**机器学习纳米学位**

##毕业项目 黄煜祺 [qier\_241078450@qq.com](mailto:qier_241078450@qq.com) 2017年10月12日

**I. 问题的定义**

**项目概述**

随着时代的发展，随之而来的交通事故发生率也随之上升，而其中很大一部分是由驾驶者的不正当驾驶行为所导致，而无人驾驶技术的出现，是可以起到由人为导致的交通事故率的，其作为一项跨学科的存在，其中作为行车安全的因素，路面探测和动作选择占了十分重要的地位。而当车辆的前置摄像头把车辆行驶中的前方图像传回控制终端，就需要控制终端作出合适的判断，控制方向盘的转动方向，以保证行车安全，本项目是通过Tesla的行驶数据（来源：<https://github.com/nd009/capstone/tree/master/deep_tesla>），对其方向进行模拟和预测，在研究的同时本人将保证不在商业行为上使用这些数据集。在数据集中，分为两种数据：

1. 车辆前置相机所拍摄的路况图像视频（xxx\_front.MKV），示例为：



1. 车辆前置相机所拍摄的路况图像视频所得到的方向盘转向数据文件（XXX\_steering.csv），其中包含3列数据，示例数据为：

|  |  |  |
| --- | --- | --- |
| **ts\_micro** | **frame\_index** | **wheel** |
| 1464305394391807 | 0 | -0.5 |
| 1464305394425141 | 1 | -0.5 |
| 1464305394458474 | 2 | -0.5 |

ts\_micro是时间戳，frame\_index是帧编号，wheel是转向角度（以水平方向为基准，+为顺时针，-为逆时针

**问题陈述**

我计划，根据公开的Tesla数据集对车辆的前置相机所拍摄的路况图像，实现对车辆转向角度的预测，以使得车辆能在特定情况下做出正确的方向选择，最终提高行车安全。

**评价指标**

我计划使用以下参数值对模型进行评估：

1） 向前/向后反应耗时值：forward/back pass ms，通过从每次车辆遇到路况到实际作出动作的时间间隔计算得出

2） 平均误差值：mean absolute error，实际和预测间的差值，据对数据集的观察，值存在正和负，因此此处将取绝对值以防止数据被抵消，根据车辆实际作出的方向盘转向值与数据集中对应帧的转向值的误差值取绝对值的总平均值算出

**II. 分析**

**数据的探索和可视化**

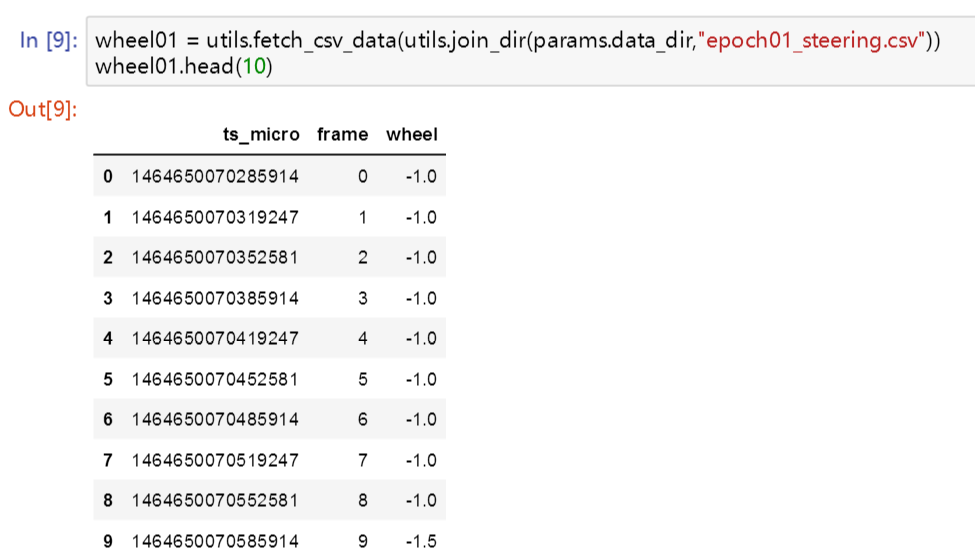
整个数据集中，共有10个前置摄像视频文件及10个转向数据文件，在开始研究问题前，先使用代码对数据集进行探索。

由于10个视频文件以mkv格式提供，而视频文件是通过具体的帧数对应转向数据文件，因此将使用OpenCV完成，具体代码及运行效果为：



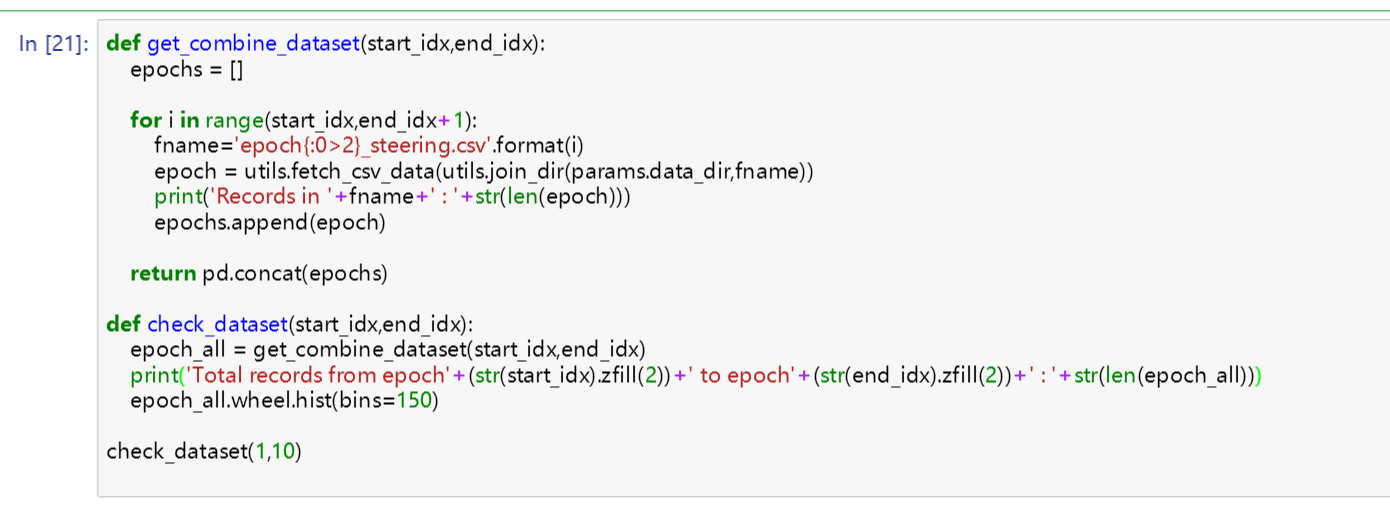
可见，每帧文件大小为1280X720，颜色深度为3。

而对应的转向数据文件，使用pandas检查其数据格式，代码和效果如下图：



可见转向数据记录是按帧顺序记录的。

而由于共有十个,经过合并数据集后，10个steering数据集的wheel的值分布的代码如下图：



运行效果如下图：



可见每个steering数据集有大概1500~3900行数据，总共有27000行数据，是足够训练一个普通的CNN模型的。但是，很显然由于多数帧的转向数据是直行决策远多于转弯决策，将有可能造成训练出来的模型更偏向直行，因此如何避免这种情况出现也是本项目的研究方向之一

**算法和技术**

在本项目中，视频文件的每一帧作为数据的输入，而每一帧对应的方向盘转向作为对应的决策，同时，由于平均一段视频中有2000多帧，每一帧为1280X720X3的图片，且最后的决策为角度转向，而由于卷积神经网络（下称CNN）本身避免了对图像的复杂前期预处理，可以直接输入原始图像，最后输出特征平面，因此将使用CNN作为训练该项目视频数据的模型。而帧作为模型的输入，将通过CNN的处理后，将输出帧的特征点，从而预测出方向盘转向。

**基准模型**

在Nvidia的无人驾驶的一篇论文（[论文正文](https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf)）中，提出了一个无人驾驶可用的CNN基准模型，结构如下图：



此模型共有一个输入层，一个标准化处理层，五个卷积层，三个全连接层，一个输出，本项目最终的模型结构将与该模型进行比较。

**III. 方法**

**数据预处理**

在上一部分中，已对数据有了初步的了解，而在除去第10个视频文件和数据文件后，剩下大约25000帧图片及对应的转向数据可供模型训练，且也已知大部分转向决策为直行，显然如果模型按照这些训练集训练出来后，不足以应对现实世界中千变万化的行驶状况，即训练集的数据过少，将造成模型的训练不够充分。因此，将使用数据增强技术（本文采用Regularization相关的方法），改变已知帧数据进行颜色变换，平移变换，缩放变换等处理，以增强训练效果。

由于帧有25000多个，下面我将在使用数据集中的示例图片（路径：.\deep\_tesla\images\img\frame\_1173.jpg）表述处理流程。首先是读取这张图片及其对应的转向数据，代码及效果如下图：

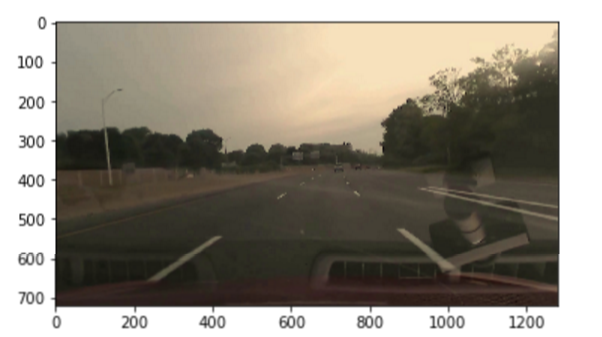
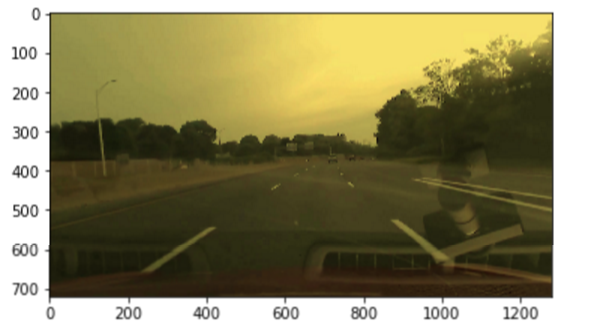


**处理一：颜色变换**

更改图片颜色色调，可模拟同模型在同一路况下不同的亮度所作出的不同的转向处理。可使用OpenCV中的cvtColor改变图片色域以及随机改变图片第三颜色通道的值，以达到效果，代码如下图：

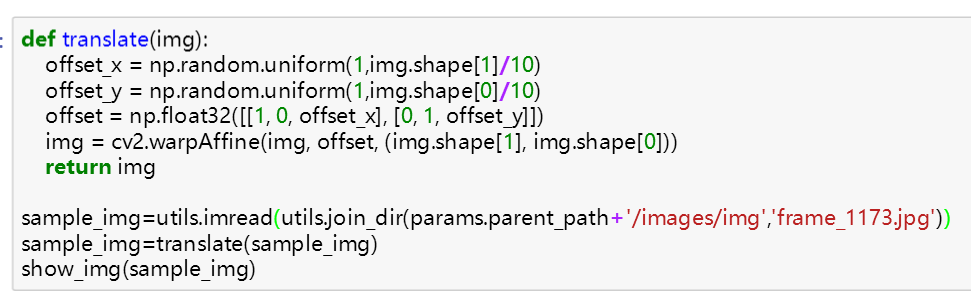


从而可获取经过随机改变第三颜色通道值后的图像：



**处理二：平移变换**

通过平移变换，可使得同一图片的前方路况被改变，可使得模型在同一环境下可学习到更多的决策。可使用OpenCV中的warpAffine并设置随机偏移量实现，代码如下图：

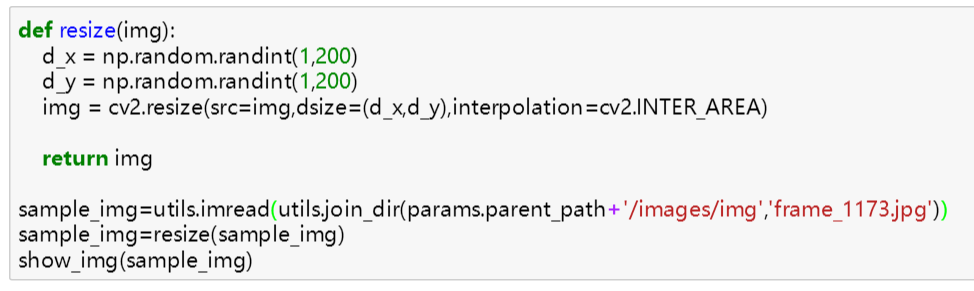


从而获得经过随机平移的图像：



**处理三：缩放变换**

除了平移变换，对图片进行缩放变换也可达到模型在同一环境下，不同路况下学习不同的决策，可使用OpenCV中的resize并设置随机压缩大小实现，代码如下图：

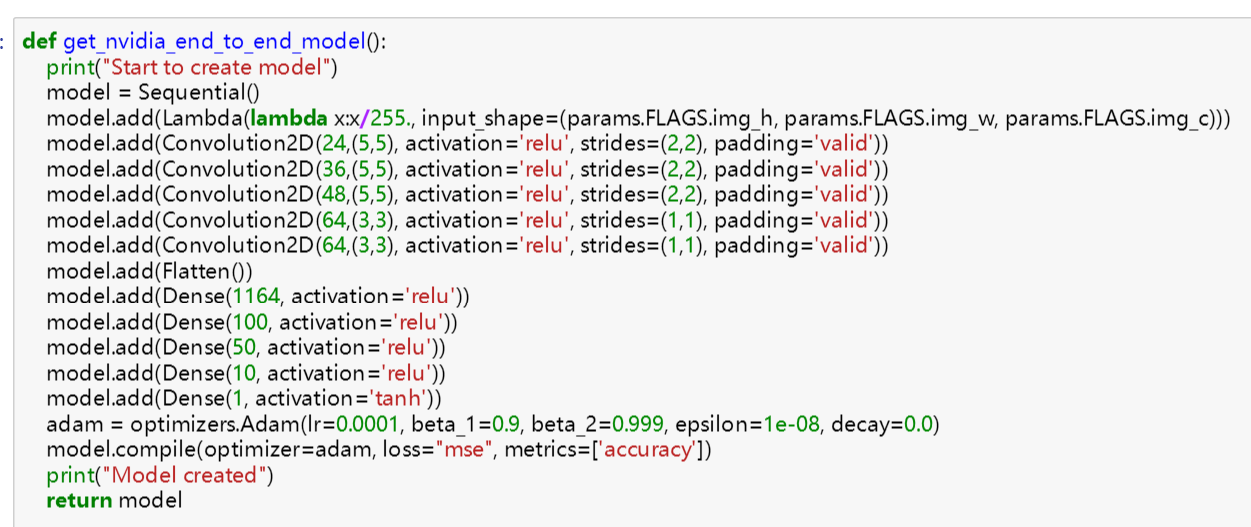


从而获得经过随机压缩的图像：



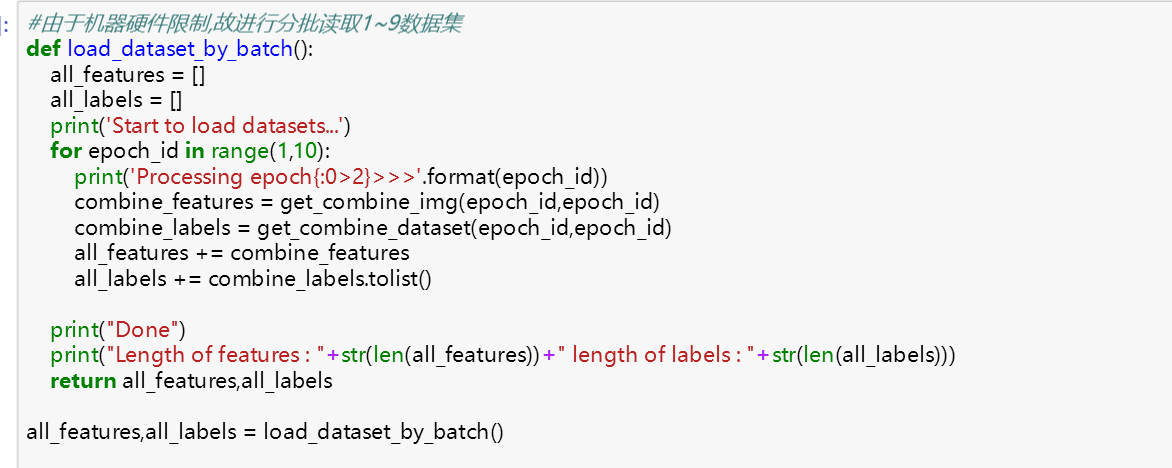
**模型设计**

开始执行前，首先需设计合适的模型，在这里将选择基于Nvidia的end-to-end基准模型，以keras作为较为清晰易懂的CNN结构包作为主要工具以展示。结构中，输入将先通过标准化层，降低其离散区间（每个维度除以255，以keras.layers.Lambda实现），以加速在不同GPU上的处理时间；然后，将通过三个5X5的kernel和2x2的stride，两个3X3的kernel和无stride的卷积层（以keras.layers.Convolution2D）；接着经过flatten（以keras.layers.core. Flatten实现）后，将通过3个全链接层（以keras.layers.Dense实现），将输出的提前特征，用于训练其转向预测，具体代码如下：

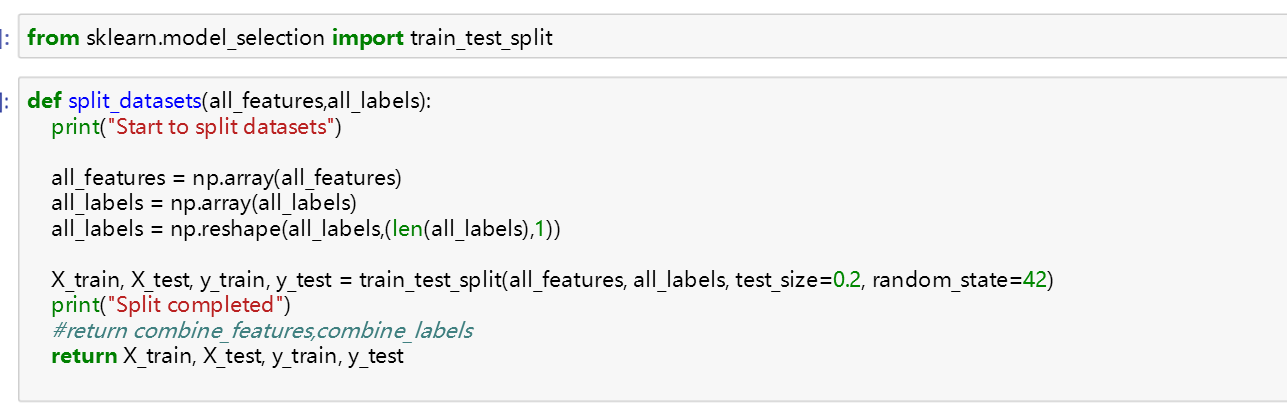


**执行过程**

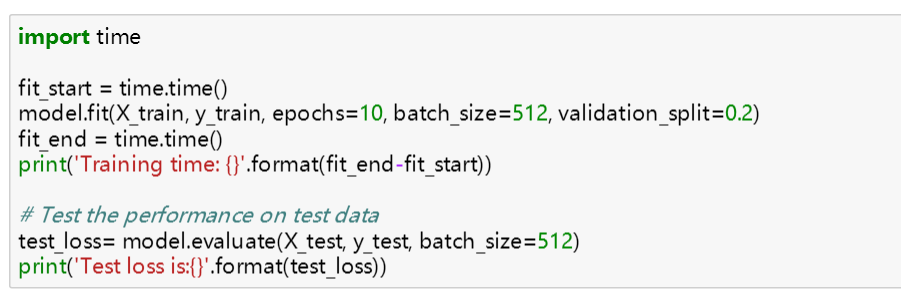
完成模型建立后，进行预测前，首先需对模型进行训练，此处将使用编号1~9的视频和转向文件合并作为训练集和测试集，生成代码为：



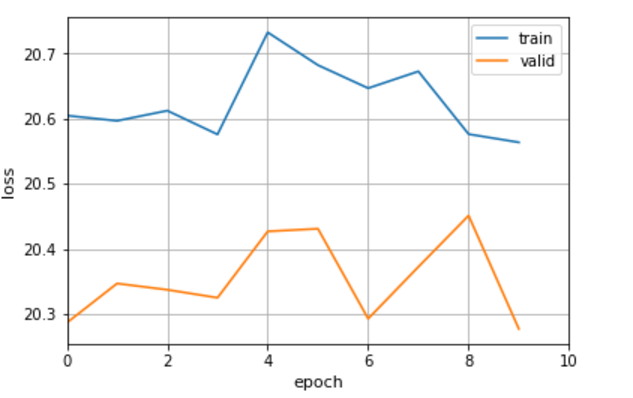
然后便分割训练集和测试集，比例为8：2，代码为：



接着便可进行模型预测，代码为：



其运行结果为：



结果中，训练集的loss徘徊在20.6~20.7，测试集的loss徘徊在20.3~20.4之间，显然此模型的loss过高

**完善**

在上一步骤中，可见loss过高，故修改模型为：

在这一部分，你需要描述你对原有的算法和技术完善的过程。例如调整模型的参数以达到更好的结果的过程应该有所记录。你需要记录最初和最终的模型，以及过程中有代表性意义的结果。你需要考虑的问题：

* *初始结果是否清晰记录了？*
* *完善的过程是否清晰记录了，其中使用了什么技术？*
* *完善过程中的结果以及最终结果是否清晰记录了？*

**IV. 结果**

*（大概 2-3 页）*

**模型的评价与验证**

在这一部分，你需要对你得出的最终模型的各种技术质量进行详尽的评价。最终模型是怎么得出来的，为什么它会被选为最佳需要清晰地描述。你也需要对模型和结果可靠性作出验证分析，譬如对输入数据或环境的一些操控是否会对结果产生影响（敏感性分析sensitivity analysis）。一些需要考虑的问题：

* *最终的模型是否合理，跟期待的结果是否一致？最后的各种参数是否合理？*
* *模型是否对于这个问题是否足够稳健可靠？训练数据或输入的一些微小的改变是否会极大影响结果？（鲁棒性）*
* *这个模型得出的结果是否可信？*

**合理性分析**

在这个部分，你需要利用一些统计分析，把你的最终模型得到的结果与你的前面设定的基准模型进行对比。你也分析你的最终模型和结果是否确确实实解决了你在这个项目里设定的问题。你需要考虑：

* *最终结果对比你的基准模型表现得更好还是有所逊色？*
* *你是否详尽地分析和讨论了最终结果？*
* *最终结果是不是确确实实解决了问题？*

**V. 项目结论**

*（大概 1-2 页）*

**结果可视化**

在这一部分，你需要用可视化的方式展示项目中需要强调的重要技术特性。至于什么形式，你可以自由把握，但需要表达出一个关于这个项目重要的结论和特点，并对此作出讨论。一些需要考虑的：

* *你是否对一个与问题，数据集，输入数据，或结果相关的，重要的技术特性进行了可视化？*
* *可视化结果是否详尽的分析讨论了？*
* *绘图的坐标轴，标题，基准面是不是清晰定义了？*

**对项目的思考**

在这一部分，你需要从头到尾总结一下整个问题的解决方案，讨论其中你认为有趣或困难的地方。从整体来反思一下整个项目，确保自己对整个流程是明确掌握的。需要考虑：

* *你是否详尽总结了项目的整个流程？*
* *项目里有哪些比较有意思的地方？*
* *项目里有哪些比较困难的地方？*
* *最终模型和结果是否符合你对这个问题的期望？它可以在通用的场景下解决这些类型的问题吗？*

**需要作出的改进**

在这一部分，你需要讨论你可以怎么样去完善你执行流程中的某一方面。例如考虑一下你的操作的方法是否可以进一步推广，泛化，有没有需要作出变更的地方。你并不需要确实作出这些改进，不过你应能够讨论这些改进可能对结果的影响，并与现有结果进行比较。一些需要考虑的问题：

* *是否可以有算法和技术层面的进一步的完善？*
* *是否有一些你了解到，但是你还没能够实践的算法和技术？*
* *如果将你最终模型作为新的基准，你认为还能有更好的解决方案吗？*