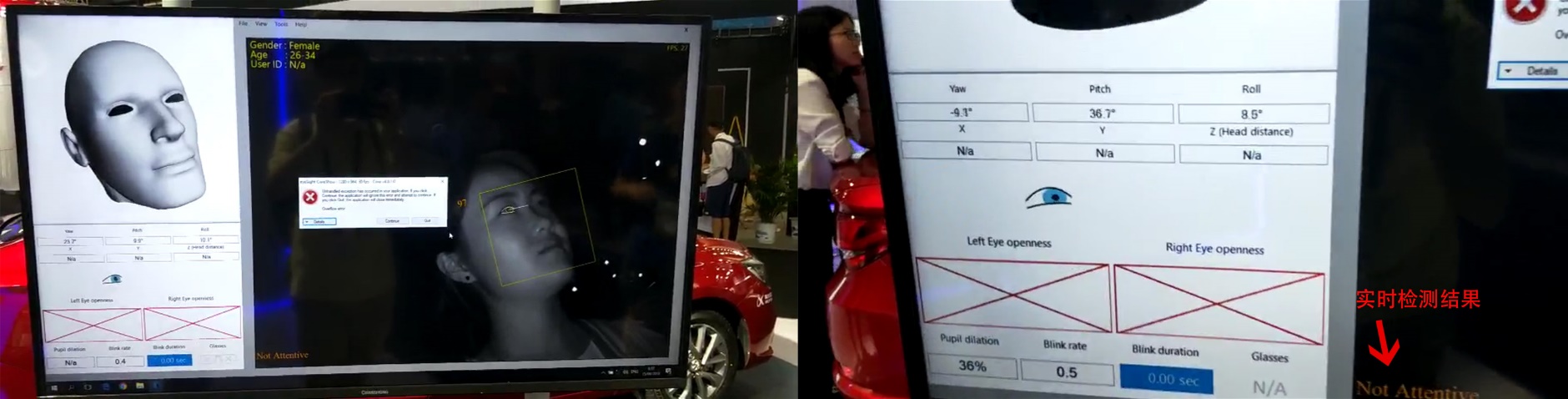
数字图像处理 期末报告

王轲 14307130048

# 摘要

　　相比于完全把神经网络当成黑盒来做训练，本文尝试了一种折中的思路：先通过人脸特征点检测获得特征点，再通过特征点预估人脸位置、角度参数，再通过一个LSTM网络进行参数的训练，并对视频做分类。

# 相关工作

2018年CES Asia展会上，科大讯飞展示了他们的驾驶员疲劳检测系统。他们的系统能通过计算机视觉的方法，从摄像头中获取人脸朝向、位置、瞳孔朝向、眼睛开合度、眨眼频率、瞳孔收缩率等数据，并通过这些数据，实时地计算出驾驶员的注意力集中程度。我在现场体验了他们的系统，非常灵敏准确。

**图1 科大讯飞展会上展出的疲劳驾驶检测技术**

# 总体思路

受到科大讯飞DEMO的影响，我做本期末报告时，也希望使用这种基于特征检测、参数计算的方法。我认为，这种方法不是完全黑盒的（相比于纯CNN网络来训练图片），在可解释性上可能会好一些，此外，这种方法还能够检测并记录别的体征数据，这些体征数据也有许多挖掘价值。

我做得实验分为以下三步：

1. 通过图片特征点检测的方法获得人脸特征点（如眼、鼻、嘴、轮廓）

这一步比较简单，可以使用著名的dlib来解决，并且有效果较好的已训练的模型：

shape\_predictor\_68\_face\_landmarks.dat。使用它，可以得到68个人脸关键点。

1. 通过68个关键点判断人脸的朝向、位置信息

这一步主要原理是使用OpenCV的solvePnP，solvePnP可以通过几组3d坐标投影到2d平面后的对应坐标，来计算出一个物体的朝向和位置。因为单目摄像机的深度信息是缺失的，不能真正得到人脸的3d模型，所以在解人脸朝向的时候，需要配合一些人体测量学的统计数据（鼻根到人脸各个器官之间的距离）。

1. 对第二步得到的6个人脸纬度信息和68个特征点信息进行预处理，即可把一副图片提取成若干个feature。把这些feature放进LSTM神经网络中进行训练，得到预测用的模型。

# 人脸特征点位置检测

人脸特征点检测用到了dlib (King, 2018)，dlib有两个关键API：

dlib.get\_frontal\_face\_detector()和dlib.shape\_predictor(predictor\_path)。

前者是内置的人脸检测算法，使用HOG pyramid，检测人脸区域的界限（bounds）。

后者是用来检测一个区域内的特征点，并输出这些特征点的坐标，它需要一个预先训练好的模型（通过文件路径的方法传入），才能正常工作。

下载预训练好的模型shape\_predictor\_68\_face\_landmarks.dat并使用，可以得到68个特征点位置的坐标，连起来后，可以有如图所示的效果（红色是HOG pyramid检测的结果，蓝色是shape\_predictor的结果，仅把同一个器官的特征点连线）。

**图2 人脸特征点位置检测**

# 通过特征点坐标计算出人脸六维信息

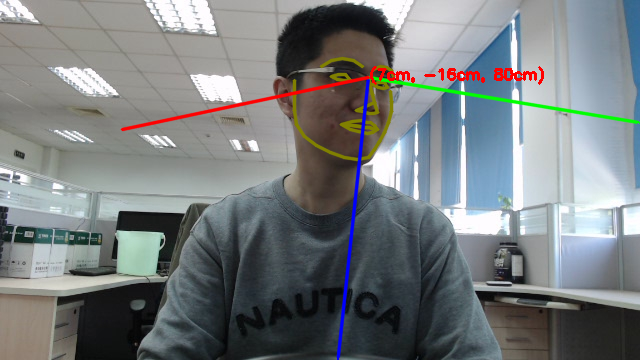
疲劳驾驶打盹时，人脸会朝下垂，有时会轻微晃动。这和人清醒时目光向前或略微向上，头部稳定转动很不一样。这是我们对疲劳驾驶直观的理解。

因此，计算人脸朝向信息和位置信息很重要。如果有可能，还应该计算瞳孔的朝向信息，但限于个人技术能力，尝试了一段时间没有能够成功。

如前文所述，单目摄像机不含有深度信息，其实单目的图像信息是无法估算人脸朝向的（因为相当于3维坐标投影到2维平面上，无法还原）。要把2维信息还原到3维信息，需要一些额外的信息（或假设），如人体测量学中，人脸五官平均距离 (Wikipedia, 2018)。

这里参考了一篇论文 (Lemaignan S. G., 2016)，和他的代码实现 (Lemaignan, 2018)。代码实现是C++的，可以用来输入一副图片，输出图片中人脸的六个维度信息。然而，这样使用起来并不方便，因此我对C++代码进行了改造，使其可以支持批量处理，并以JSON数组的格式输出处理完之后的数据。具体的改造和代码运行方法请看我提交的源码。

使用本算法批量处理数据集中的所有图片。处理完成之后，便额外得到了六维特征，效果如下：



**图3 人脸六维信息的计算**

这六维信息加上之前检测出来的68个坐标点，总共构成142个feature。预处理基本就完成了。把一张图抽取为142个和脸有关的feature，可以屏蔽掉许多图片细节，如光照、肤色、头的位置（和身高有关）、背景里窗帘的飘动……虽然可能丢掉许多有用的信息，但不失为一个探索方向。更加重要的是，一张图片变为142个数字，大大简化了输入数据，每一帧输入数据变简单，就让处理帧和帧之间关系（时序信息）变得可能（计算量不至于过大）。

# 训练LSTM网络模型并进行分类

做到这一步，我感到有一些困难。因为我大学期间并未接触机器学习，传统方法也不知道怎么继续做下去。我进行了一些调查后，觉得还是只能用机器学习的方法进行预测，并发现了LSTM网络，可以用来处理时序信息。我尝试了Tensorflow，但感觉对新手来说比较难写，后来切换到了Keras，用了Keras封装的更加高层、易懂的API。

Keras的LSTM层输入要求是三维的数据：（样本编号，时间帧编号(timestep)，特征向量），预处理完的数据正好是这样三维的，大多数的shape为：(?, 64, 142)。

在LSTM层后，我添加了一层Dense层，基本参考Keras文档中的代码 (Keras, 2018)，模型就搭建完成了。

训练时，我用80%的数据作为训练集，10%的数据作为验证集，10%的数据作为测试集。最终结果为：测试集上58.33%的正确率。

# 结果分析和讨论

最终分类结果不尽如人意，本章节讨论原因和可能的改进方法。

1. **神经网络结构、参数设置较差**

虽然我已经比较尽力，但还是无法理解许多机器学习方面的知识。因此，我认为特征提取完之后，我在模型训练这一步做得比较糟糕。网络有许多参数可以调，调整结构和参数、增加训练次数后，结果可能能有所提升。

1. **单目摄像机获得信息少，人脸信息估计较差**

因为人脸信息估计用到了统计学数据，但这个数据不是很精确，所以最终得到的角度估计效果并不是很好（一些帧存在跳跃现象）。换成双目摄像机，或者RGB-D摄像机，相信特征点提取的效果会好很多。如果疲劳驾驶检测要被广泛投入生产应用，那么升级硬件设备是可行的。

1. **基于人脸检测的图像处理健壮性略差**

用HOG pyramid方法，一些帧中检测不出人脸（可能因为光照、帽子等原因），虽然这些帧不多，但是它们对整体效果可能有比较大的影响。这些帧，我目前的处理方法是跳过（可能导致时序上不连续）。也可以尝试使用插值的方法来弥补缺失的帧。

1. **人脸关键点坐标转换为特征时，太过简单粗暴**

关键点坐标是二维的，现在的做法是，直接把x、y坐标摊平，68个点转换为136个特征。或许转换为8-链码能有更好的效果。

1. **特征抽取仍然不足**

科大讯飞做的系统，可以抽取图片中眼球方向、瞳孔收缩率作为特征。然而本文并没能够抽取出这些特征，因此可能遗漏掉很多关键的信息。

# 总结

　　本文尝试了一种基于特征和参数估计的方法，虽然最后一步分类时，仍然用到了深度学习的方法。受限于个人技术能力，最终准确率比较低，本文对准确率较低的原因做出了分析，并提出了可能的改进策略。

# 参考文献

Keras. (2018年Jun月8日). Sequence classification with LSTM. 检索来源: Keras Documentation: https://keras.io/getting-started/sequential-model-guide/

KingE.Davis. (2018年May月26日). dlib. 检索来源: dlib C++ Library: http://dlib.net/

Lemaignan. (2018年Jan月8日). 3D head pose estimation using monocular vision. 检索来源: GitHub: https://github.com/severin-lemaignan/gazr

LemaignanGarcia, F., Jacq, A., Dillenbourg, P.S.,. (2016). From Real-time Attention Assessment to “With-me-ness” in Human-Robot Interaction. Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference.

Wikipedia. (2018年May月9日). Human head. 检索来源: Wikipedia: https://en.wikipedia.org/wiki/Human\_head