

# 分神司机检测

——机器学习纳米学位开题报告  
李佳朋

## 问题的定义

### 项目概述

项目源自 State Farm 保险公司于 2016 年在 Kaggle 数据竞赛平台上举办的比赛——State Farm Distracted Driver Detection[1]。根据 CDC 机动车安全公司对于交通事故的调研报告，大约五分之一的交通事故是由于司机在驾驶过程中分神导致的，这样的交通事故每年会造成 425000 人受伤，多达 3000 人死亡，对人们的生命财产造成重大损失的同时也会使得保险公司支付大量的赔偿金额。

为了通过机器学习等技术准确地对司机驾驶过程中的分神行为（如：打电话、喝水等）进行分类，并针对性的对具有不同驾驶习惯的司机进行投保，State Farm 公司联合 kaggle 数据竞赛平台举办了这场比赛。

本项目属于有监督学习中的分类问题，具体为图片分类问题。为采用机器学习方法去求解这个问题，比赛的主办方提供了包括 22424 张尺寸为 640\*480 的有标注训练图片集和包括 79726 张同样尺寸图片的测试图片集，其中训练集图片分为 10 个类别，分别对应着司机驾驶过程中的不同行为，会在下文中进行详细说明。

### 问题陈述

该项目旨在通过训练集中记录司机驾行为的图像作为有标注数据来训练统计模型，然后利用该模型对未标注的司机驾驶过程中的行为图片进行检测，达到将图像精准地分类到 10 个类别的目的，在本项目中这 10 个类别分别被编号为 c0-c9，每个编号对应的驾驶行为如下：

- C0：正常驾驶
- C1：右手打字
- C2：右手打电话
- C3：左手打字
- C4：左手打电话
- C5：操作收音机
- C6：喝水
- C7：向后伸手
- C8：整理头发和化妆
- C9：和乘客说话

综上，该问题可以被归为计算机视觉中的图片分类领域，该领域中有一系列现存的成熟的方法来解决此问题，在这些方法中深度卷积神经网络在图片分类上有最好的表现，所以在

此项目中拟采用该模型。

## 评价指标

为量化基准模型和解决方案，采用 Kaggle 比赛 Private Leaderboard 的得分作为评估指标。该得分是，利用服务器上测试集的真实标签，对 69%比例的测试集计算得到的多类对数损失。

其中，多类对数损失公式如下：

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

式中，N 为样本数量，M 为类别数量， $y_{i,j}$  为表征样本 i 是否实际属于类别 j 的 0-1 指示函数， $p_{i,j}$  为样本 i 在类别 j 上的预测概率。将该评价标准用于本项目是合理的，一方面，由公式 1 可知，该评价指标可以用于多分类问题；另一方面，最小化 logloss 的过程就是在最大化模型预测值与样本真实值的似然函数的过程[2]。综上，该指标适用于本项目，并且能够有效地指导模型的优化。

## 分析

### 数据的探索

官方提供的数据包括训练数据，测试数据和训练数据列表三部分，其中训练数据包含 22424 张有标注图片，这些图片较为均匀地分布在上述的 10 个类别之中，测试数据包含 79726 张未标注图片，这些图片是我们需要去分类的。训练照片列表中有三栏数据，分别是人员编号 (subject)，图片所属类别 (classname) 和图片文件名称 (img)，值得注意的是人员编号并不是连续的，而是 p002~p081 这个区间中的 26 个编号，也就是共计采集了 26 人的驾驶行为图片。这些图片的获取方法：State Farm 公司将这些志愿者放入汽车的驾驶室，使用拖车拖动汽车在公路上行驶，与此同时让志愿者在驾驶室中做出一系列的指定动作（如上述 c1-c9，10 个分类中描述的那样）并将整个过程通过视频进行记录，最后将视频中的每一帧图片取出，然后根据时间区间将视频片段标注为不同行为类别，属于这段视频的每一帧图片都被归入到相应的分类中作为有标注的训练数据，因为司机的行为和指令不能完美匹配，所以这种对图片进行标注的方法会存在一些错误标注的现象。

为了评价模型的效果，我们需要将全部训练数据分为新训练数据集和验证集两部分，其中新训练数据集用于训练模型，验证集用于验证模型的分类效果并以此为根据来调节学习率，训练代数等超参数。测试数据是我们要去分类的图片，我们要使用在训练集中学习得到的模型去分类测试集的数据得到最终结果。因为测试集中所有司机都未在训练集中出现过，所以我们在划分验证集的时候也要保证训练集和验证集没有交叉样本，这样才能使验证集得到的评估指标尽量客观。

## 探索性可视化

从训练集 10 个分类的图片中每一类随机选取一张图片组合在一起得到图 1。



图 1 各分类司机姿态图

图 1 中 10 张图片分别对应着 c01-c09 这 10 个类别，通过对每一个类别的图片进行观察可以发现：

- 不同类别的图片之间有许多相似信息，例如：因为测试车辆相同，所有图片的车内环境都是相似的；来自 c0 分类的女性驾驶者和来自 c1 分类的女性驾驶者具有相似的发型和衣服的颜色；来自 c2 和 c5 的男性驾驶者是同一人，所以肤色以及着装完全相同。
- 用于区分各个类别的关键信息主要是司机的面部信息和手部信息，尤其以手部信息最为关键，因为 10 张图片中每个类别司机的手部动作都不一样。

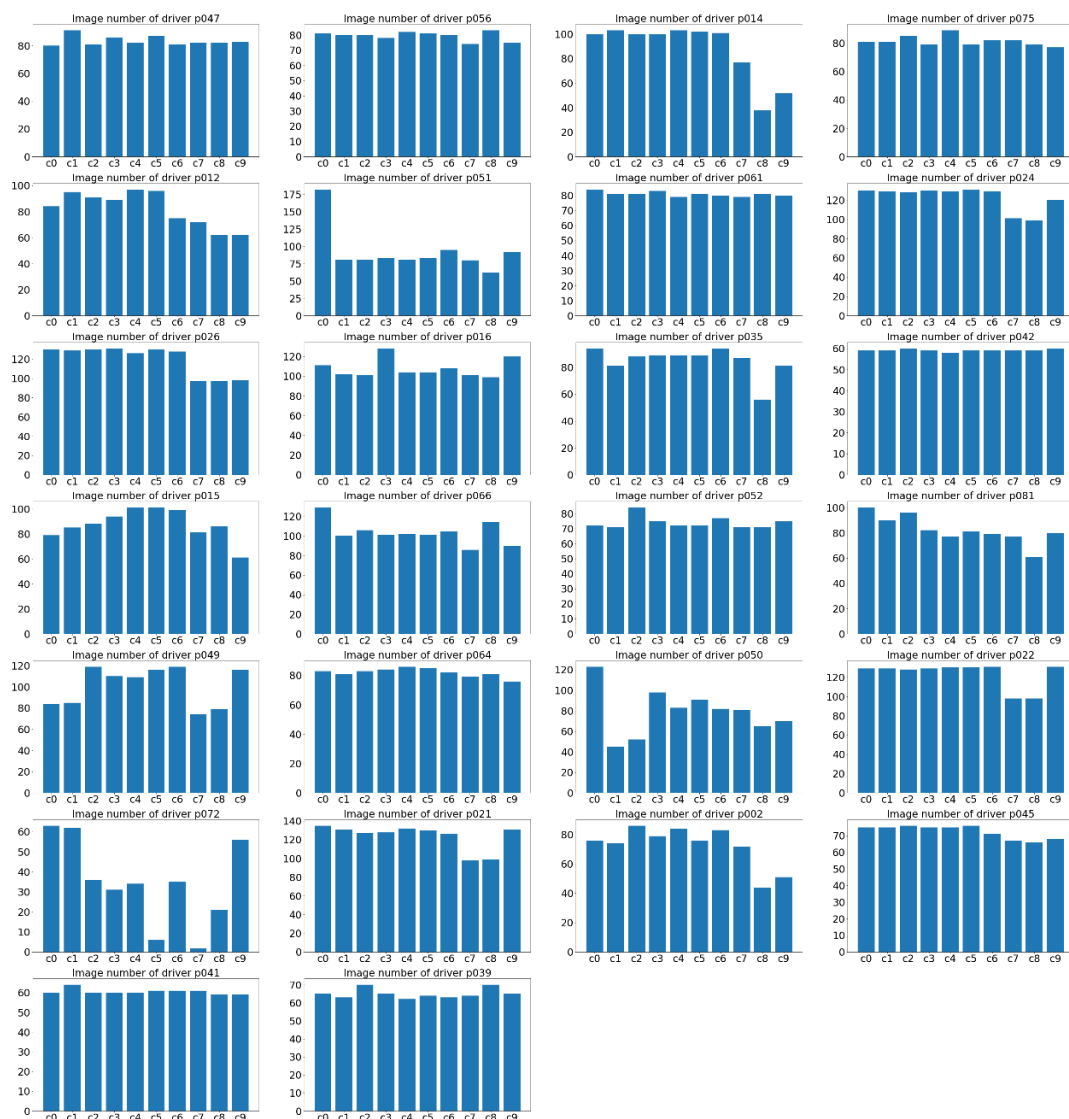


图 2 各个分类样本数量统计图

对训练集中 26 个司机的图片按照图片所属类别对其数目进行统计，绘制条形统计图如图 2 所示。图 2 由 26 个条形统计图构成，其中每个条形统计图对应一个司机，司机对应编号在条形统计图的 title 中标出，每个条形统计图中有 10 个条形，每个条形对应的横坐标代表其所属类别，纵坐标代表属于该类别的图片数目。

从 26 张条形图的数据分布可以看出，只有少数司机（p072）的图片没有均匀分布在 10 个类别中，所以训练数据中各个类别的图片数量较为均衡，可以免去重采样的步骤。

## 算法和技术

卷积网络（CNN）是一类尤其适合计算机视觉应用的神经网络，因为它们能使用局部操作对图片特征进行分层抽象。有两大关键的设计思想推动了卷积架构在计算机视觉领域的成功。第一，CNN 利用了图像的 2D 结构，并且相邻区域内的像素通常是高度相关的。因此，CNN 就无需使用传统神经网络中所有像素单元之间的一对一连接，而可以使用分组的局部连接。第二，CNN 架构依赖于特征共享，因此每个通道（即输出特征图）是在所有位置使

用同一个过滤器进行卷积而生成的。

鉴于上述 CNN 在计算机视觉领域的优势，本项目的算法都是采用基于 CNN 的网络架构。使用 CNN 解决此问题的方案可概括为如下步骤：

- 1、划分训练数据集和验证数据集
- 2、选取合适的迁移学习模型，并根据所选模型的特性对输入图片做相应的预处理
- 3、将所选迁移学习模型的 top layers 去掉并接入全局平均池化层和全连接层，使用训练数据对模型进行训练，使用验证集数据调整超参数，进行 fine-tune
- 4、当所选模型都能够达到一定的分类精度时将模型得到的结果进行集成
- 5、进行可视化操作

## 基准模型

采用经 ImageNet 数据集训练的 VGG16 [3]模型作为基准模型，该模型在总体结构上沿用了 AlexNet 中卷积-池化-全连接的基础架构，只是将卷积核的尺寸缩小为 3\*3，同时增加网络的深度，从而达到了更好地对图片特征进行抽象的目的。VGG 是 2014 年 ILSVRC 竞赛中分类比赛的第二名，适用于本项目的分类任务，因此考虑使用 VGG16 作为基准模型。使用方式如下。

- a) 构建网络，载入在 ImageNet 上训练的 VGG16 模型，去掉尾部的全连接层并加入全局平均池化层，将 VGG16 的卷积层输出的 512 张特征图以图为单位取平均，变成 512 个特征值，随后接入 10 个节点的全连接层，使用 softmax 激活函数用以输出图片被归为 10 个类别的概率。使 VGG16 的卷积层权重保持其在 ImageNet 上训练的值不变，只对最后的全连接层进行训练
- b) 划分训练集和验证集，将 26 个司机中的 3 个的图片取出作为验证集
- c) 数据预处理，按照论文中的方法，输入图片尺寸为 224\*224，减去 ImageNet 训练集图片的 RGB 均值
- d) 使用上文中介绍的最小化 logloss 作为优化目标，RMSprop 作为优化算法，学习率设置为 1e-5，分类准确率作为评价指标

开始训练，5 个 epoch 之后得到如下准确率和 logloss 的变化图：

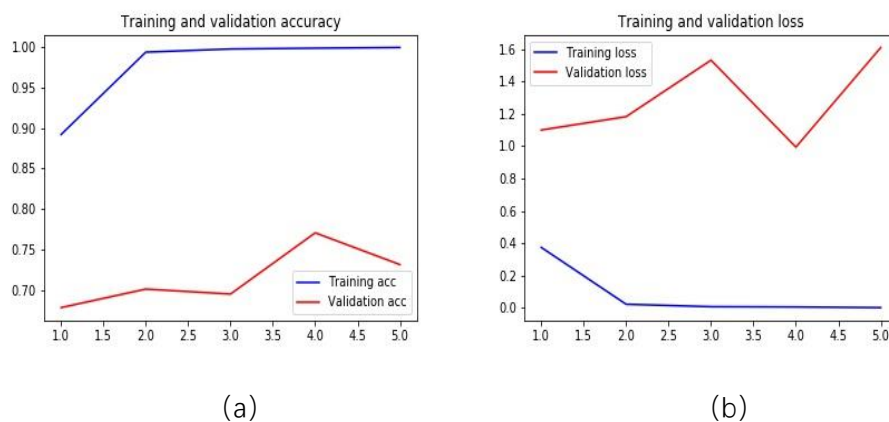


图 3 VGG16 精度图 (a)，损失图 (b)

从图 3 (a) 中我们可以看出模型在训练集上的准确率在第二个 epoch 就达到了 99%，而在验证集上的准确率是 70%左右，同时 (b) 图中验证集的损失函数值出现了增长，所以模型出现了过拟合现象。又经过两个 epoch 验证集的准确率提高到了 78%，同时 (b) 中的

loss 值也下降了，而到了第 5 个 epoch 又出现了更加严重的过拟合现象，所以将训练过程回滚到第 4 个 epoch，并使用相应的模型参数进行预测。使用该模型在测试集上的分类结果在 kaggle 上提交能够得到 1.04588 的 Private Score。

## 方法

本项目使用深度卷积神经网络进行图片分类, 拟采用在该领域分类效果较好的且具有代表性的三个模型：ResNet [4]、InceptionV3 [5]和 Xception [6]，在对项目提供的数据集进行观察之后发现该数据集规模不足以训练上述三个较为大型的网络，所以本项目使用迁移学习，利用上述三个模型在 ImageNet 中预训练的权重，然后使用本项目的数据集进行 fine-tune，在三个模型均达到比较好的效果之后再对三个模型的分类值进行融合，得到最终的分类结果。

## 数据预处理

首先以司机为单位划分训练集和验证集，由于我们使用了三个不同的模型，对于每个模型在训练前我们都对训练集和验证集进行一次划分，每次从 26 个司机中随机选取 3 个作为验证集其余司机作为训练集。

由于本项目提供的数据集中的图片尺寸是 480\*640，需要根据所使用的模型不同做出相应的调整，对于 ResNet 使用 224\*224 的输入图片尺寸，对于 InceptionV3 和 Xception 均使用 299\*299 的图片输入尺寸。

## 项目设计

项目的总体流程如图 4 所示：

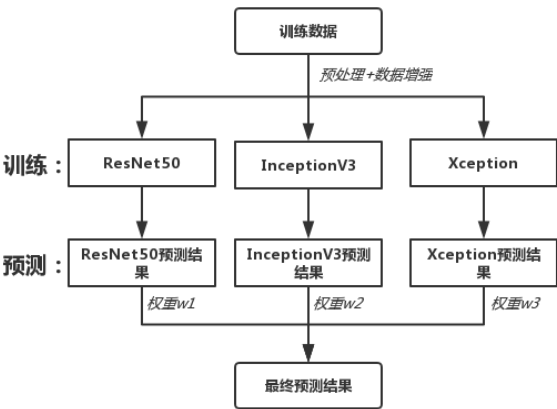


图 4 项目结构图

上述三个模型中 ResNet50 共有 174 层，23608202 个参数，InceptionV3 有 311 层，23626728 个参数，Xception 有 132 层，22855952 个参数，都是非常大规模的网络。由于本项目中提供的数据量不足以训练这三个较大的模型，以上三个模型均载入了在 ImageNet 数

数据集上的预训练权重,同时开放一定数量的尾部层,使用现有数据集中的图片进行 fine-tune。

由于在基准模型 VGG16 的训练过程中出现了严重的过拟合现象,初步分析造成过拟合现象的原因主要有两点,第一是训练数据不足,第二是模型过于复杂,所以在项目设计过程中分别在数据处理和模型结构上进行了一些优化。

一方面,使用了数据增强来增加训练数据量,三个模型对应的数据增强操作有:图片旋转、横向平移、纵向平移、剪切变换和随机缩放 5 种,其具体参数值在下一节中进行详细讨论。

另一方面,在全局平均池化层和输出层之间添加了 Dropout 层,丢弃概率设置为 0.5,用来进一步缓解过拟合现象。调整结构之后,每个子模型的最终结构都可以用图 5 描述:

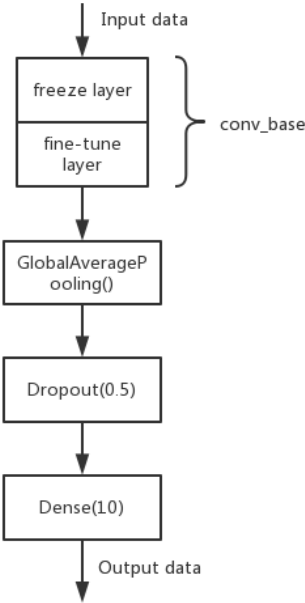


图 5 Resnet50, InceptionV3 和 Xception 模型结构图

## 执行过程

在完成模型结构设计之后就是如何去实现这个模型了,下面对具体实现过程进行详细描述。本项目使用开发环境以及配置如表 2 所示:

表 1 配置信息

操作系统	Win10
Python 版本	Python 3.6.2
包与环境管理器	Anaconda
深度学习框架前端	Keras 2.0
深度学习框架后端	Tensorflow 1.1.0
GPU 架构	Cuda 8.0
GPU 加速库	CuDnn 6.0

对于所选取的 3 个子模型 ResNet50, InceptionV3 和 Xception,均采用如下执行过程:

1. 载入模型以及 ImageNet 上预训练的权重
2. 去除载入模型的 top layer 作为 conv\_base,并按照图 5 中的结构搭建新模型



3. 将 conv\_base 中的权重冻结，开放后接入的全连接层权重进行训练
4. 开放 conv\_base 中一定数量的尾部层对其权重进行 fine-tune
5. 观察每一个 epoch 之后模型在验证集上的表现来确定训练周期，并保存模型
6. 若出现验证集 loss 上升则将模型回滚到上一个 epoch，作为最终子模型
7. 将所有子模型结果进行融合后得到最终结果

下面阐述这样执行算法的原因，以及在执行过程中遇到的问题：

由于深度神经网络在开始的层主要用来提取较为浅层的特征，这些特征具有较强的泛化能力，因此即使 ImageNet 上的数据集与本项目中数据集大相径庭，其训练出的初始权重对本项目仍然具有很强的指导性，又因为我们后接入的全连接层的权重是随机初始化的，如果将模型全部开放训练，随机初始化的权重会对预训练权重进行更改，导致预训练权重被破坏。为了避免发生这种现象我们在上述步骤 3 中单独对全连接层的权重进行训练。

问题 1：步骤 5 在训练过程中出现了训练集 loss 值持续居高不下的情况，一直维持在 2.5 左右，经过分析与实验证明，确定导致这种现象的原因是学习速率过大，后将学习速率由  $1e-3$  调整至  $1e-4$  后得到解决。

问题 2：出现 loss 值无法下降的原因是没有对训练数据进行混洗，导致各个分类的训练数据不能在每一个 batch 中均匀分布，模型在一定阶段只能学习去识别某个类别，效果自然不好，该问题在正确混洗数据之后得到解决。

模型执行的过程就是反复使用不同的参数执行上述步骤 1~6 的过程，这个过程中需要调节的参数主要有如下几种：

1. Fine-tune 层数
2. 数据增强操作的参数，包括图片旋转角度，横向平移量和纵向平移量等
3. 优化方法的学习速率
4. 训练周期 epoch

调参的依据是模型在验证集上的准确率，经过多次调节确定 3 个子模型最优参数如表 3 所示，最优结果是经过若干组参数测试之后得到的，测试的参数组合有很多这里就不一一列出，只将其中一组作为代表在表 2 中给出。

表 2 中间参数组合及准确率

操作类型\模型名称	ResNet50	InceptionV3	Xception
Fine-tune 层数	70	196	86
rotation_range	20	10	20
width_shift_range	0.1	0.05	0.1
height_shift_range	0.1	0.05	0.1
shear_range	0.2	0.2	0.2
zoom_range	0.2	0.2	0.2
learning rate	$1e-4 \sim 1e-5$	$1e-4 \sim 1e-5$	$1e-4 \sim 1e-5$
epoch	3	3	3
Valid accuracy	87.42%	86.10%	90.48%

表 3 最优参数组合及准确率

操作类型\模型名称	ResNet50	InceptionV3	Xception
-----------	----------	-------------	----------



Fine-tune 层数	77	96	20
rotation_range	10	10	20
width_shift_range	0.05	0.05	0.1
height_shift_range	0.05	0.05	0.1
shear_range	0.1	0.2	0.2
zoom_range	0.1	0.2	0.2
learning rate	1e-4~1e-5	1e-4~1e-5	1e-4~1e-5
epoch	3	3	4
Valid accuracy	90.42%	89.63%	93.78%

## 完善

在模型调试的过程中通过解决上一节中描述的问题和调节模型参数使得模型的效果有了显著提升，但是单一模型依然难以达到满意的效果，鉴于此，本项目决定将各个子模型的预测结果结合来进一步完善模型的性能。

深度神经网络由于具有超多的参数，往往容易过拟合训练数据。同时由于上述各个子模型的模型结构，训练数据与验证数据划分方式不同等原因，会导致它们从不同的方向去过拟合训练数据集，因此一个缓解这种过拟合的方法就是将各个子模型得到的预测结果进行融合，本项目采用的是求平均的方法，将上述各个最优子模型的预测结果进行多次融合得到最终预测结果。

## 结果

### 模型的评价与验证

表 3 中对应的模型参数能够得到上述 3 个子模型的最优 Private Score 以及融合后的最终 Private Score 如表 4 所示。

表 4 子模型最优 Private Score

操作类型\模型名称	ResNet50	InceptionV3	Xception
Best Private Score	0.29806	0.31699	0.28532
Best Merge Score	0.185		

上述结果均为在 kaggle 上的提交结果，其中经过融合之后的最好 Private Score 可以排进 Leaderboard 的 TOP50。

merge_pred.csv 13 days ago by JiapengLi <a href="#">add submission details</a>	0.18598	0.21884	<input type="checkbox"/>
--	---------	---------	--------------------------

# 合理性分析

相比较基准模型 Vgg16 得到的 1.04588 的 Private Score，最终模型的结果达到了 0.185 的 loss，可以说是较好地发挥了预训练模型的潜力，借助了迁移学习的优势，很好地解决这个项目司机驾驶行为分类的问题。

# 项目结论



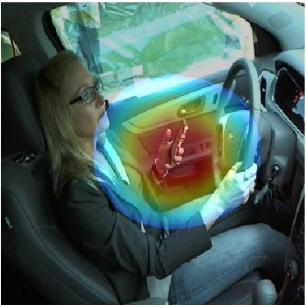

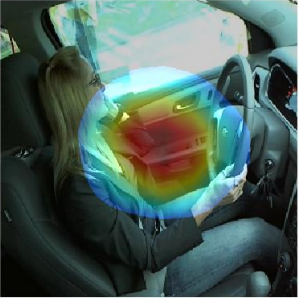

## 结果可视化

为了对训练出来的分类模型的工作原理有一个直观的理解，同时也为了进一步验证模型的合理性，本项目参考 Bolei Zhou 等人 [7]在 2016 年发表在 CVPR 的文章中的方法对模型进行可视化，核心思想是将模型各层的激活值由输出层的神经元反向映射回卷积层，然后以卷积层的激活值为基础绘制热力图与原图像叠加显示，用热力图的不同颜色表示原图像中不同区域对模型最终输出的影响，具体操作步骤如下：

- 1) 对于每一个分类分别找出一张模型预测正确与预测错误的图片，如果在该分类中不存在预测错误的图片则跳过。
- 2) 使用模型对该图片进行预测过程中产生的三个张量：尾部卷积层激活值 conv\_out 取出（对于本项目使用的 Xception 模型来说，是一个 10\*10\*2048 的张量），输出层的权重 weight（2048\*10 的张量），输出层预测值 prediction（10\*1 的张量）。
- 3) 计算 conv\_out\*weight\*prediction 作为热力图的基准值，是 10\*10\*1 的张量，并将这 100 个像素点的值归一化，为了去除噪声干扰将小于 0.2 的值置零。
- 4) 将 step 2) 中的 10\*10 张量缩放到模型输入尺寸与原图像叠加显示，图片中亮度越高的区域对模型最终输出的激活值贡献越大。

表 5 Xception 子模型在各个分类中预测（正确/错误）CAM 可视化

	预测正确	预测错误
安全驾驶	<div>Valid: safe driving   Pred: safe driving 67.94%</div> 	<div>Valid: safe driving   Pred: talking 95.85%</div> 

<p>右手打字</p>	<p>Valid: texting - right   Pred: texting - right 97.95%</p> 	<p>Valid: texting - right   Pred: phone - left 19.54%</p> 
<p>左手打字</p>	<p>Valid: texting - left   Pred: texting - left 100.00%</p> 	
<p>右手打电话</p>	<p>Valid: phone - right   Pred: phone - right 99.90%</p> 	
<p>左手打电话</p>	<p>Valid: phone - left   Pred: phone - left 99.63%</p> 	
<p>向后伸手</p>	<p>Valid: reaching behind   Pred: reaching behind 99.99%</p> 	






整理头发/化妆	Valid: hair and makeup   Pred: hair and makeup 99.15% 	Valid: hair and makeup   Pred: drinking 96.09% 
说话	Valid: talking   Pred: talking 97.98% 	Valid: talking   Pred: safe driving 59.38% 
操作收音机	Valid: operation radio   Pred: operation radio 99.96% 	

表 5 中空白的区域代表模型在该分类中没有预测错误的图片，可见模型对验证集中的大部分图片都能做出正确的预测。通过对表 5 的观察可以发现热力图的高亮区域大多集中在司机的手部，可知模型主要通过观察司机的手部动作来判断司机现在的驾驶状态属于哪个类别。

另外也可以观察到验证集中存在 一些标注错误的图片，例如表中第 8 行第 3 列的图片中司机明显是在饮水，但却被标注为整理头发和化妆，因此我们可以推测训练集和测试集中也存在标注错误的现象，这也是导致模型准确率难以提升的原因之一。表中第 3 行第 3 列的图片热力值比较平均，证明模型没有找到重点区域，这也是导致这张图预测错误的原因。

## 对项目的思考

没有一个万能的模型，只有适合问题的方法。在尝试解决本项目问题的过程中我参考了很多别人的其他项目，比如猫狗分类项目[8][9]等，这些项目的作者同样选择 VGG，ResNet 等流行的模型，取得了很好的效果，但是同样的做法却并不适用于本项目。例如：在猫狗大战项目的数据扩增时，旋转角度参数往往可以选取 40，因为猫和狗在图片中的位态更加多变，测试集中也可能大量存在旋转角度很大的猫和狗的图片。但是对于本项目来说，司机在驾驶室中的位置大多比较固定，这时如果使用 40 的旋转角，就很容易产生一些噪声数据，干扰模型的训练过程，因此要适当减小旋转角度，调参实验过程也证实了这个分析，10-20 的旋转往往能得到更好地效果。



总而言之, 在解决每一个问题的时候我们都不能奢望将数据直接丢给一个模型就能得到很好的效果, 而是要结合问题本身的特性去设计一些有针对性的解决方案, 比如根据数据量大小和数据集相似性去确定是否使用迁移学习, 若使用需要开放多少层去做 fine-tune。

在完成本项目的过程中对项目中涉及的模型 (VGG、ResNet、Inception 和 Xception) 的主要思想以及工作原理都有了较为深刻的认识, 也发现了训练多个模型然后对预测结果进行融合能够取得比反复调试单个模型更加显著的提升效果。

## 改进

从训练数据的获取方式中我们可以得知训练数据中许多不同的图片都来自于同一个人, 这就导致了这些图片中的大部分信息是相同的, 例如司机的肤色特征, 着装颜色以及车内环境等, 这些信息与司机驾驶行为没有显著联系, 所以在我们分类司机的驾驶行为时都属于冗余信息同时又占据了图片中的大部分像素点, 因此记录一个人不同驾驶行为的图片很可能因为这些相同的冗余信息而被神经网络归为同一类。而在本项目中我们希望在训练数据中学习的模型具有很好的泛化能力, 即能够将未见过的司机的驾驶行为准确分类。所以我们要根据训练照片列表以不同的司机来分割训练数据和验证数据, 这样模型在验证集上的表现才能客观的反映模型的效果。

其次, 既然司机的肤色特征, 着装颜色以及车内环境等信息是我们不关心的, 那么相应的在分类过程中我们应该重点关心的就是司机的面部特征和手部动作等与驾驶行为关系较为密切的信息, 所以在上述步骤 3.2 中的模型输入拟采用整张图片与图片中面部和手部截取出来的子图片相结合的方式, 用以强调图片中我们关心部分的信息。

基于以上考虑, 将整张图片的左上部 (像素区间: [100,50,350,200]) 与右下部 (像素区间: [400,200,600,350]) 截取出来, 如图 1 所示:



图 1

从图中可以看出左上部包含司机的脸部, 右下部显示手是否握在方向盘上, 这两部分的图片信息对于 10 个类别的判断起到重要作用, 所以将这两个部分截取出来与原图片分别传入 vgg16 来提取图片特征, 随后将 3 张图片提取的特征组合并与对应的标签结合作为新的训练

数据用来训练神经网络。

但上述改进方案取得的效果并不理想, 模型融合之后得到最佳 Private Score 为 0.8965。分析原因如下: 因为我们采用了截取图片的操作, 所以原本数据集中的每张图就对应了三个输入, 为了将三张图片的特征结合进行预测我们只能使用 ImageNet 上训练好的模型去提取特征, 将三张图的特征进行拼接之后再输入到全连接网络进行预测, 这就失去了对卷积层调参的过程, 又因为 ImageNet 数据集于本项目的数据集相差较大, 所以直接使用原本模型的权重并不能有效地提取最关键的特征, 导致取得的效果不佳。

## 参考文献

- [1] [State Farm Distracted Driver Detection \(Kaggle\)](#)
- [2] [A Friendly Introduction to Cross-Entropy Loss](#)
- [3] VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION
- [4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [5] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
- [6] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." arXiv preprint (2016).
- [7] Zhou, Bolei, et al. "Learning deep features for discriminative localization." Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. IEEE, 2016.
- [8] Chollet, Francois. *Deep learning with Python*. Manning Publications Co., 2017.
- [9] [杨培文, 手把手教你如何在 Kaggle 猫狗大战冲到 Top2%](#)