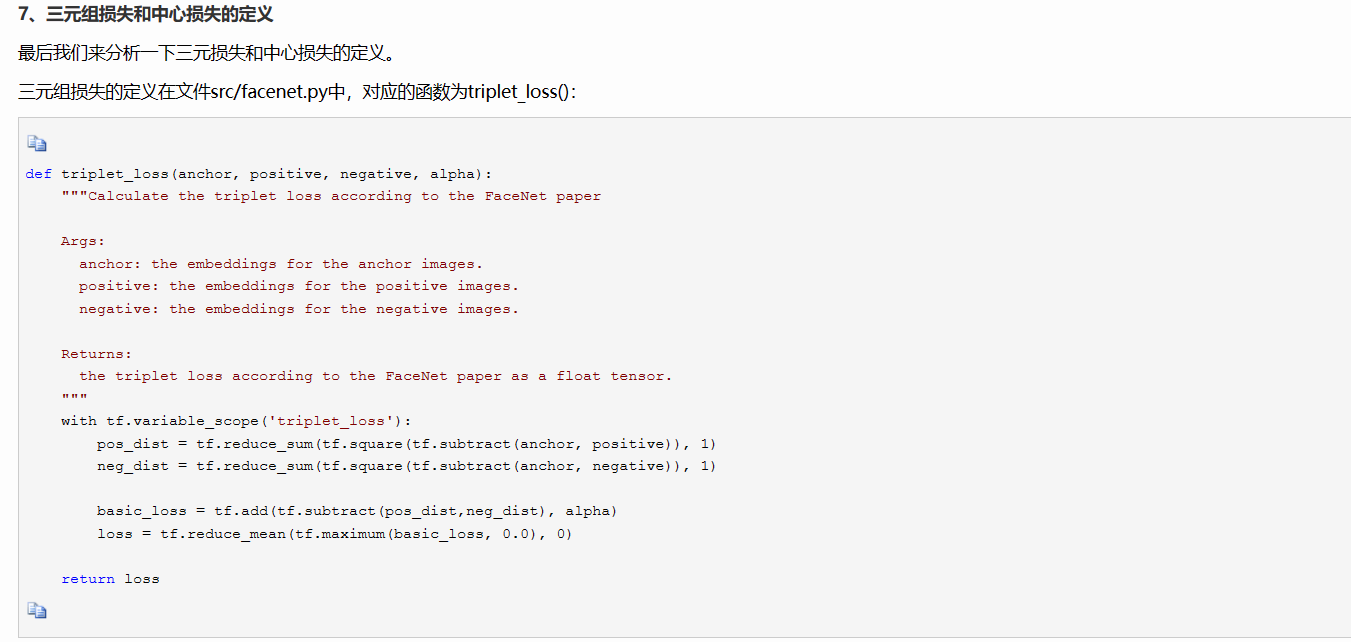


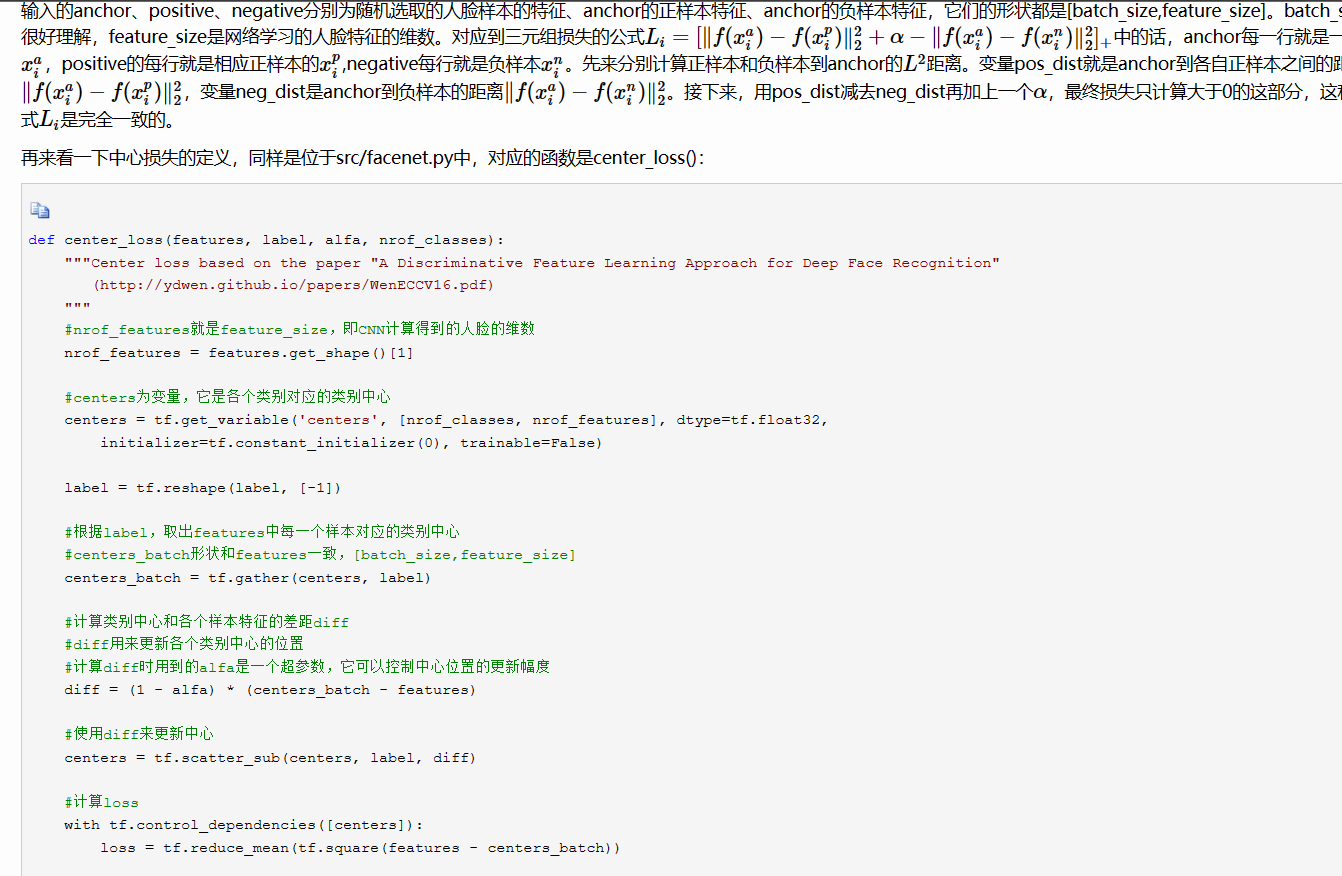














1. 小组分工说明

段慧敏：UI交互设计、人脸识别算法

杨语朦：相似度检测算法

许可：原型设计、测试调参

王豪琴：数据搜集整理、测试调参

1. 参考文献

[Tkinter教程（非常详细） (biancheng.net)](http://c.biancheng.net/tkinter/)

[Python-opencv人脸检测和识别实现（有代码和资源） - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/192747600)

[第三十七节、人脸检测MTCNN和人脸识别Facenet(附源码) - 大奥特曼打小怪兽 - 博客园 (cnblogs.com)](https://www.cnblogs.com/zyly/p/9703614.html#_label0)

[(108条消息) FaceNet源码解读：史上最全的FaceNet源码使用方法和讲解（一）（附预训练模型下载）\_msceleb\_model\_东写西读李老湿的博客-CSDN博客](https://blog.csdn.net/u013044310/article/details/79556099)

[(108条消息) FaceNet源码解读2：史上最全的FaceNet源码使用方法和讲解（二）\_东写西读李老湿的博客-CSDN博客](https://blog.csdn.net/u013044310/article/details/80481642)

[FaceNet: A Unified Embedding for Face Recognition and Clustering - 百度学术 (baidu.com)](https://xueshu.baidu.com/usercenter/paper/show?paperid=b5457de3c4f74fbbb7d4703e93dc5974&site=xueshu_se) [人脸识别网络 FaceNet - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/609825555)

[(108条消息) 利用MTCNN和facenet实现人脸检测和人脸识别\_AI吃大瓜的博客-CSDN博客](https://panjinquan.blog.csdn.net/article/details/84896733?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-84896733-blog-119447454.235%5Ev38%5Epc_relevant_sort_base3&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-84896733-blog-119447454.235%5Ev38%5Epc_relevant_sort_base3&utm_relevant_index=2)