浙江大学

本科实验报告

实验课题：　构建实时人脸识别系统

学院：　计算机学院

专业：　数字媒体技术

指导教师：　潘纲

组长：　王维维　3110105133

组员：　伊力哈木江·图拉麦提　3110105131

装 订 线

一、实验目的

输入:

实时输入摄像头视频

任务描述:

1. 学习人脸识别的基础知识以及 OpenCV 中关于人脸检测和人脸匹配的相关函数

2. 用 OpenCV 的人脸检测以及人脸识别功能，构建一个人脸识别系统，实现实时识别摄像头中最大人脸的身份。

3. 要求已知的人脸库中至少有 5 个不同的人，其中必须包括助教人脸照片（提供 10 张，大家根据需要选取一张或多张），其他人自己采集。

输出要求：

a. 将视频实时显示出来

b. 如果检测到人脸，则在视频上面实时叠加显示检测到的人脸框

c. 如果识别出来某个人，则将人的名字写在人脸框的上方，并将这个人在库里对应的最像的照片缩小叠加显示在边上

d. 每识别出一个人，将结果显示出来后，就停止检测与识别。直到按了空格键，才继续开始检测与识别过程。

二、开发环境与技术说明

1.开发环境

IDE: Visual studio 2010

第三方库:opencv 库文件

语言： **C++**

2.实验中相关技术说明:

(1)摄像头读取

(2)人脸检测算法

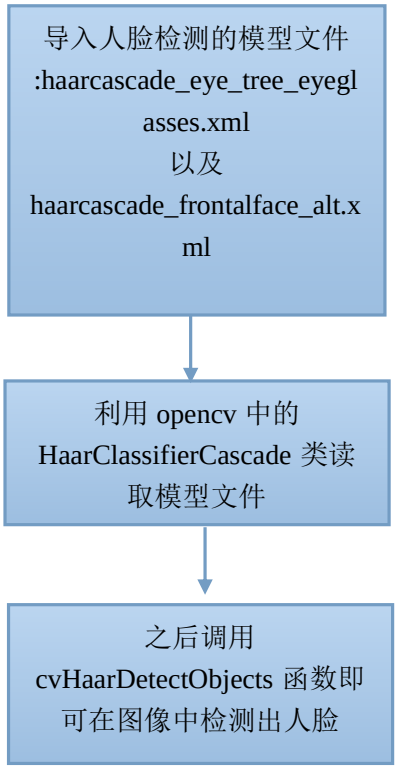
(3)Eigenface 人脸识别模型

(4)Fisherface 人脸识别模型

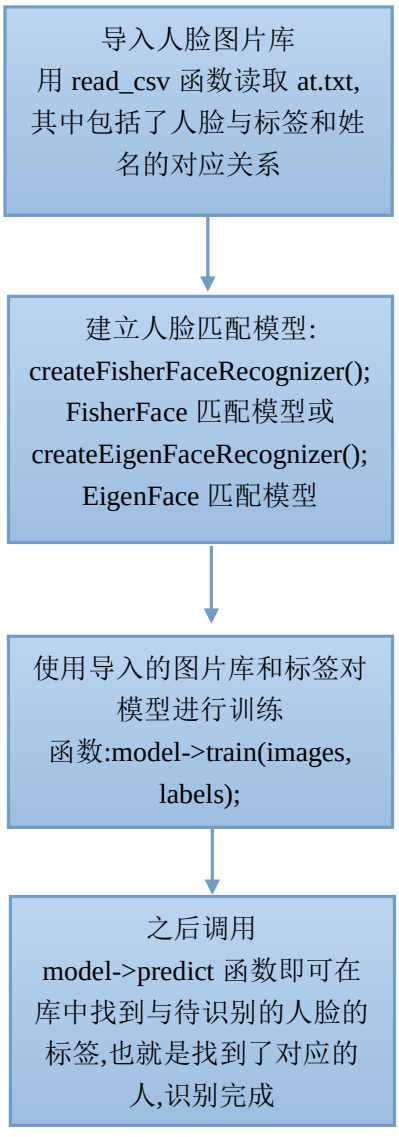
(5)LBPH 人脸识别模型

3.系统架构:

人脸检测模块:



人脸识别模块:



4.大致原理:

人脸检测属于目标检测(object detection) 的一部分，主要涉及两个方面

先对要检测的目标对象进行概率统计，从而知道待检测对象的一些特征，建立起目标检测模型,也就是本文中提到的检测模块。

人脸检测原理:

OpenCV 在物体检测上使用的是 haar 特征的级联表，这个级联表中包含的是 boost 的分类器。人们采用样本的 haar 特征进行分类器的训练，从而得到一个级联的 boost 分类器。

首先将输入的图片统一成相同的尺寸，这个过程被称为归一化，然后进行统计。一旦分类器建立完成，就可以用来检测输入图片中的感兴趣区域的检测了，一般来说，输入的图片会大于样本，那样，需要移动搜索窗口，为了检索出不同大小的目标，分类器可以按比例的改变自己的尺寸，这样可能要对输入图片进行多次的扫描。计算机中的图片是一个数字组成的矩阵，程序先计算整个窗口中的灰度值 x，然后计算矩形框中的黑色灰度值 y，然后计算(x-2y)的值，得到的数值与 x 做比较，如果这个比值在某一个范围内，则表示待检测图片的当前扫描区域符合边界特征，然后继续扫描。

因为是基于视频流的目标检测，我们事先不太可能知道要检测的目标的大小，这就要求我们的级联表中的分类器具有按比例增大(或者缩小)的能力，这样，当小的窗口移动完整个待检测图片没有发现目标时，我们可以调整分类器的大小，然后继续检测，直到检测到目标或者窗口与待检测图片的大小相当为止。

人脸识别原理:

opencv 中有一个 FaceRecognizer 类,目前支持的 3 种人脸识别的模型:

特征脸 EigenFace、Fisher 脸 FisherFace、LBP 直方图 LBPHFace。分别调用函数 createEigenFaceRecognizer( ), createFisherFaceRecognizer( ) ,createLBPHFaceRecognizer()建立模型。

这个类经建立模型后,便可将模型中的图片的 PCA 特征提取出来,PCV 特征的维度可以自己定义。之后传入测试人脸图片,则可以将图片中的 PCV 特征提取出来,之后与模型中的每个 PCV 特征进行比较,差别最小的便是人脸识别的结果。

三、具体步骤

(一)算法步骤

(1) 首先用 (CvHaarClassifierCascade*)cvLoad(cascadeName.c_str(), cascadeMemory, 0, 0)函数导入人脸检测的模型文件

(注 : 模 型 文 件 即 :haarcascade_eye_tree_eyeglasses.xml 以 及 haarcascade_frontalface_alt.xml 如果读取失败,则直接退出程序)

(2)导入人脸图片库

用 read_csv 函数读取 at.txt,其中包括了人脸与标签和姓名的对应关系

(3)建立人脸匹配模型:

createFisherFaceRecognizer(); FisherFace 匹配模型

createEigenFaceRecognizer(); EigenFace 匹配模型

注:这两个模型中任选一个即可,各自的优缺点下文会提到

(4)使用导入的图片库和标签对模型进行训练

函数:model->train(images, labels);

(5)通过 capture 从摄像头中读取帧画面，使用 cvHaarDetectObjects 函数即可在图像中检测出人脸,将人脸数据切割出来调整尺寸之后调用 model->predict 函数即可在库中找到与待识别的人脸的标签,也就是找到了对应的人

(6)在库中的同一 label 中通过图像差值比较算法找到最相似的一张图像作为识别的结果图

(7)显示出当前帧并调用 cvRectangle 在检测出的人脸周围绘制红框,调用 cvSetImageROI 和 cvCopy 函数在框左边显示识别的结果图,在红框上方显示姓名

(8)算法完成

(三)关键函数说明

(1)cvLoad(cascadeName.c_str(), cascadeMemory, 0, 0)

功能:导入人脸检测的模型文件

(2)cvHaarDetectObjects

功能:在视频帧中检测出人脸

(3)createEigenFaceRecognizer()

功能:建立 EigenFace 人脸识别模型

(4)createFisherFaceRecognizer( )

功能:建立 FisherFace 人脸识别模型

(5)model->train(images, labels);

功能:使用导入的图片库和标签对模型进行训练

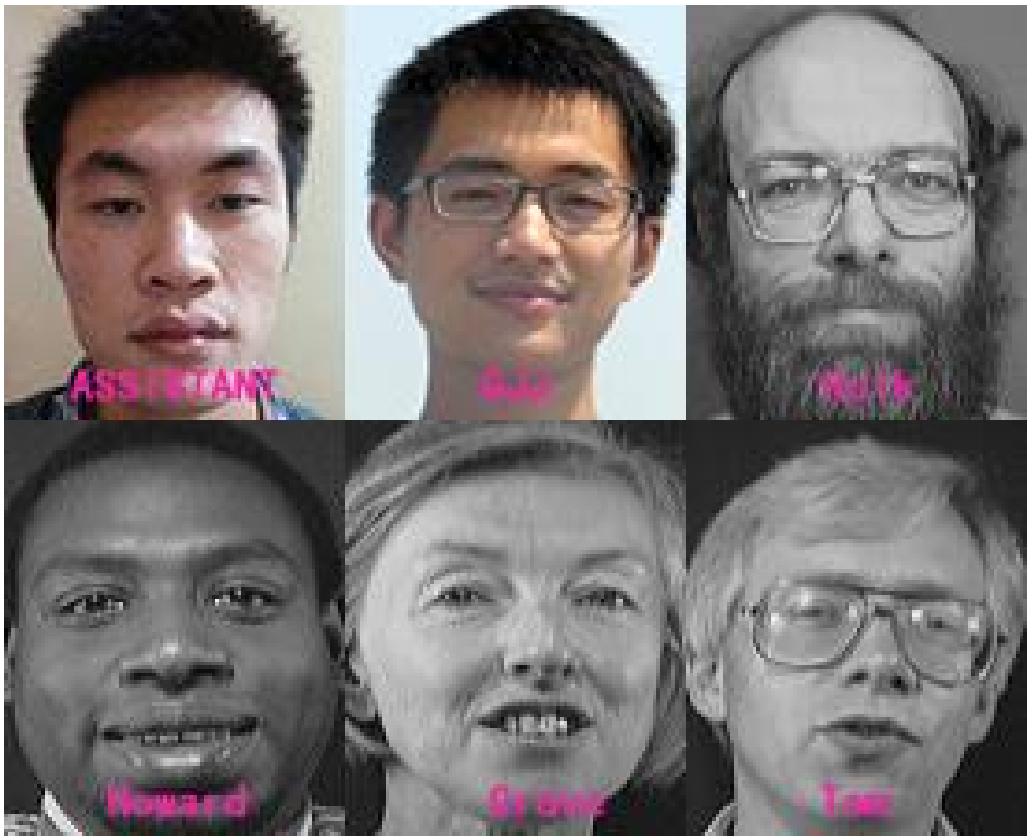
(6)model->predict(detectFace, predictedLabel, confidence);

功能:在库中找到与待识别的人脸的标签,也就是找到对应的人

四、实验结果与分析

实验结果:(程序可以有很多种检测方式,一种是传入图片进行检测,还有一种是通过摄像头进行实时检测,这里只列出图片检测的结果,因为检测的步骤和识别的步骤与摄像头检测一样,因此结果也是一致的)

本实验中的人脸库图片(共有 6 个人,总共有 18 张,按下图中的顺序依次是 ASSITANT(6 张),GJJ(4 张),Hulk(2 张),Howard(2 张),Grove(2 张),Tom(2 张))



人脸识别结果:

(1)Tom 成功识别(fishFacer 和 eigenFace 两种模型均可识别成功)

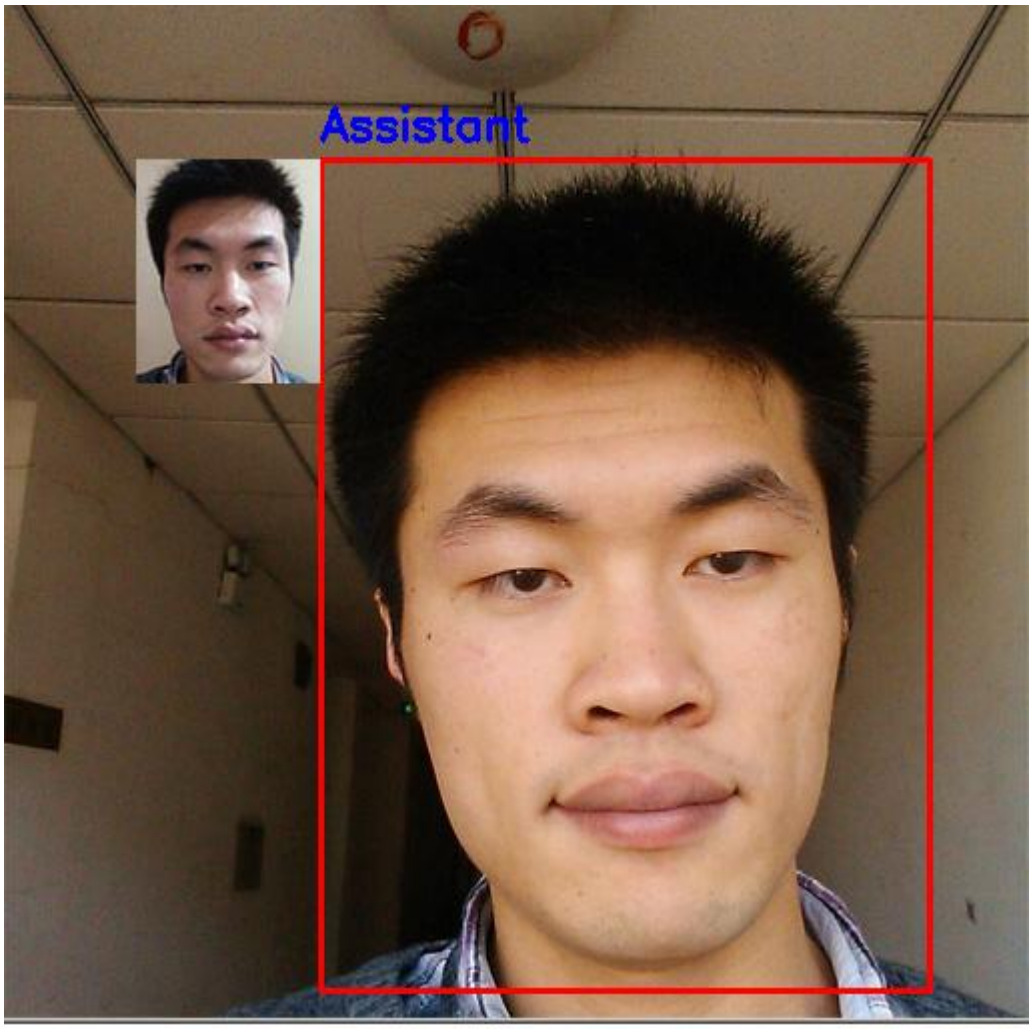
关于(1)的分析:

此人脸识成功,因为此图片的光照和面部旋转角度均良好.

由于红框的大小为整张图片,因此显示在框上方的姓名并不能被看到。

(2)assistant(助教)成功识别

(fishFacer 和 eigenFace 两种模型均可识别成功)

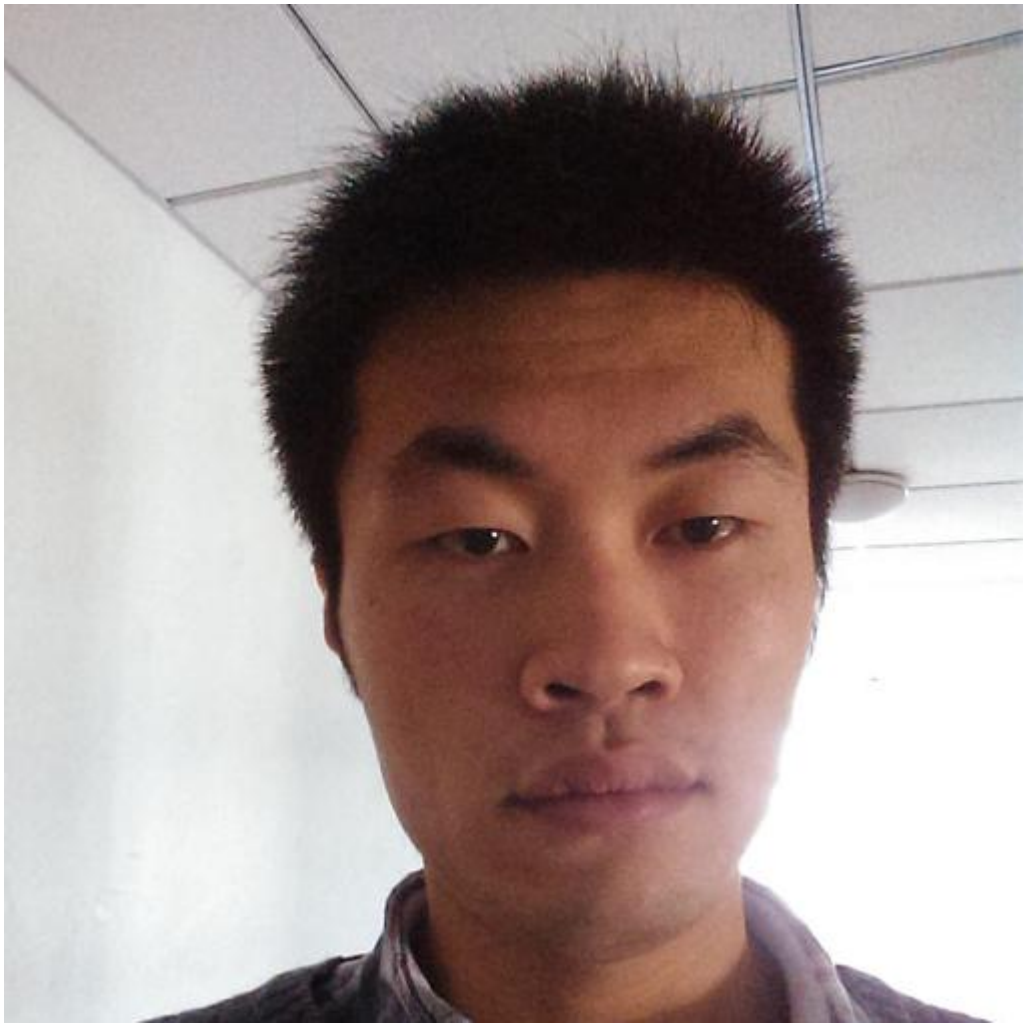


关于(2)的分析:

助教人脸识成功,由于光线较好,头的朝向也比较正,因此可以识别成功。

(3)assistant(助教)检测人脸失败

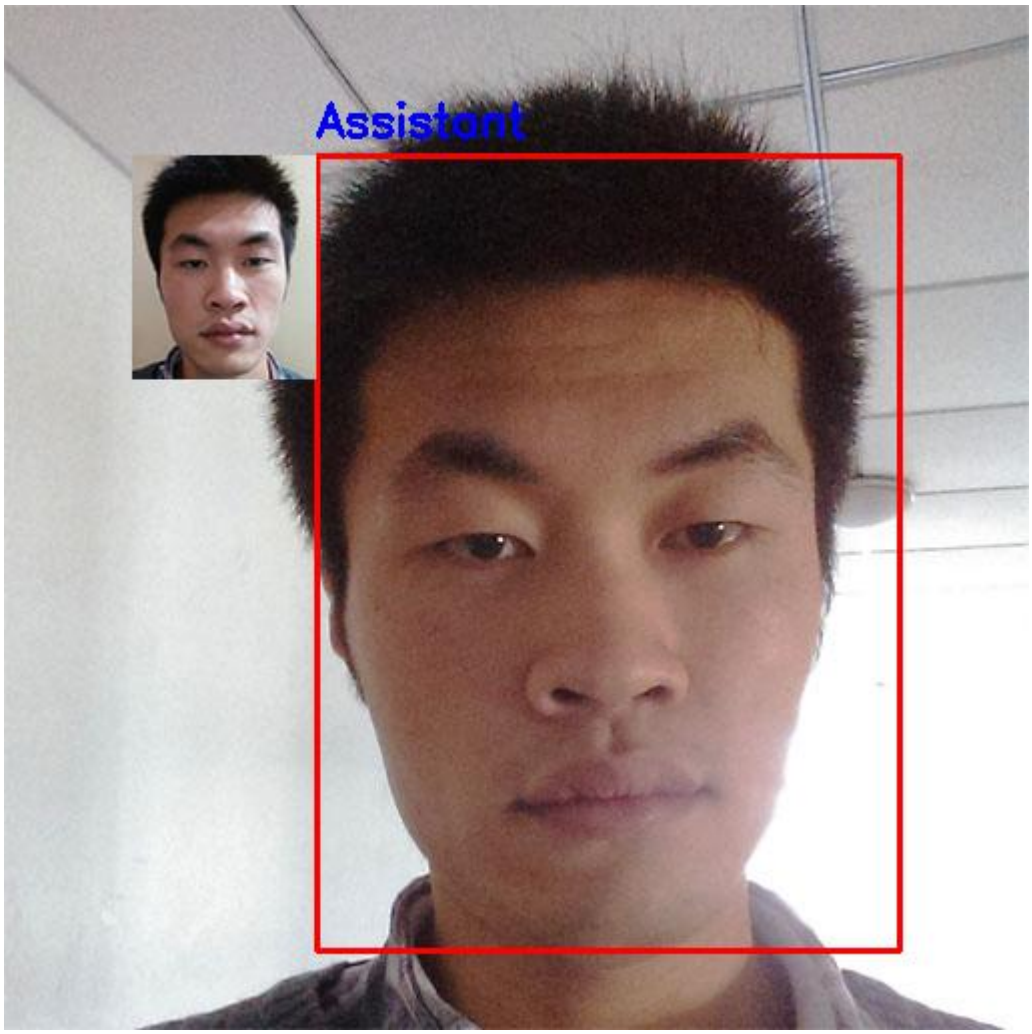
```
Processing 1 ./test_pictures/assistant_test2.jpg
人脸匹配模型训练中......
人脸匹配模型训练结束!
图片读取中...
人脸检测开始!
detection time = 368.39 ms
人脸检测结束!
SOORY,没有检测到人脸,换个角度试试!
人脸匹配结束!
按任意键退出!
```



分析：因为光线的问题，脸部特征提取有误，因此没有检测到助教的脸。自然没有调用人脸识别模块的功能，不能绘制出任何结果。

下面是用 photoshop 将光线调整后的结果：

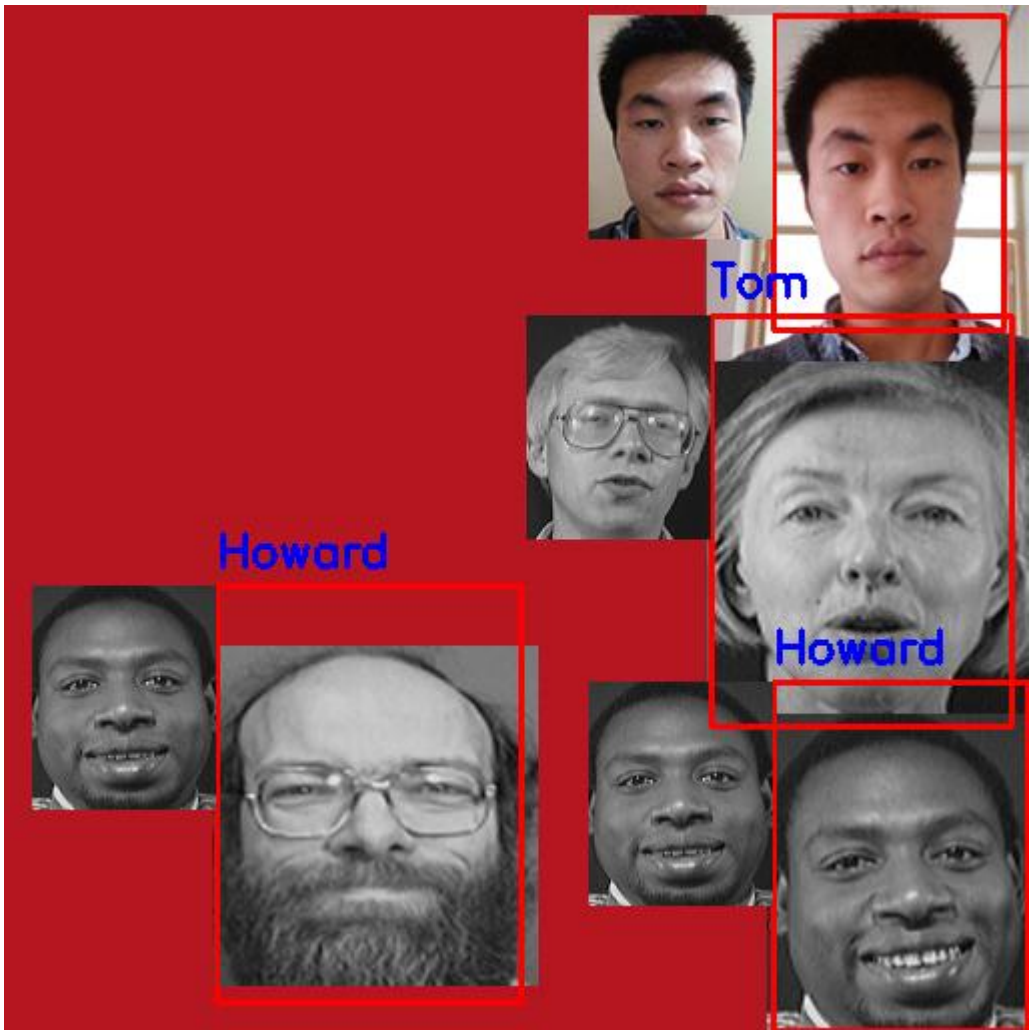
两种模型的结果一致：



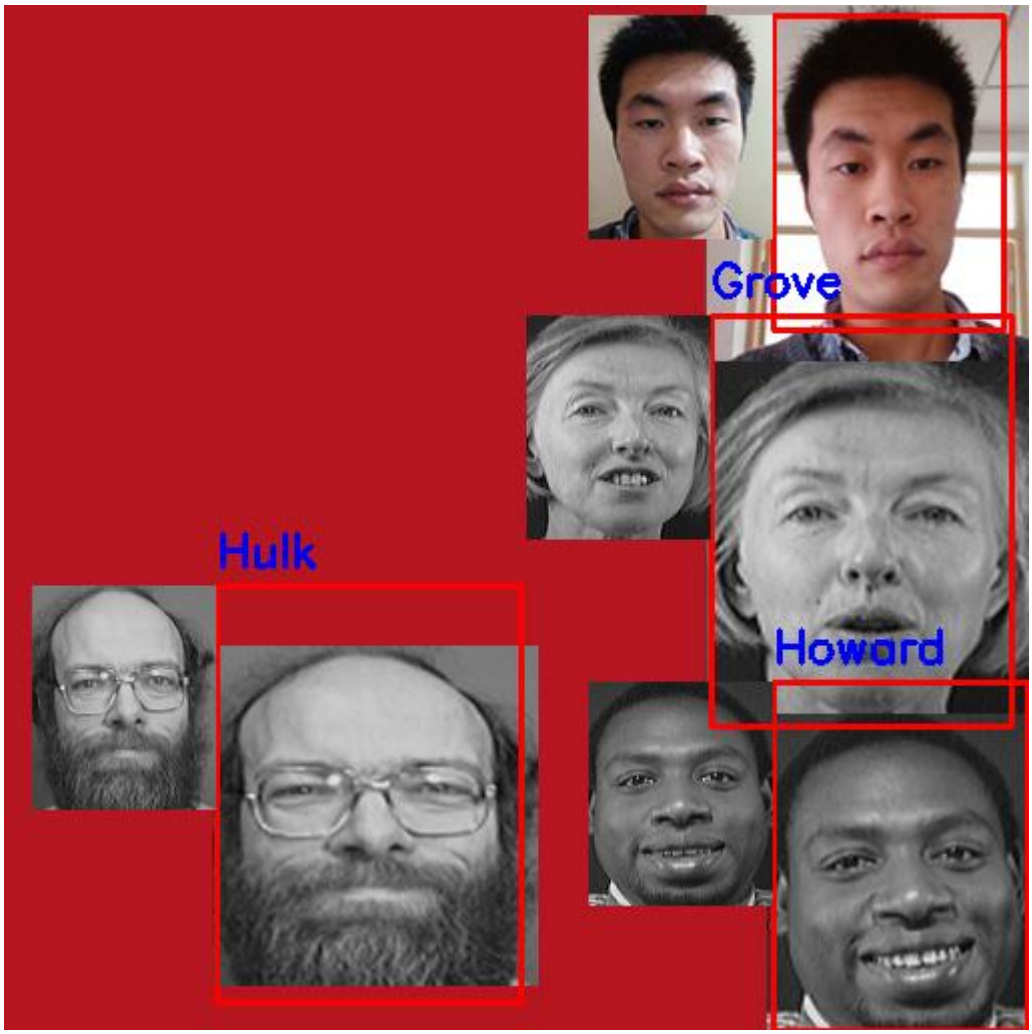
分析:通过调整便能够检测到人脸，说明光照对人脸检测算法的影响比较明显。

(4)一张图片中的多人脸识别

fisherFace 识别结果(识别部分失败)



EigenFace 识别结果(全部识别成功)



分析:

这张图的识别两种模型的结果不一样。是因为使用的人脸识别模型不一样。

FisherFace 是基于 LDA 方法，eigenFace 是基于 PCA 方法。

PCA 考虑的是所有不同种类样本在哪些方向上分布最广，没有考虑不同类之间的关系，而 LDA 则考虑了如何使不同类样本尽量分得最开，是同类样本靠的较近。

因为在人脸检测时，人为地控制了矩形框的大小缩放程度，因此在混合检测时，会将多余的其他人脸的部分截取进去，从而导致了一些不可预料的差错。

五、总结体会

通过此次实验，我们初步掌握了基于 LDA 方法的 FisherFace 以及基于 PCA 方法的 eigenFaced 的使用方法。并能够利用 opencv 中相关函数实现人脸实时识别的程序。经过一系列的测试，得到了许多结果。有许多识别成功的例子，但更多的还是识别失败的案例。这些结果说明了人脸识别是一项难度颇大的技术。在目前的算法实现下，不能够百分之百地识别出正确的人脸。但是随着算法的不断地深入研究，将来定能够找到高效且准确的人脸识别方法。一旦找到这种方法，那么我们的很多生活方式便会发生很大的改变。