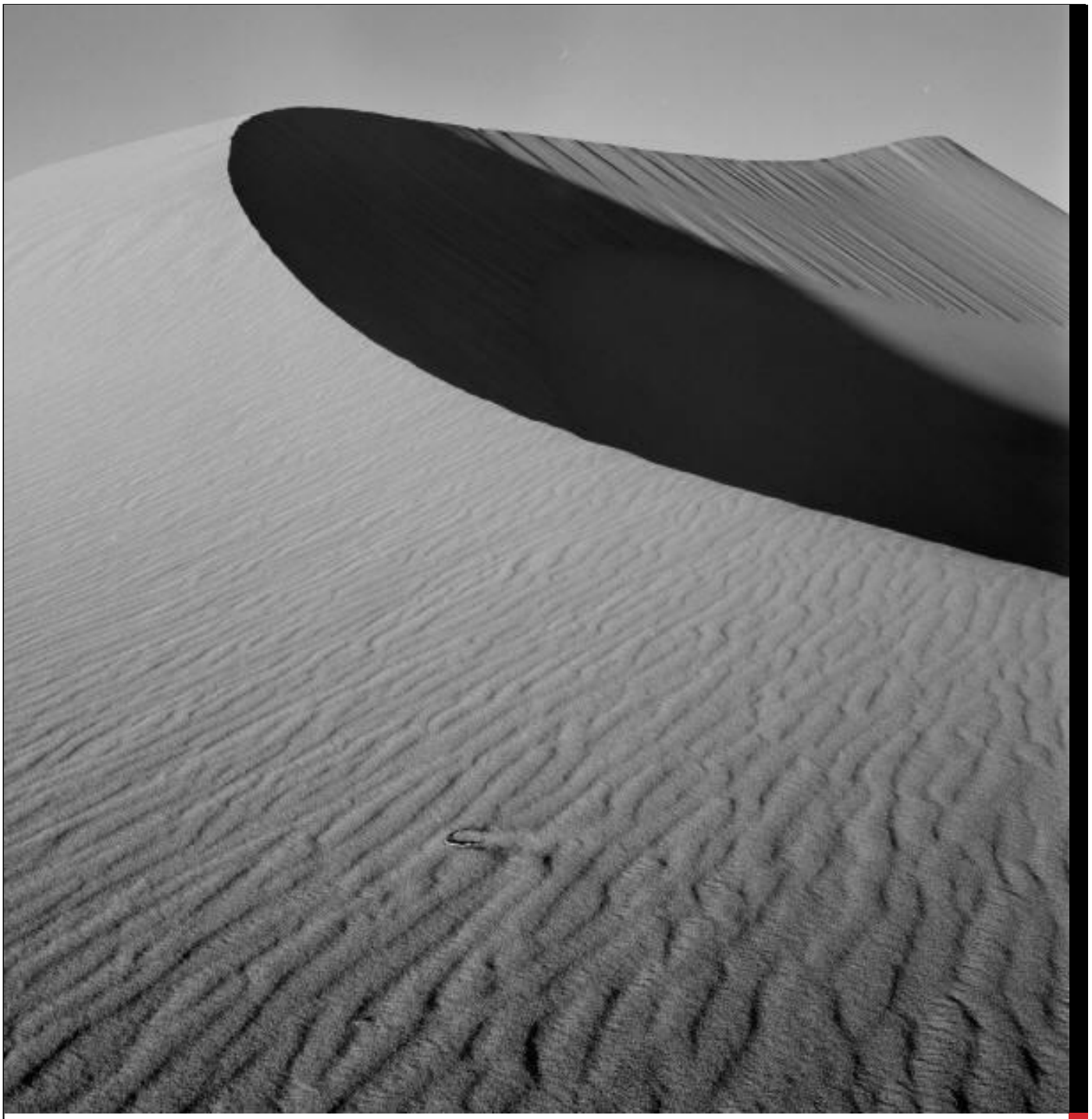


侦测走神司机

UDACITY 机器学习毕业项目

施梦杰

这个项目利用深度学习，通过对司机的照片进行司机状态的分类，来检测当前司机的驾驶状态。

目录

1 问题的定义

1.1 项目概述

每年都有数以万计的人死于交通事故，其中人为原因占绝大部分。开车走神，疲劳驾驶这些不经意的行为都有可能造成严重的后果。这些情况可能会导致车辆的突然加速、减速，行驶状态不及时避让。这些行为对行车安全埋下了重大的隐患。

中国社科院与某保险机构的联合调研显示,道路交通风险最大的是分心驾驶,分心驾驶主要表现在疲劳驾驶和开车使用手机等方面,是道路安全的头号潜在杀手。

这个项目的目标是，通过车载的摄像头对司机的行为进行自动监测，通过这样的方式去改善分心驾驶的隐患。

我们可以将这个问题转换成用深度学习模型对图片进行分类的问题，在该领域已经获得很多成果。比如将数据集（CIFAR-10）图片十分类的问题，优秀的深度学习模型的正确率可以达到 95%（如 Fractional Max-Pooling）。我们可以参照优秀的深度学习模型来建立分类司机状态照片的模型来完成这个项目。

1.2 问题陈述

通过车载的摄像头，按一定的频率截取照片，然后对照片上司机的状态进行分类，我们检测的状态有一个正常的驾驶状态和九个危险驾驶的状态。如果是安全驾驶状态，则维持正常状态。如果监测到司机有走神的行为，则发出警报，提醒司机。

我们侦测的走神行为有这么几种：驾驶过程中打字、打电话，调节收音机，喝饮料、拿后座东西，整理头发和化妆以及同其他乘客说话。

1.3 评价指标

我选用的评估指标为 log loss。

kaggle 评估的方式是 muti-class logarithmic loss，公式如下：

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

这个评估方式，对不同信心的预测惩罚不一样，比如更大信心的错误判断会比稍微小一点的信心的错误判断的惩罚更大，而更大信心的正确判断的惩罚会比更小信心的正确判断更小。不适用准确率作为评估标准，是因为准确率作为评估标准是不如 $LogLoss$ 有说服力的，如果一个模型对司机的一种状态有 99%的判断，另一种模型只有 60%的判断，这样准确率是相同的，但是$LogLoss$的区别会很大，这时候虽然无法用准确度区分模型的好坏，但是可以用 $LogLoss$ 来判断，因此我选用 $LogLoss$ 作为评

估标准。此外，另外测试集没有 label，只能通过提交到 kaggle 得到测试集的 loss，因此使用准确率也是不具备可操作性的。

2 分析

2.1 数据的探索

输入数据来自于 kaggle（数据下载）

这个页面中提供了三个文件：

driver_imgs_list.csv

这个文件中，包含了训练数据中，每个图片文件名，司机的编号以及这张图中司机的状态。

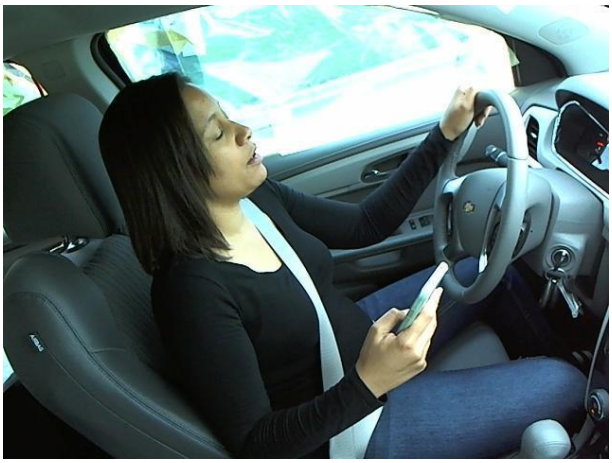
Imgs.zip

这个文件解压出两个文件夹，其中 train 中包含 10 个子文件夹，意为 10 个 class，代表不同的司机状态，已经分类成 c0-c9）。test 文件夹中是提供测试评分的测试集，共有七万多张测试图片。

sample_submission.csv

这个文件是提交在 kaggle 上测试数据预测结果的样本。

接下来，我抽样了一些样本，简要地看一下提供的数据集。

c0：安全驾驶 	c1：右手打字 
c2：右手打电话	c3：左手打字

c4：左手打电话

c5：调收音机

c6：喝饮料

c7：拿后面的东西

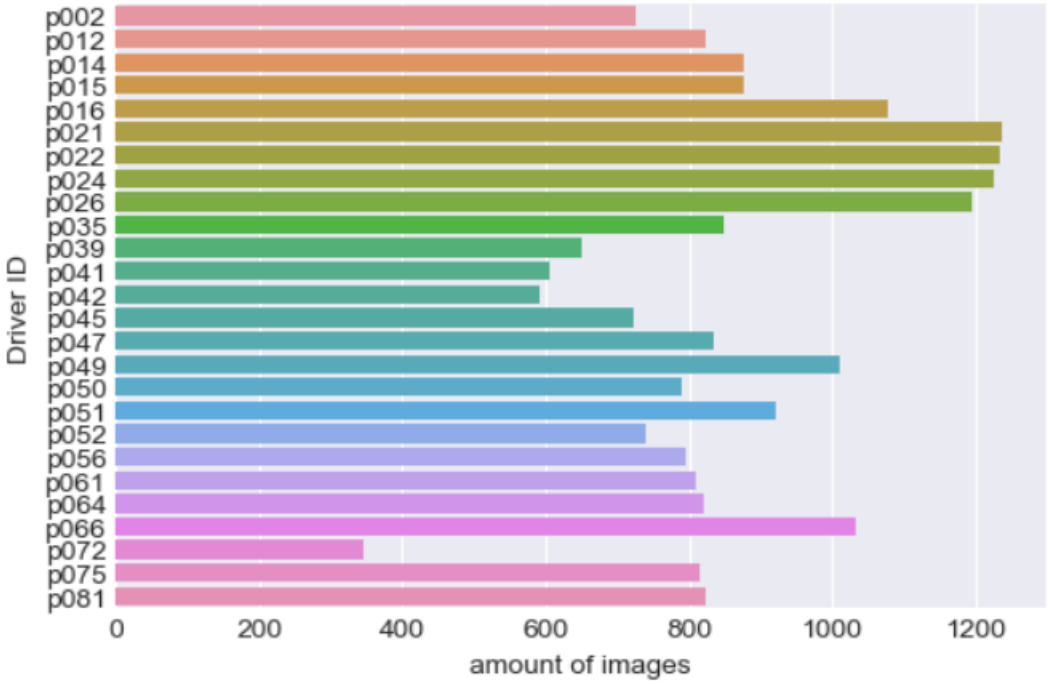
c8：整理头发和化妆

c9：和其他乘客说话

我们可以看到，这些图片是通过副驾驶上方位置的摄像头拍摄的，司机是有重复的。为了进一步了解数据集，我将对数据进行可视化。

2.2 数据可视化

通过分析 driver_list_imgs 文件，可以知道总共有 26 位司机，22424 张图片，下面是每位司机的照片数量分布。

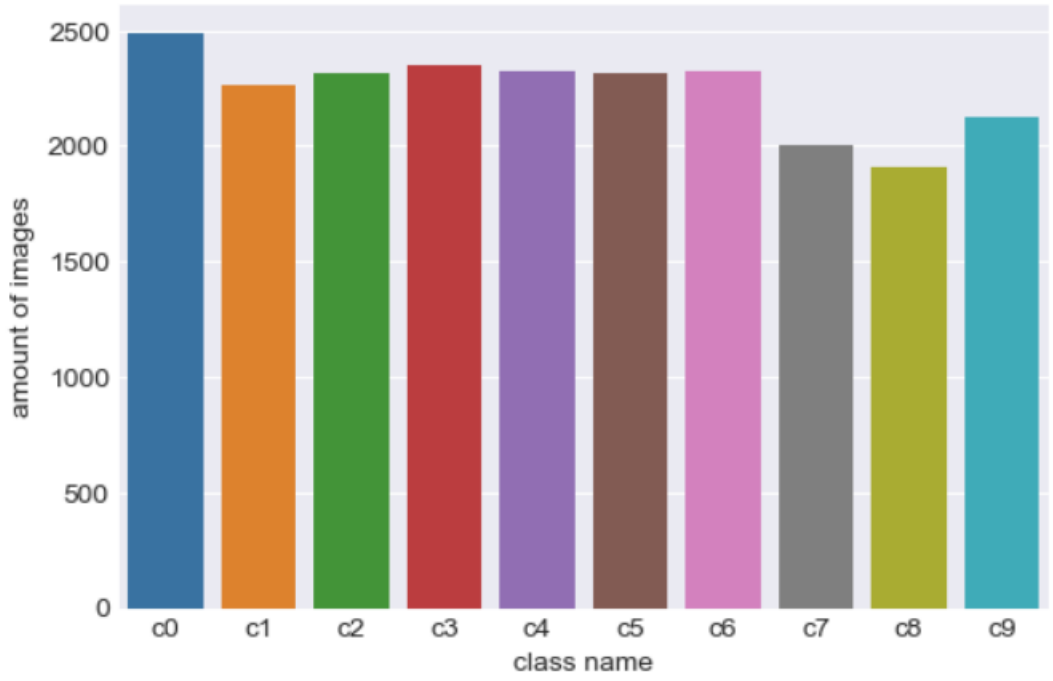


我们可以看到，除了 p072 司机，其他司机图片数量基本处于 600-1200 张这个区间中，大体上还是比较均匀的。

接下来我们看一下每个司机状态类别的分布。

```
Mean: 2242.4
Standard Deviation: 175.39

Text(0.5,0,'class name')
```



我们可以从上图观察到，状态类别的均值为 2242.4 张，分布在 2000-2500 张之间，标准差为 175.39，可以判断，状态类别的分布是比较均匀的。

2.3 算法和技术

项目的目标：将输入的图片进行多分类。

基于这个目标，我将使用卷积神经网络。

2.3.1 卷积神经网络

完成这个项目，我使用卷积神经网络（CNN），它是近年来流行起来，具有高效识别能力的一种方法。CNN 是神经网络中的一种，它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。由于该网络避免了对图像的复杂前期预处理，可以直接输入原始图像，因而得到了更为广泛的应用。

2.3.2 损失函数的选择

我用 keras 搭建模型，选择的损失函数是 `categorical_crossentropy`,针对多分类问题，对应于的损失函数是 `multi-class logarithmic loss`，公式是：

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

2.3.3 优化器的选择

在优化器的选择上，我使用了 adam 算法以及随机梯度下降法（SGD）这两种方法。Adam 算法是适应性学习率算法，一般地来说，适应性学习率算法的基本思想是如果损失函数对于某个给定模型参数的偏导保持相同的符号，那么学习率会增加。这样的操作在一定程度上加速神经网络收敛的速度。这个方法不仅存储了 AdaDelta 先前平方梯度的指数衰减平均值，而且保持了先前梯度 M(t)的指数衰减平均值，这一点与动量类似：

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

其中 Mt 为梯度的第一时刻平均值，Vt 为梯度的第二时刻非中心方差值。参数更新的最终公式为：

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t$$

Adam 算法可以使深层网络模型快速收敛，训练更加复杂的深层神经网络，但是 Adam 算法有非收敛性，为了达到更好的结果，我还将用 SGD 算法继续优化。SGD 算法收敛相对于 Adam 算法更加慢，结合两种算法进行优化。

2.3.4 迁移学习的选择

迁移学习通俗来讲，是把已学习训练好的模型参数迁移到新的模型来帮助新的模型进行训练。大部分数据或者任务是存在相关性的，比如浅层次的卷积层提取的是点和线，深层次的卷积层提取更加复杂的特征。也就是说浅层次的卷积层的任务是类似的。所以我们可以将已经学到的模型参数分享到新的模型，从而加快并优化模型的学习效率，不用从零开始学习。在本项目中，我将使用在 ImageNet 数据集中预训练的模型做迁移学习，这些模型大部分卷积层都已经将特征提取出，我只需要训练一些深层次的卷积层。这样的处理方式对训练集很小的项目有很大的改善。

2.3.5 模型融合策略

在这个项目中，我训练了多个模型，原因是项目的训练数据比较少，容易出现过拟合的情况，无法很好地提取到关键的特征信息。而不同模型因为涉及的思路不同，模型的结构不同，所以提取的图片特征也不同，我将多个模型相结合，能够有更好的效果，事实上，在实际运用中，多模型融合的方式能在一定情况下提升预测的精度。我选择的策略有两种，一是将多种模型预测的结果求均值。二是用多个模型对训练集提取特征，然后用构建一个简单的神经网络，用这些特征继续训练新的神经网络，完成模型融合的步骤。

接下来我将简单介绍一下我在这个项目中使用的模型。

2.4. 模型的选择和简介

我在这个项目中使用的模型都是在各种实践中表现非常好的模型，如 VGG16，ResNet50，以及 InceptionV3，这些模型的结构和设计在项目实践和研究中都有很好的表现。

2.4.1 VGG16

VGGNet 是牛津大学计算机视觉组（Visual Geometry Group）和 Google DeepMind 公司的研究员一起研发的的深度卷积神经网络。通过反复堆叠 3*3 的小型卷积核和 2*2 的最大池化层，VGGNet 成功地构筑了 16~19 层深的卷积神经网络。其中最常使用的模型是 VGG16 和 VGG19，也就是分别用了 16 层和 19 层深度的卷积神经网络。神经网络的结构如下：

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

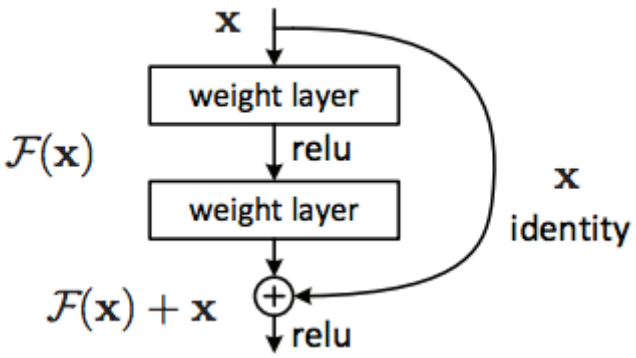
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

图片来源：HTTPS://ARXIV.ORG/PDF/1409.1556.PDF

VGG16 和 VGG19 是图中的 D 和 E。

2.4.2 RESNET50

ResNet（Residual Neural Network）由微软研究院的 Kaiming He 等 4 名华人提出，成功训练 152 层深的神经网络，在 ILSVRC 2015 比赛中获得了冠军，取得 3.57%的 top-5 错误率，同时参数量却比 VGGNet 低，效果非常突出。ResNet 的结构可以极快地加速超深的神经网络训练，而且利用残差学习单元，可以避免深层次的网络梯度下降慢的情况，模型的准确率也有很大的提升。



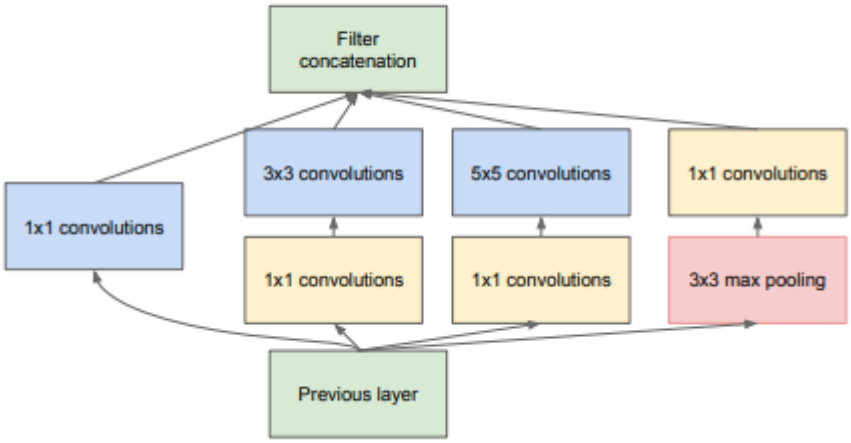
RESIDUAL LEARNING: A BUILDING BLOCK

来源：HTTPS://ARXIV.ORG/PDF/1512.03385.PDF

假定某段神经网络的输入是 x，期望输出是 H(x)，如果我们直接把输入 x 传到输出作为初始结果，那么此时我们需要学习的目标就是 F(x)=H(x)-x，即残差。传统的卷积层在传递信息的时候可能会有信息丢失损耗的问题，这个模型通过将输入绕道到更深的层次，保护信息的完整性。

2.4.3 INCEPTIONV3

在 GoogLeNet 出来之前，模型上主要的图片是加深网络，加宽网络。这样的操作很造成过拟合和计算量大幅增加的问题。Inception 解决这两个问题的方案是增加网络深度和宽度的同时减少参数。首先将卷积分块，将卷积核分组。



INCEPTION MODULE

图片来源：HTTPS://ARXIV.ORG/PDF/1409.4842.PDF

这种设计增加了特征表达的能力，同时减少了计算量。而在模型中，使用了 BatchNorm 和 Gradient Clipping 使训练更加稳定。

2.5 项目优化方案

在这个项目中，我除了使用迁移学习几个成熟的模型，还运用了几个优化的方案，具体如下：

2.5.1 使用 BATCHNORM 层

BN 层通过 mini-batch 来对相应的 activation 做规范化操作，使得结果（输出信号各个维度）的均值为 0，方差为 1。在反向传播的过程中，如果大多数权重小于 1，反向传播的梯度会变得很小很小而无法继续训练。同样，如果大多数权重都大于 1，又会出现梯度爆炸的问题。而 **BN 解决了反向传播过程中的梯度问题（梯度消失和爆炸），因为经过规范化的处理，所有的权重都在一个范围内，不会因为神经网络层数的增加，使得梯度累积到过大或者过小。**

2.5.2 使用数据增强

本项目中，训练集样本数量不多，用数据增强的方式（旋转、平移等方法）增加样本数量，可以在一定程度上增加模型的预测能力。

2.6 项目实现思路

这个项目的核心是一个图片分类的问题，训练出的模型有这样的功能：输入一张图片，返回这张图片属于每一个类别的概率。主要的步骤如下：

首先，分割训练集，划分为训练集和验证集。在项目前期的实验中，发现如果打乱训练集，随机挑选验证集，按照这样的方式分割，会导致过拟合的现象，验证集的评分会非常高，而在测试集的评分会很低，主要的原因是项目提供的训练集的分布，它是有少量的司机的大量图片组成的，而每个司机的每个类别的图片内容中的特征也是相似的，如果用打乱随机抽取的方式划分验证集，那么训练集和验证集中很容易出现内容相似的图片，从而导致严重的过拟合现象，所以解决的方案是抽取两个司机，用他们的图片作为验证集，剩下的作为训练集。同时，用数据增强的方式增加数据的多样性。

然后，用几个优秀的模型做迁移学习，对单个模型不断优化，调节各种参数，加强模型的预测能力。

最后，用训练好的多个模型进行预测，最后将预测结果取均值。

2.7 基准模型

这个项目是一个已经结束的 kaggle 竞赛，在评估衡量解决方案性能时，我观察了这个竞赛的排行榜，最终我选定的阈值为 MutiClass Loss 为 0.3，这个成绩在排行榜上在 193/1440 的位置，大约为 13% 的成绩。

3 实现

3.1 在训练数据集中划分训练集和验证集

首先，我先随机挑选两位司机，然后将这两位司机的所有图片做为验证集，其他图片作为训练集。具体的原因我在 2.6 环节已经说明，这里就不具体讲解。这样方式分割数据集，训练集和验证集的占比大致为 13:1，一般来说，训练集和验证集的占比大致为 8:2，但是考虑到这个项目中，训练数据集比较少，同时又有一个很大的测试集，所以适当缩小验证集的大小。

3.2 数据增强

为了降低过拟合的风险，我选择了用数据增强的方式增加样本的多样性，在 keras 框架中，我使用 ImageDataGenerator 来实现这样的功能，我使用到的增强方法主要有：

- rotation_range 随机旋转图像一定的角度
- width_shift_range 在图像平面上对图像进行水平平移
- height_shift_range 在图像平面上对图像进行垂直平移
- shear_range 裁剪一部分图像
- zoom_range 按照一定比例放大或者缩小图像

通过这样的处理方式防止过你和的问题，在实践这个项目的过程中，这样的处理方式有一定的效果。

3.3 训练模型

在这个项目中我主要使用了三种模型做迁移学习，VGG16，ResNet50 和 InceptionV3。

在训练模型的过程中，我锁定大部分卷积层，不训练被锁定的层数，这样的操作能够加速收敛，并且一定程度上防止过拟合。

主要的参数如下：

模型名称	**锁定层数**	**输入图像大小**	**图片预处理**
VGG16	15	224*224	减去 imageNet 均值
ResNet50	160	224*224	减去 imageNet 均值
Inception	200	299*299	输入范围为（-1,1）

需要提的是，输入图像的大小等于原模型的输入大小，也是推荐的参数。对于 VGG16，ResNet50 模型用 keras 做迁移学习的话，图片预处理的代码部分可以省略。而锁定的层数是经过实验，挑选出最佳的参数搭配。在设定参数之后，对每个模型先用 Adam 算法训练 5 代，然后用 SGD 随机梯度下降法进行细微的 10 代训练（5 代学习率为 1e-4，5 代学习率为 1e-5）。

3.4 模型预测

用训练好的模型对测试集做出预测，然后提交到 Kaggle 上进行评分：

模型名称	**锁定层数**	**训练集选择**	**训练方式**	**Logloss**
VGG16	15	除 p75 和 p81	10 代 Adam	0.434
ResNet50	160	除 p75 和 p81	10 代 Adam	0.438
InceptionV3	200	除 p75 和 p81	10 代 Adam	0.525
VGG16	15	除 p02 和 p47	5 代 Adam，10 代 SGD	0.408
InceptionV3	200	除 p02 和 p47	5 代 Adam，10 代 SGD	0.427

以上是训练出来的比较优秀的模型，但是我们可以看到这些单个模型训练的成绩并不是很理想，然后我将做进一步的处理

3.5 用平均预测值的方法

因为这个项目的训练数据比较少，为了防止过拟合，训练的代数也比较少，在这种情况下，各个模型提取出来的特征是不同的，我将这些模型进行融合，用最简单的平均结果得出最终的结果。在进行这个项目是，我将上面所描述的五个模型融合，最终得出的结果如下：

merge_five_subm.csv 6 days ago by MENGJIE add submission details	0.25775	0.28486	☐

4 结果

4.1 模型的评价和验证

我使用的各个模型的评分已经在单元 3.4 中详细列出，最终的模型是将五个优秀的模型做预测结果均值处理，选定融合后的模型作为最佳模型，原因就是最直观的 kaggle 评分，测试数据共有近 80000 个样本，得出的评分是有一定的参考价值的。最终的评分为 0.28486，大致 kaggle 竞赛的 12%，相对来说是比较好的。

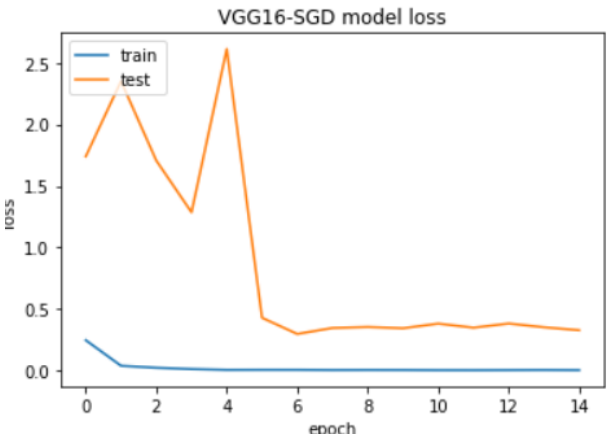
4.2 合理性分析

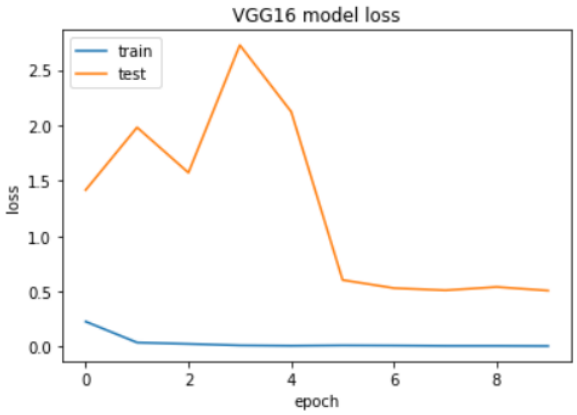
我最后得到的结果 Private Score 为 0.25775，Public Score 为 0.28486，我选定基准模型的阈值为 0.30，最终的模型比基准模型更为出色。对于司机状态分类这部分问题已经有了很好的成效，对于实际解决侦测司机状态的问题，还需要进行还需要将模型部署到检测设备上，这里不做更深入的研究。

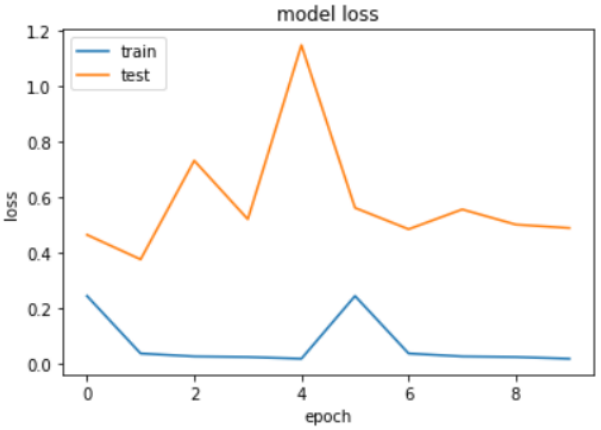
5 项目结论

5.1 结果可视化

我最后的结果是融合了五个优秀的模型，将其预测结果累加并计算均值，得出最后的结果。在结果可视化环节，我们来看一下他们在训练集和验证集上 loss 的变换情况。

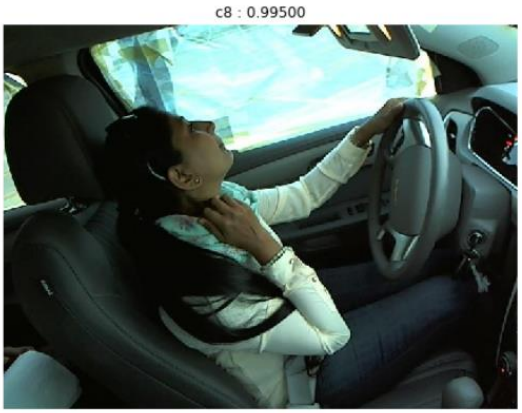




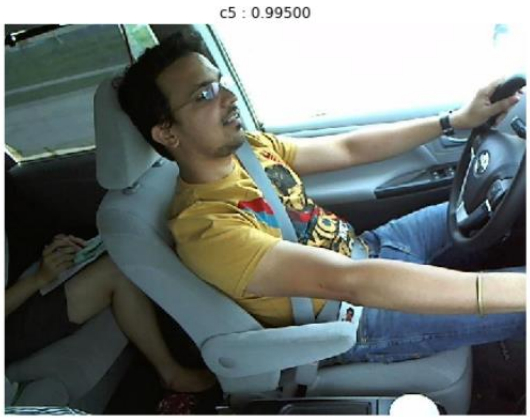


我选取了三个模型的可视化，我们可以看到，这些模型在五代之后 loss 有了明显地下降，这是因为我调整了学习率，用了 SGD 算法配合更低的学习率去训练模型，这种方式也是我在各种实验下的最终方案，而为了防止过拟合，我看到 10 代左右的，loss 趋于稳定，所以模型的训练次数选择为 10 代或者 15 代，具体的模型评分已经在单元 3.4 中列出。

最后我们来随机抽取几个测试集图片以及模型预测的结果。



左图 c8 为整理头发和化妆（正确），右图 c3 为左手打字（正确）



左图 c5 为调节收音机（正确），右图 c6 为喝饮料（正确）

5.2 总结

本项目使用了 keras 框架搭建模型，相对于其他的框架部署更加快捷。因为训练数据有限，并且模型设计和参数调整方面经验不足，所以选择了成熟的模型做迁移学习，同时也能加速训练。在数据集验证集的选择上，针对了这个项目所以提供的数据，做了个性化的选择，而不是传统的方式，用打乱数据集随机挑选的方式进行划分。在模型的训练过程中，尝试了很多种方式去防止过拟合，主要的方式有数据增强和增加 BN 层，同时也测试了 Dropout 层，最终选定了 BN 层。最后为了进一步的提升评分，用了预测结果取均值的方式做迁移学习，得到了很大的提升。总体来说，通过此项目，经历了完整的图像处理领域的项目，无论从项目的总体进程，如何解决具体问题等方面有了更多的经验，对于项目的结果，比较满意，接下来提出几个改进的方向。

5.3 需要做出的改进

5.3.1 用滑动平均算法的方式进一步优化模型

我们可以在数据集观测到，数据集大部分的照片是前后几帧的照片，这些图片是相似的，状态也基本相同，我们找到某个照片邻近的几张图片，然后将原本图片和邻近图片的预测结果按一定权重累加，原照片权重最大，然后越邻近的图片权重越大，得出最后的结果，可以一定程度上提升预测的精度。

而在实际项目中（用摄像头监控司机的状态），也可以运用类似的方法，连续预测几帧图片，如果预测结果均为危险驾驶状态，则发出警报，这样能一定程度上减少错误警报的次数，但是选取的帧数一定要小，不然会失去时效性。

引用

- VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION Karen Simonyan, Andrew Zisserman (Submitted on 4 Sep 2014 (v1), last revised 10 Apr 2015 (this version, v6))
- Deep Residual Learning for Image Recognition Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (Submitted on 10 Dec 2015)
- Going Deeper with Convolutions Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich (Submitted on 17 Sep 2014)
- Keras-5 基于 ImageDataGenerator 的 Data Augmentation 实现
- Classification of Driver Distraction Samuel Colbran, Kaiqi Cen, Danni Luo
- Kaggle. A brief summary jacobkie
- 手把手教你如何在 Kaggle 猫狗大战冲到 Top2% 杨培文 (发表于 2017-03-18)