**作业2**

1. 不断拍照保存太麻烦，导致训练数据集太少，效率较低

解决方法：先收集适当的原始数据，再对原始收集到的数据进行仿射变换来达到数据增广的目的

1. 采集标志牌数据的时候，由于边界裁切导致标志牌不完整

解决方法：通过加入padding量，在车牌框周围加入一些多余小量，扩大框选范围

1. 虽然卷积层数和迭代次数都比较小，但是模型训练速度还是稍微有点慢

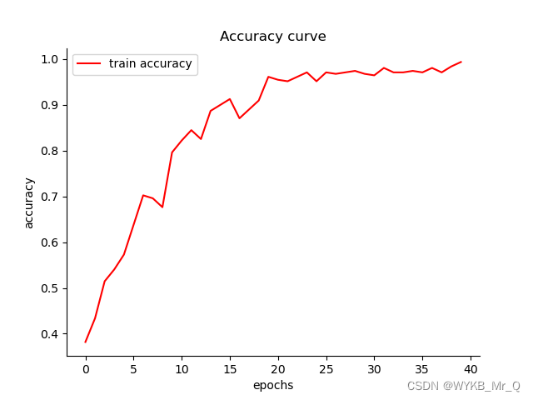
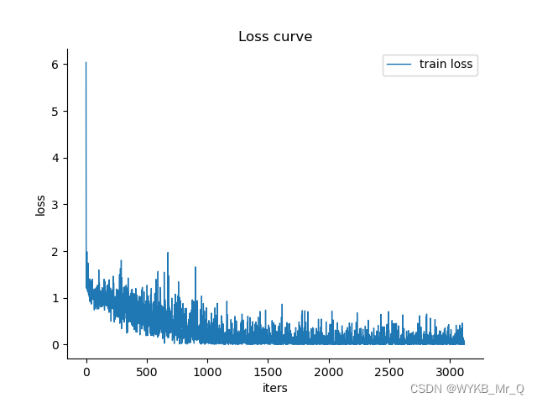
解决方法：鉴于本人电脑似乎没有GPU，故在引入multiprocessing库，并在主函数中使用multiprocessing.freeze\_support() 使用多线程，可以加快训练速度

1. 模型对于训练集识别正确率可达99%以上，但对测试集识别正确率只能达到84%左右，出现“过拟合”现象

解决方法：原先只对四种指示牌各读入100张照片，后改为读入所有照片，每种指示牌约500-600张；将迭代次数num\_epochs由20次改为8次

1. 训练结果不够清晰

解决方法：利用matplotlib可视化Loss和Accuracy随epochs变化曲线



1. 模型在电脑本地测试集上正确率很高，但是在树莓派上的效果出现异常的差，甚至完全识别不出来，出现置信率为NaN(Not a Number)的情况

解决方法：树莓派上的系统是32位的，默认的数据精度是float单精度浮点数，而在本地的训练环境是64位的系统，默认的数据精度都是double双精度浮点数，因此直接在树莓派上进行推理会存在数据精度的差异，导致推理效果不一致。

可以通过 .double() 方法将网络精度和网络的输入都设置成double类型

1. 小车摄像头中会识别多个蓝色区域，并且都能识别出一个结果

解决方法：为标志牌识别区域设置阈值，规定只识别大于特定大小的蓝色区域，并且只保留最大的那块蓝色区域

1. 识别到转弯标志牌后，转弯过程中并不是直对着转弯指示牌，识别有问题

解决方法：转弯过程中通过sleep()函数使小车暂停识别，并充分转弯

1. 小车有时候会将park识别成right，导致停车后可能还会出现移动

解决方法：小车一旦识别到park，立马退出循环，停止一切运动

1. 蓝色区域识别过程中，出现蓝色HSV判断区域不准确、识别到其他类蓝色的区域错

解决方法：调整环境亮度，输出标志牌的蓝色HSV数组，界定实验环境下蓝底HSV范围