

# 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<sup>[1]</sup>，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<sup>[2][2]</sup>，也有使用卷积神经网络对司机行为进行分类的研究<sup>[1]</sup>，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<sup>[3]</sup>，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<sup>[4]</sup>，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<sup>[5]</sup>，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<sup>[6]</sup>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	编号	司机 id	图片数 (张)	约占总数比例
1	p002	725	0.032331	14	p045	724	0.032287
2	p012	823	0.036702	15	p047	835	0.037237
3	p014	876	0.039065	16	p049	1011	0.045086
4	p015	875	0.039021	17	p050	790	0.035230
5	p016	1078	0.048073	18	p051	920	0.041027
6	p021	1237	0.055164	19	p052	740	0.033000
7	p022	1233	0.054986	20	p056	794	0.035408
8	p024	1226	0.054674	21	p061	809	0.036077
9	p026	1196	0.053336	22	p064	820	0.036568
10	p035	848	0.037817	23	p066	1034	0.046111
11	p039	651	0.029031	24	p072	346	0.015430
12	p041	605	0.026980	25	p075	814	0.036300
13	p042	591	0.026356	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<sup>[7]</sup>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。

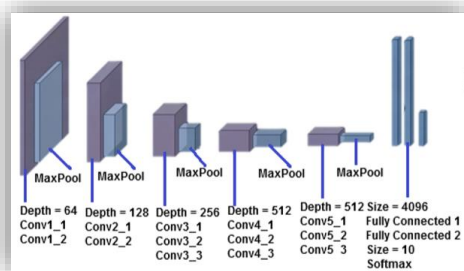


图 1 VGG16 模型架构

## 5 基准模型

将 Kaggle 官网上能找到的最高分 kernel“statefar\_facepalm\_fork”<sup>[8]</sup>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 matrices ; 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 ;

```
def create_model_v1(img_rows, img_cols, color_type=1):
    nb_classes = 10
    # number of convolutional filters to use
    nb_filters = 8
    # size of pooling area for max pooling
    nb_pool = 2
    # convolution kernel size
    nb_conv = 4
    model = Sequential()
    model.add(Convolution2D(nb_filters, nb_conv, nb_conv,
                            border_mode='valid',
                            input_shape=(color_type, img_rows, img_cols)))
    model.add(Activation('relu'))
    model.add(Convolution2D(nb_filters, nb_conv, nb_conv))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(nb_pool, nb_pool)))
    model.add(Dropout(0.6))

    model.add(Flatten())
    model.add(Dense(128))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))

    sgd = SGD(lr=0.1, decay=0, momentum=0.4, nesterov=False)
    model.compile(loss='categorical_crossentropy', optimizer=sgd)
    return model
```

图 2 模型

- (6) 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

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij})$$

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  $y_{ij} = 1$ , 否则  $y_{ij} = 0$ ,  $p_{ij}$  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](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，随机裁剪图片裁剪成 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]. Pattern Recognition 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>.