# 侦测走神司机开题报告

## 1 项目背景

绝大多数交通事故的产生都是由于司机的走神，根据美国政府官方网站的调查显示，2015年有3477人因为司机的分心驾驶死亡，391000人为此受伤[1]，在科技迅速发展的互联网时代，驾驶人员更容易做出接打电话、聊天、玩手机等影响自己和乘客安全的行为。

国外关于司机分心驾驶的研究较多，有基于生理测量的机器学习方法检测驾驶员注意力不集中的研究[2][2][2]，也有使用卷积神经网络对司机行为进行分类的研究[1]，还有通过Kinect算法提取眼部特征、胳膊位置和头方向等特征实现分心驾驶检测的研究[3]，相比而言，国内相关研究较少，关于在司机驾驶过程中状态检测的研究大多数为“检测司机疲劳驾驶”，通过人脸识别技术检测人眼，进而分析司机是否为疲劳驾驶[4]，鲜有包含司机聊天、接打电话、玩手机等分心驾驶的研究文献。

随着数据量的增大以及神经网络技术的发展，通过采集司机驾驶过程中的动作，并对司机的动作进行分类是完全可能的，因此希望通过现有技术，训练一个能准确分类司机动作的模型，在司机分心驾驶时进行提醒，避免悲剧的产生。

## 2 问题描述

司机分心驾驶侦测要解决的问题为根据司机驾驶过程中的截图，对司机当前所处的状态进行分类，具体有以下十类：（1）安全驾驶；（2）右手打字；（3）右手打电话；（4）左手打字；（5）左手打电话；（6）调收音机；（7）喝饮料；（8）拿后面的东西；（9）整理头发和化妆；（10）和其他乘客说话。最终给出输入图片分别属于十个分类的概率。概率最高者即为当前司机的状态。

卷积神经网络在图像识别中表现很好，在街道级别的图片中识别牌号达到99.8%的正确率[5]，加上数据集中含有大量图片和类别标签，因此可以采用卷积神经网络结合深度神经网络解决问题。

## 3 输入数据

项目使用StateFarm[6]提供的数据集，包含一个名为img.zip的训练/测试图片集，共有22424张囊括安全驾驶、左右手打字、左右手接电话等类别的RGB图片训练集，且图中司机包含各个种族，具体描述如表1所示，测试集包含79726张各个类别的图片，大小像素类别信息等于训练集相同。还包含一个名为driver\_imgs\_list.csv的文件，该文件记录了训练集图片的司机id、类别和图片名。

表1数据集基本情况

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类别 | 类别名称 | 图片总数（张） | 单张图片宽度（像素） | 单张图片高度（像素） | 详细描述 | 示例图 |
| c0 | 安全驾驶 | 2489 | 640 | 480 | 司机双手握方向盘，目视前方，且无其他类别的行为 |  |
| c1 | 右手打字 | 2267 | 640 | 480 | 司机右手拿有手机 |  |
| c2 | 右手打电话 | 2317 | 640 | 480 | 司机右手拿手机且放于右耳 |  |
| c3 | 左手打字 | 2346 | 640 | 480 | 司机左手拿有手机 |  |
| c4 | 左手打电话 | 2326 | 640 | 480 | 司机左手拿手机且放于左耳 |  |
| c5 | 调收音机 | 2312 | 640 | 480 | 司机调整收音机 |  |
| c6 | 喝饮料 | 2325 | 640 | 480 | 司机喝饮料 |  |
| c7 | 拿后面东西 | 2002 | 640 | 480 | 司机转向后座或单手伸到后面拿东西 |  |
| c8 | 整理头发和化妆 | 1911 | 640 | 480 | 司机看着镜子整理头发或者化妆 |  |
| c9 | 和其他乘客说话 | 2129 | 640 | 480 | 司机未目视前方且与其他乘客说话 |  |

将训练集图片划分为90%的训练集与10%的验证集，使用测试集中的79726张无标签图片作为测试集，上传到kaggle网获得模型最终表现。

由于图片是从视频中截取出来的，相邻帧的图片相似，所以相邻帧的图片不能既出现在训练集里，又出现在验证集里，否则容易使模型过拟合，因此在划分验证集时，不能随机划分，可根据司机id来划分。

通过对driver\_imgs\_list.csv文件的分析，共有26位司机从状态c0到状态c9的数据，经过统计每个司机的图片数及占总数的比例得到下表：

表2标签数据情况表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 司机id | 图片数（张） | 约占总数比例 |
| 1 | p002 | 725 | 0.032331 |
| 2 | p012 | 823 | 0.036702 |
| 3 | p014 | 876 | 0.039065 |
| 4 | p015 | 875 | 0.039021 |
| 5 | p016 | 1078 | 0.048073 |
| 6 | p021 | 1237 | 0.055164 |
| 7 | p022 | 1233 | 0.054986 |
| 8 | p024 | 1226 | 0.054674 |
| 9 | p026 | 1196 | 0.053336 |
| 10 | p035 | 848 | 0.037817 |
| 11 | p039 | 651 | 0.029031 |
| 12 | p041 | 605 | 0.026980 |
| 13 | p042 | 591 | 0.026356 |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 司机id | 图片数（张） | 约占总数比例 |
| 14 | p045 | 724 | 0.032287 |
| 15 | p047 | 835 | 0.037237 |
| 16 | p049 | 1011 | 0.045086 |
| 17 | p050 | 790 | 0.035230 |
| 18 | p051 | 920 | 0.041027 |
| 19 | p052 | 740 | 0.033000 |
| 20 | p056 | 794 | 0.035408 |
| 21 | p061 | 809 | 0.036077 |
| 22 | p064 | 820 | 0.036568 |
| 23 | p066 | 1034 | 0.046111 |
| 24 | p072 | 346 | 0.015430 |
| 25 | p075 | 814 | 0.036300 |
| 26 | p081 | 823 | 0.036702 |

根据计算，第23~26位司机的图片数约为总数的10%，因此选择第1~22位司机数据为训练集，选择第21~26位司机数据为验证集。

## 4 解决办法

使用预训练权重的VGG16网络[7] 作为核心（网络架构如图1），将输入数据处理成网络需要的输入格式，选择合适的评估标准，训练模型，将结果存储并提交到Kaggle。

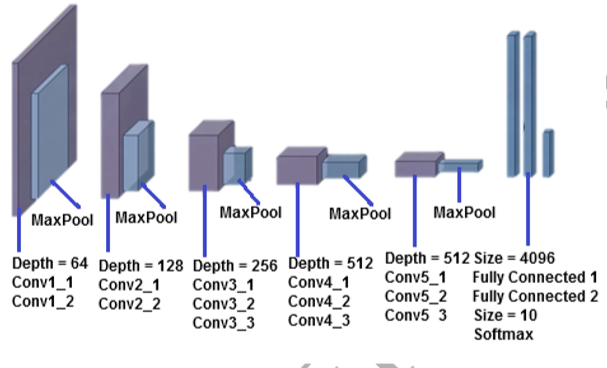


图1 VGG16模型架构

## 5 基准模型

将Kaggle官网上能找到的最高分kernel“statefar\_facepalm\_fork”[8]作为基准模型，该模型的Private Score是1.80188，Public Score是1.55292，大约排名670左右。

该模型的大体思路为：

1. 将司机信息从driver\_imgs\_list.csv中读取到字典里，每一条字典数据key为图片名（xxxx.jpg），value为司机id；
2. 将训练集图片处理成大小为40x50的灰度图，放入训练集X\_train，将图片标签放入y\_train，再将与X\_train的每张图对应的司机id找出来放入driver\_id中，将所有司机的id去重放入unique\_drivers；
3. 将X\_train的shape变为(图片总数，颜色通道，行，列)，即（22424, 1, 40, 50），并除以255进行归一化，y\_train变为10个类别的binary class matrics；测试集数据同样做次处理，得到test\_data和test\_id(图片名称)；
4. 将训练数据中的第1~25位司机数据划分为训练集（id为p049的司机除外），第26位司机数据划分为验证集；
5. 将输入数据输入下述模型，进行拟合；



图2 模型

1. 将训练好的模型用于预测测试数据，将结果保存，组织成Kaggle要求的提交样式。

## 6 评估指标

使用多类对数损失（multi-class logarithm loss）作为评估指标，给出图片属于每个类的可能性，计算公式为

其中N是测试集中图片的数量，M是类别标签的数量，如果i属于j则，否则，是i属于j的概率。

之所以使用mlogloss而非accuracy，是因为准确率相同的情况下，因此mlogloss可以更进一步比较两个模型（mlogloss小的模型更好），粒度更小，所以使用mlogloss作为评估指标。

## 7 设计大纲

使用TensorFlow1.5.0实现模型，因为GPU环境搭建使用了Cuda9.0，只支持tensorflow1.5.0，而keras支持的是tensorflow1.4，通过搜索资料，tensorflow1.5内部集成了keras，因此用tensorflow1.5也可以。

主要实现思路为：

1. 将司机信息从driver\_imgs\_list.csv中读取到字典里，每一条字典数据key为图片名（xxxx.jpg），value为司机id；
2. 将训练集图片处理成大小为224x224x3的图，放入训练集X\_train，将图片标签放入y\_train，再将与X\_train的每张图对应的司机id找出来放入driver\_id中，将所有司机的id去重放入unique\_drivers；
3. 将训练集图片的通道从RGB转为BGR，然后再减去RGB的平均值；
4. 将第1~22位司机数据划分为训练集，第21~26位司机数据划分为验证集；
5. 实现标准的VGG16模型，再加载keras自带的预训练模型（https://www.tensorflow.org/api\_docs/python/tf/keras/applications），作为模型初始权重；
6. 将mlogloss作为评估标准，训练拟合数据；
7. 在训练集上预测数据，将结果存在csv文件中，提交到Kaggle。

如果成绩不满足要求，在数据预处理环节，可以采用更换图片大小，随机催图片裁剪成224x224的方法，或者对图片进行其他可行处理；在模型方面，可以更换预训练模型，或者修改卷积层kernel、步长等超参数来调整模型。

**参考文献**

[1] MASOOD S, RAI A, AGGARWAL A, et al. Detecting Distraction of drivers using Convolutional Neural Network[J]. PatternRecognition Letters (2018), 2017.

[2] SAHAYADHAS A, SUNDARAJ K, MURUGAPPAN M, et al. A physiological measures-based method for detecting inattention in drivers using machine learning approach[J]. Biocybernetics and Biomedical Engineering, 2015,35(3):198-205.

[3] CRAYE C, KARRAY F. Driver distraction detection and recognition using RGB-D sensor[J]. arXivJournal, 2015.

[4] 赵雪竹. 基于AdaBoost算法的驾驶员疲劳检测[D]., 2010.

[5] GOODFELLOW I J, BULATOV Y, IBARZ J, et al. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks[J]. Computer Vision and Pattern Recognition, 2014.

[6] STATEFARM. State Farm Distracted Drivers Dataset[EB/OL]. (2017-06-15)[6.15]. https://www.kaggle.com/c/state-farm-distracted-driver-detection.

[7] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[8] ZVEROLOFF. statefarm\_facepalm\_fork[EB/OL]. https://www.kaggle.com/zveroloff/statefarm-facepalm-fork.