

# 机器学习工程师纳米学位

## 毕业项目提议-Distracted Driver Detection

---

Yuxiang Jiang  
May 8th, 2017

### 题目

---

Distracted Driver Detection (分心司机检测)

### 背景

我们都有过类似的经历:绿灯亮的时候,在你前面的车辆在前面不让步导致你不能快速通过;或者一个不起眼的车辆突然在你前面减速并且左右摇摆。

当你从这些令你厌恶的车辆旁经过时,你能想到你会看到什么吗?当你看到车里的司机正在发短信,或者似乎沉浸在社交媒体中,又或者正在拿着手机通话时,你一定不会感觉到太惊讶。



根据 CDC 机动车安全部的数据显示,20%的车祸是由司机分心所致。这意味着每年因此导致 425000 人受伤,3000 人死于分心驾驶。

State Farm 保险公司改善这些令人震惊的统计数据,通过仪表盘上的摄像头自动检测司机是否从事了分心驾驶的行为,保证客户安全。State Farm 保险公司提供了 2D 仪表盘相机记录下来的数据集,希望能够对驾驶员的行为进行分类。

### 问题陈述

我们拥有驾驶员的图像信息,这些图像信息记录的驾驶员在车中正在做的动作(发短信、吃东西、打电话、化妆、转身谈话等等)。我们的目标是预测每一个图片的司机在做什么的概率。



10 种预测结果:

- c0: safe driving
- c1: texting - right
- c2: talking on the phone - right
- c3: texting - left
- c4: talking on the phone - left
- c5: operating the radio
- c6: drinking
- c7: reaching behind
- c8: hair and makeup
- c9: talking to passenger

我将会采用深度神经网络的方法，使用 ResNet（微软的深度神经网络模型）的结构，解决这个分类问题。

## 数据集和输入

数据集来源于 State Farm 保险公司，并发布于 kaggle 数据分析比赛网。数据可以从如下网址下载:<https://www.kaggle.com/c/state-farm-distracted-driver-detection/data>

文件描述:

- imgs.zip - zipped folder of all (train/test) images
- sample\_submission.csv - a sample submission file in the correct format

- driver\_imgs\_list.csv - a list of training images, their subject (driver) id, and class id

其中训练集有 22424 张，测试集有 79729 张图，每张照片都是 640x480x3。其中收集了 26 个司机的不同的驾驶状态。

## 解决方法概述

深度神经网络是一种潜在的解决方法。通过大量多层的卷积提取图像中抽象的结构，从而进行识别。VGG16 是一种深度卷积神经网络，我们可以利用它在 imagenet 上预训练的参数，对模型进行更快的训练。然后根据这个模型的表现进行调参，减小过拟合。

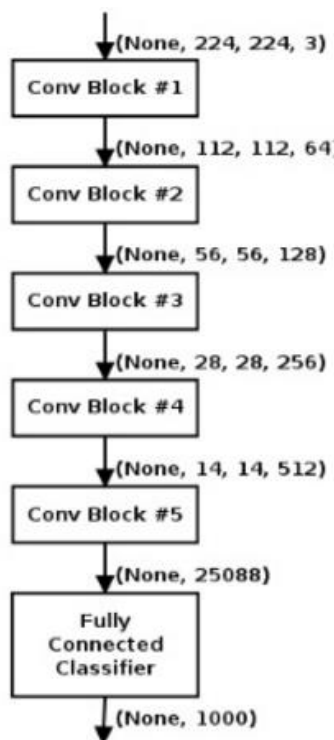
## Benchmark Model（基准模型）

使用 VGG16 即有 16 层的深度卷积神经网络，并使用 imagenet 预训练初始化权重。

VGG16 的结构如下：

### Keras VGG-16 Model

```
( 0, 'input_6',      (None, 224, 224, 3))
( 1, 'block1_conv1', (None, 224, 224, 64))
( 2, 'block1_conv2', (None, 224, 224, 64))
( 3, 'block1_pool',  (None, 112, 112, 64))
( 4, 'block2_conv1', (None, 112, 112, 128))
( 5, 'block2_conv2', (None, 112, 112, 128))
( 6, 'block2_pool',  (None, 56, 56, 128))
( 7, 'block3_conv1', (None, 56, 56, 256))
( 8, 'block3_conv2', (None, 56, 56, 256))
( 9, 'block3_conv3', (None, 56, 56, 256))
(10, 'block3_pool',  (None, 28, 28, 256))
(11, 'block4_conv1', (None, 28, 28, 512))
(12, 'block4_conv2', (None, 28, 28, 512))
(13, 'block4_conv3', (None, 28, 28, 512))
(14, 'block4_pool',  (None, 14, 14, 512))
(15, 'block5_conv1', (None, 14, 14, 512))
(16, 'block5_conv2', (None, 14, 14, 512))
(17, 'block5_conv3', (None, 14, 14, 512))
(18, 'block5_pool',  (None, 7, 7, 512))
(19, 'flatten',      (None, 25088))
(20, 'fc1',          (None, 4096))
(21, 'fc2',          (None, 4096))
(22, 'predictions', (None, 1000))
```



这个模型在使用预训练权重训练 20 次时只有训练集 98%/验证集 60%的准确率。

## 评价标准

结果使用 “multi-class logarithmic loss” 来评估。每一个图片已经被分为其中一个类。 对于每一个图片，将会有每一种类别的预测概率。 评估公式如下：

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

其中  $N$  是测试集中的图片数量， $M$  是图片的分类标签的数量， $\log$  是自然对数，如果观测到的  $i$  属于类别  $j$ ， $y_{ij}$  是  $1$ ，否则为  $0$ 。 $p_{ij}$  是观测到的  $i$  属于  $j$  的概论。

其中一个给定的图片的概率和不需要等于  $1$ ，因为它们在评分前被重新调整过（每一行被除以行的和）。

## 项目设计

工作流程：

- 导入数据
- 数据样例显示
- 数据预处理
- 设计训练模型
- 模型可视化
- 训练模型
- 计算准确率
- 预测样本和测试集
- 保存模型