

基于人脸关键点的疲劳驾驶监控系统

徐帅

四川大学锦城学院计算机与软件学院 四川 成都 610000

【摘要】随着汽车的发展,许多人拥有了汽车,但疲劳驾驶对交通的威胁也越来越大。驾驶员疲劳驾驶很容易引发交通事故,每年因疲劳驾驶而重伤或死亡的人数有很多。为了预防因疲劳驾驶而引起的交通事故,在驾驶员疲劳时提醒驾驶员注意休息,本文提出了利用人脸识别技术实时对驾驶员进行实时监控,并对驾驶员疲劳驾驶进行提示,防止因为疲劳驾驶而导致的车祸,造成财产和身体生命的损失。本文基于 Dlib 的人脸检测器,对人脸的 68 个关键点进行提取,通过对驾驶员的眼睛和嘴巴关键点的比例进行判断,从而实现对驾驶员的精神状况进行实时监控。

【关键词】人脸识别; 疲劳驾驶; Dlib; 人脸关键点; 人眼定位; 嘴巴定位

引言

导致交通事故的原因有很多,疲劳驾驶是主要原因之一。疲劳驾驶是指驾驶员连续很长时间驾驶或很长距离驾驶,产生心理机能和生体机能的失调,而在客观上出现驾驶技术下降的情况。睡眠质量不佳、长时间行驶在高速路上等^[1]都会引起疲劳驾驶。驾驶员疲劳驾驶就会引发想睡觉、精力不集中等问题,很容易引发交通事故,从而产生身体生命和财产的损失。为了防止这些损失,本文基于 Dlib 的人脸识别器,实现了了人脸识别技术,通过对人脸 68 个关键点进行提取,再分别提取出眼睛和嘴巴的关键点,计算出眼睛和嘴巴的特征值,从而实现了对驾驶员疲劳的实时监控。

为了实现上述方法,本文采用了 Dlib 的人脸检测接口和人脸特征值算法对驾驶员的嘴巴和眼睛进行特征值的计算。具体实施如下:通过 Dlib 的人脸关键点的提取,提取出人脸的 68 个关键点,再取出第 37-42 个关键点表示左眼关键点,第 43-48 个关键点表示右眼关键点,第 49-68 个关键点表示嘴巴关键点,通过计算出其上下左右关键点占比的变化,判断出是否打哈欠或者闭眼,如果打哈欠和闭眼超过三次,则会弹出语音提示注意休息。

1 基本原理

1.1 Dlib 人脸检测器

Dlib 一个开源的 python 第三方库,它不依赖于其他库(自带图像编解码库源码)。它现在广泛运用于各种大性能高计算环境,也时常运用在人脸识别中。更重要的是,Dlib 的文档非常完善,例子非常丰富。就像很多库一样,Dlib 也提供了 Python 的接口,安装非常简单,

先安装好 dlib 的 wheel 文件,再进入这个文件所在的文件夹中,直接使用语句 pip 语句即可: pip install dlib。

本文采用 Dlib 已经训练好的 shape_predictor_68_face_landmarks.dat 模型对人脸的 68 个关键点算法^[2],通过提取通过对嘴巴和眼睛的关键点提取进行计算,判断是否劳累。如下为脸部特征图 1:

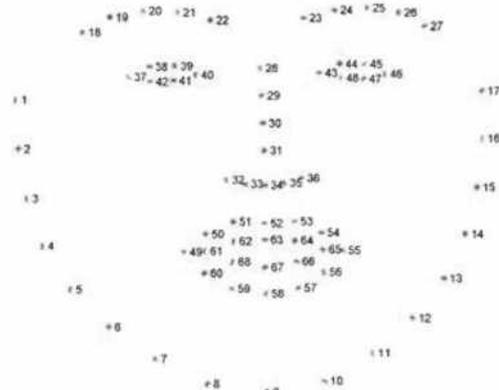


图 1 脸部特征图

1.2 眨眼原理分析

通过 Dlib 提取 68 个关键点中的第 37-42 个关键点作为左眼的关键点,第 43-46 个关键点作为右眼的关键点,由于单判断某个点的位置的话会因为角度或者光线等问题而影响判断,所以采用了 Soukupov á 和 Čech 在其 2016 年的论文中推导出反映这种关系的方程,称为眼睛纵横比 EAR (Eye Aspect Ratio)^[3]。分别传入 EAP 算法,分别计算出其比例:

其中 p1-p6 分别为眼睛的六个关键点如图 2:

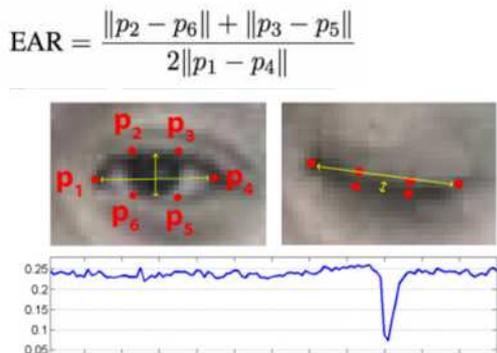


图 2 眼睛关键点及眼睛纵横比

由于睁开眼睛时， p_2-p_6 和 p_3-p_6 的距离最大，通过 EAR 公式计算出来的值就越大，而闭上眼睛的时候他们又最小， p_1-p_4 的距离几乎没有什么变化，所以通过 EAR 公式计算出的值就越小。从而就可以由这个变化值，对眼睛的睁或闭进行判断。通过一个为 0 的变量 a 对闭眼时间进行控制，如果两只眼睛的纵横比的平均值小于 0.2，则算一次闭眼，变量 a+1，如果这个变量超过了 10，就算一次疲劳的闭眼，系统就会记录一次。

1.3 打哈欠原理分析

同样的原理，通过 Dlib 提取的人脸 68 个关键点中的第 49-68 个关键点表示嘴巴关键点，如下图 3：

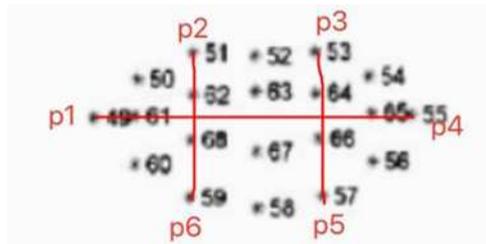


图 3 嘴巴关键点

分别将这几个点传入上部分所叙述的 EAR 公式中，计算出其特征值。由于嘴巴是判断是否张嘴打哈欠。当嘴巴闭着的时候 p_1 和 p_4 的欧式距离最小， p_2-p_6 和 p_3-p_5 的欧式距离最小；当深度打哈欠时， p_2-p_6 和 p_3-p_5 的欧式距离最大， p_1-p_4 的欧式距离最小^[4]，通过 EAR 计算出来他们之间的比值，当深度打哈欠时，EAR 公式计算出来的值最大，通过设定一定的阈值对是否深度打哈欠进行判断，如果大于这个阈值，并且也是生成一个变量 b 来计算打哈欠的时间，如果超过 10 次判断都大于这个阈值^[5]，则算一次打哈欠，系统就会记录这次打哈欠，如果打哈欠和闭眼超过三次，就会弹出语音提示，注意休息。

1.4 人脸库的加载与保存

为了实现身份识别，首先要驾驶员进行人脸和身份

的录入。人脸和身份的录入分为两种方法，一种是离线录入，一种是实时录入^[6]。离线录入则是直接将含有人脸的图片放在指定的人脸文件夹中，在本文的实验平台上，对离线录入人脸的图像有如下要求：

(1) 所有要加入人脸库的图片必须放在同一指定文件夹中，这样放以便后续使用对人脸库加载时，能对每张人脸都进行读取和保存^[7]；

(2) 图片的名字以该驾驶员的名字命名，以这样命名是为了满足照片中获取到的人脸和人脸所对应的身份进行对应起来，也方便读取所有的身份信息；

(3) 格式为 JPG，图片的大小不得超过 500KB，分辨率不得超过 1000×100 ，这样规定是为了让整个系统不会因为图片格式的问题而出现无法读取人脸或人脸信息的情况，而图片大小的规定则是不让图片因为图片太大，而造成图片人脸识别时间太长；

(4) 图中必须有正脸，如果图片中没有正脸，在获取图片人脸时，就会出错，无法获取人脸和人脸信息。

实时录入则是通过 OpenCV 调用本机的摄像头，输入被录入驾驶员的姓名，就可以开始拍照，再通过 OpenCV 对人脸照片和所输入的驾驶员姓名进行保存。然后通过循环遍历每张图片的路径和文件名，通过 face_recognition 的 face_encodings 计算出每张人脸的 128 脸部编码^[8]进行计算并进行保存，再生成个名字列表对每张图片的人脸所对应的名字进行保存，就生成了人脸库。

1.5 驾驶员身份识别

本文采用了 OpenCV 的 VideoCapture 对本地摄像头进行调用，以便进行驾驶员身份的识别。对摄像头的每一帧进行读取，每 10 帧进行一次人脸的身份判断。对每 10 帧传入的图片通过 face_recognition 的 face_locations 对图像中的人脸进行定位和提取，再通过 face_recognition 的 face_encodings 对人脸的 128 面部编码进行计算，最后传入 face_recognizer 类^[9]中的 face_distance 接口，与刚才获取的人脸库作比较，循环遍历计算出驾驶员与每张人脸库中的人脸的欧式距离以表示两张脸的相似度，欧式距离如果越大，表示这两张脸的相似度越小；欧式距离越小，则表示两张人脸的相似性越大^[10]。再取出驾驶员与每张脸的欧式距离中最小的距离所对应的那张脸，如果这个最小值小于 0.55，则表明驾驶员的身份就是这张脸所对应的名字，如果这个最小值也大于 0.55，则表示该驾驶员不在人脸库中，需要重新对该驾驶员的人脸和信息进行录入才行。

2 系统实现

本系统共四个功能，分别是人脸库的加载，驾驶员身份的识别，打开摄像头开始行车（开始对驾驶员进行

实时监控)和停止行车关闭摄像头四个功能。

人脸库的加载主要是对人脸库中的图片进行人脸和驾驶员身份的提取,提取到的身份和人脸用于后面的驾驶员身份的识别。驾驶员身份辨认功能主要是对驾驶员的人脸与人脸库中的人脸通过比对,从而判断出驾驶员的身份,识别成功后状态栏就会弹出驾驶员身份信息。打开摄像头开始行车功能主要是对人脸的嘴巴的关键点和眼睛的关键点进行识别判断,从而判断出驾驶员的精神状态,如果打哈欠和闭眼超过3次就会弹出语音提示。关闭摄像头则是对上面功能产生的数据清零,以便下次使用。具体过程是加载人脸库,对驾驶员进行人脸识别,开始行车,实时监控人脸眼部和嘴部特征,计算出其特征值,判断是否满足一定的阈值,如果不满足,则算一次打哈欠或闭眼,如果超过三次弹出语音提示。具体框架如下图4:



图4 系统框架图

具体实现图如下图5:



图5 具体实现图

打开摄像头开始行车之后,系统就会对人脸进行定位,同时对人脸的68个关键点进行提取,再取出眼睛和嘴巴的关键点进行判断,图中ECOUNTER和COUNTER分别统计闭眼和打哈欠的时间,EAR和MAR分别显示的是眼睛和嘴巴的横纵比,就是通过判断这两个值,对驾驶员的疲劳情况进行监控。如果驾驶员打哈欠或者闭眼,上面的SHUT-EYES和Yawn就会统计一次,驾驶员状态栏中也会输出驾驶员的动作和时间,如果打哈欠和闭眼的总次数超过3次,就会弹出语音提示,注意休息。

3 结束语

本文基于人脸识别技术的优势,设计一套驾驶人员疲劳监控系统,具有较高的社会价值。本文利用了Dlib的人脸检测器对人脸的68个关键点进行精确定位,通过提取到人脸和嘴巴的关键点进行提取,再通过EAR公式计算出其横纵比,实现起来也比较容易。

本文中提到的获取人脸库的方法适用于人脸库比较小的情况,如果人脸库稍微大些,读取人脸库就会耗费很长的时间。本文中的疲劳驾驶实时监控对驾驶员的疲劳情况判断还有些误差,后期会做出进一步的优化或加入对驾驶员整个头部判断,从而更准确地对驾驶员精神状况进行判断。

【参考文献】

- [1] 刘金星,付阳,李可民. 基于USB3.0传输的人脸识别近红外相机系统[J]. 电声技术,2018,42(8):9.
- [2] 涂鑫. 基于卷积神经网络的课堂人脸打卡算法[J]. 现代计算机,2019(29):39-43.
- [3] 侍新兰,杜友福,王同喜. 基于OpenCV的活体检测[J]. 电脑知识与技术,2020(15):211-213.
- [4] 李刚,王银焯,张政儒,许艳亭,丁茹伊. 基于Python的驾驶员打哈欠在线检测方法研究[J]. 汽车实用技术,2020(3):137-138+141.
- [5] 耿磊,袁菲,肖志涛,张芳,吴骏,李月龙. 基于面部行为分析的驾驶员疲劳检测方法[J]. 计算机工程,2018(01):274-279.