中国人民大学

2016 级信息学院图灵实验班

本科生课程报告

报告主题：人工智能课程总结

课程名字：人工智能导论

学院：信息学院

班级：图灵实验班

姓名：江流洋

学号：2016202185

指导老师：胡鹤

2018 年 12 月 25 日星期二

一、引言

为期半年的人工智能课程即将落下帷幕，在学期将要结束的时候，写下了这篇总结，作为一次学习的记录，记录下自己的收获，同时记录下自己的目标，鞭策自己继续学习。

在这个学期，我一共参与的与机器学习有关的项目有三个。第一个是在范举老师的实验室进行的项目 ***Entity Matching with Deep Learning***，这个项目使用深度学习的方法实现实体匹配；第二个是在 SLP（Spoken Language Processing）课程上作为小组长实现的大作业 ***Voice Conversion***，这个项目是用机器学习的方法实现一个 many-to-one 的变声器；第三个就是在这门 AI（Artificial Intelligence）课程上作为小组长实现的智能小车项目 ***AI-Car***。

下面我将会一一的介绍我参与的这些项目。

二、项目

1、*Entity Matching with Deep Learning*

(1)项目介绍

给出两个对于实体的描述，判断这两个实体是不是同一个实体。这一类问题的研究对于构建知识图谱有很大的帮助。

(2)项目工作

在这个项目工作中，我的主要工作是阅读了 deepmatcher 的相关论文，并且阅读了 deepmatcher 的源代码，其源代码是用 pytorch 写的，我的工作是用 tensorflow 来重现这个 pytorch 项目。虽然是有源代码的支持，但是 tensorflow 和 pytorch 的区别还是相当大的，期间遇到的一些问题，也让我对于 tensorflow 的使用和深度学习的方法有了更深入的了解。

Deepmatcher 主要是用 seq2seq 的理念，将两个实体的描述匹配起来，需要先对描述（sequence）做 word_segment，word_embedding，得到序列的向量表示，作为输入传入模型中，计算两个序列的每个描述特征相似度，最后再把所有特征的相似度做权重的加和，作为两个实体的相似度，用训练得到的阈值来判断两个实体是否匹配。

我的下一步工作是想办法改进模型，使得模型在 Abt-buy 数据集上能够得到更好的效果。由于项目还在进展状态，就不再过多的赘述。

2、*Voice Conversion*

项目地址：Voice-Transformer(https://github.com/prime51/Voice-Transformer)

(1)项目介绍

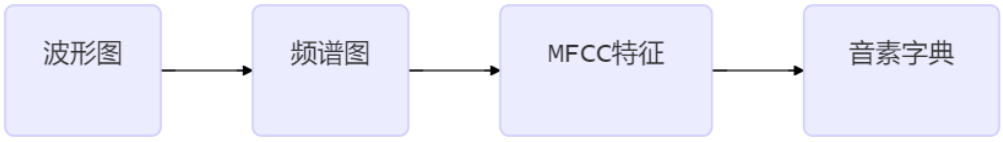
变声器，传统的声音转换，使用的是 parallel data training 的方法，得到的是 one-to-one 的声音转换系统，也就是训练集的来源是源说话人与目标说话人必须说相同的语音内容，然后得到的变声系统只能从源说话人的声音转换到目标人说话的声音，这些传统方法里面以 DBLSTM 的效果最好。而我们小组这次实现的是 Non-parallel-dataset，Many-to-one 的变声系统，不需要使用 parallel dataset，并且能够实现任意声音到训练好的目标声音的转换。

(2)项目实现

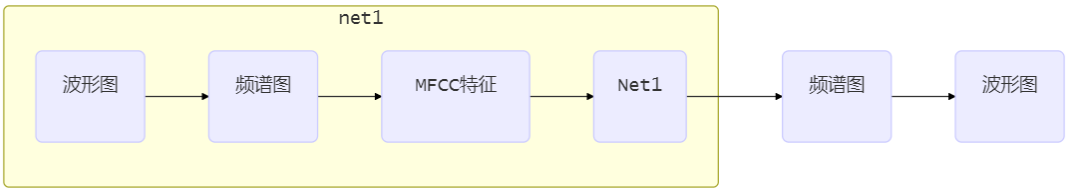
(i)模型结构

模型的结构主要由两个网络构成：

Net1： 分类器，其作用是将语音中的不同频谱分类成不同音素。它的输入是各种语音材料，以及与语音对齐的标签，训练的语音是与说话者相关的，但是训练出来的分类器确实独立的，也就说任何一段语音都能拿来用它分类。**Net1** 的训练集是 TIMIT ，一个有 630 名不同人录的 10 段语音的数据集。它的流程图如下：



Net2： 合成器，它包含一个 **Net1**，将 **Net1** 分类得到的音素，加入源声音的特征合成出新的语音。它的输入是源声音的音频，输出是目标声音的音频，其中需要先用 **Net1** 将源声音音频分类音素，然后学习不同音素的特征，再加入到目标声音的音素中，得到转换后声音的频谱图，最后再将频谱图转换成波形图。它的流程图如下：



(ii)模型训练

模型的训练需要注意以下几点：

a) 为了防止过拟合，最后需要加入 L2 正则化项，提高模型的泛化能力。

b) 需要 GPU 加速，本次项目利用了 Google Colab 提供的 GPU 训练平台。

c) 利用 Tensorboard 实时查看模型的训练情况（如 loss、accuracy 等值的变化情况），以便确定模型什么时候达到最佳效果。

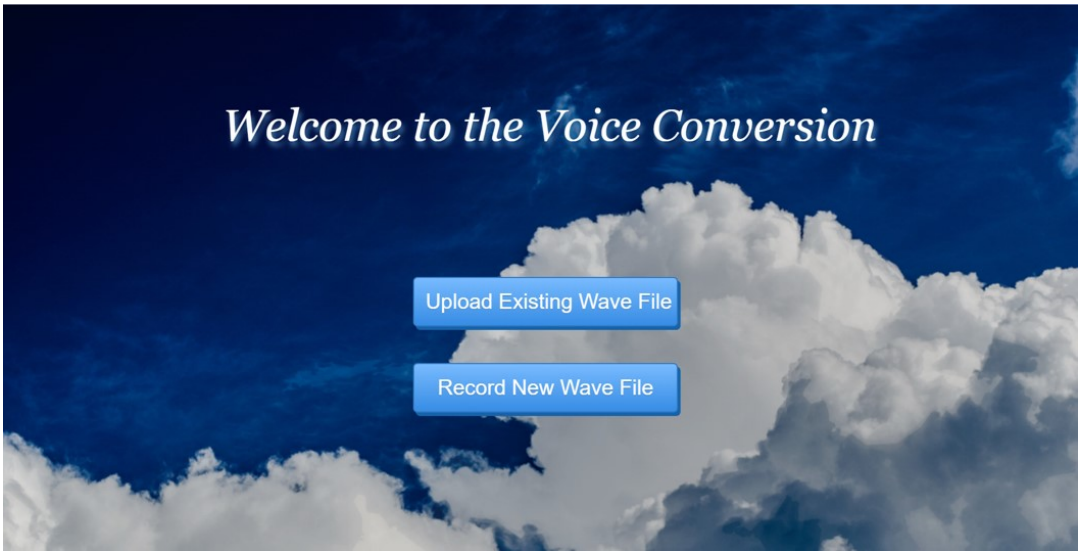
(iii)前端设计

使用的是 **html + CSS(3) + Javascript**。

主要分为三个页面：**Main page**（主页）、**Upload page**（上传本地音频）、**Record page**（录制音频页）。

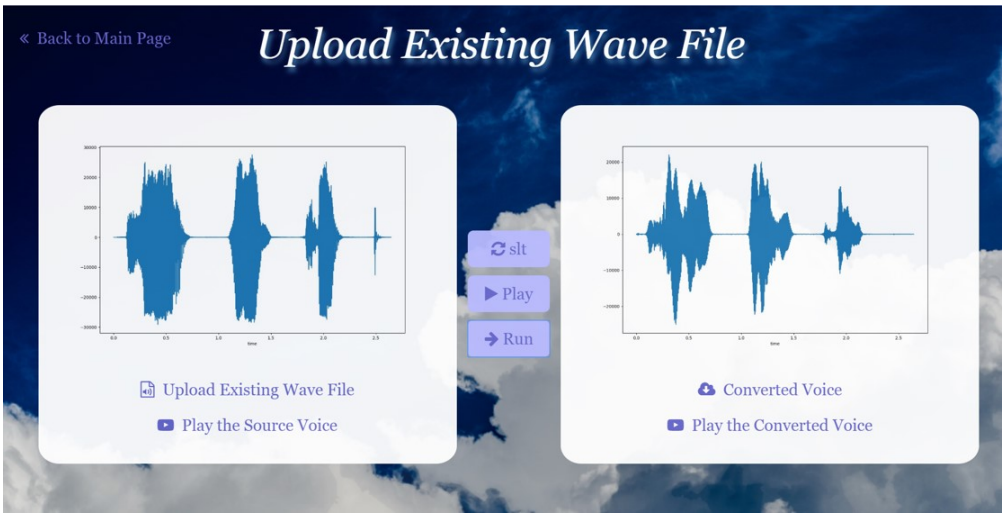
页面设计主要是为了能实现最终的效果，比较转换前的语音，目标语音和转换后的语音。

I. Main page



Main page 有两个按钮，Upload Existing Wave File 和 Record New Wave File，分别对应的功能是跳转到 Upload page 和 Record page。

II. Upload page



Upload page 可以分为三个部分，左边是上传本地的 wav 文件，作为转换前的音频文件，但是因为后端的路径问题没有解决，所以这里只可以上传 /Voice Conversion/static/audio/ 文件夹下面的 wav 文件。这里会把上传的音频的频谱图绘制并显示在页面中，用于查看最后的效果。中间是三个按钮，第一个是切换目标声音的按钮，因为项目时间有限，我们只训练了两个目标人的声音—— slt 和 ksp，所以目标声音只能在两者之间切换。第二个按钮是播放目标声音的一段示例录音。第三个按钮则是开始将源声音转换成目标声音，转换的过程大概需要花费 30s-50s 时间，需要测试人耐心等待。转换成功后的语音的频谱图将会在右边显示出来，可以通过点击 Play the converted voice 按钮来播放转换后的语音，查看最终的效果。

III. Record page

Record page 和 Upload page 页面上大致类似，主要是源声音的来源不同，可以临时录制一段长不超过 10s 的音频，保存在 /Voice Conversion/static/audio/ 文件夹下面，点击 Run 按钮即可开始转换。

(iiii)后端设计

使用 Flask + jQuery + Ajax。

整体框架使用的是 python 的轻量级 web 框架 Flask。另外使用 Ajax 和 jQuery 来实现前后端的交互。

交互过程主要是为了运行绘制频谱图（draw_wave.py）和转换声音（final_converion.py）这两个 python 文件。然后将结果显示在页面上。所以，交互的信息主要是需要绘制频谱图的 wav 文件名，待转换声音的源 wav 文件名，以及转换声音的目标人物的名字。服务端收到来自前端的这些信息，就可以在后端命令行运行相应的 python 指令，保存结果，并返回给前端。由于前后端交互时的文件路径问题没有解决，所以这里的源 wav 文件都只能放置在 /Voice Conversion/static/audio/ 文件夹下面。

由于 Flask 框架的不稳定性，在测试过程中有可能出现服务端崩溃的情况，这个时候需要重新启动服务端，清除缓存刷新页面即可。

(3)项目总结

(i)本次项目达到了将目标声音融合源声音音色特点的实验效果，并且不需要完全对齐的训练集，且最终的转换是多对一的，即只需要学习到源声音的特征，可以将任何人的声音转化为源声音

(ii)在深度神经网络的模型中，GPU 加速尤其重要，而 Google Colab 给我们提供了一个免费试用的平台

(iii)除了课堂学习到的传统提取音频特征的方法，我们借助这个项目学习了如何使用神经网络的模型自动学习音频的特征，而且这些隐藏特征很有可能是传统方法无法发现的。

(iiii)我们的项目还存在一些问题：

a) 输入的目标声音必须保证在一定的时间内，即音频长度是固定的。
b) Flask 框架的不稳定性导致前端平台可能出现崩溃的情况。

3、*AI Car*

项目地址：AI-Car(https://github.com/jaingmengmeng/AI-Car)

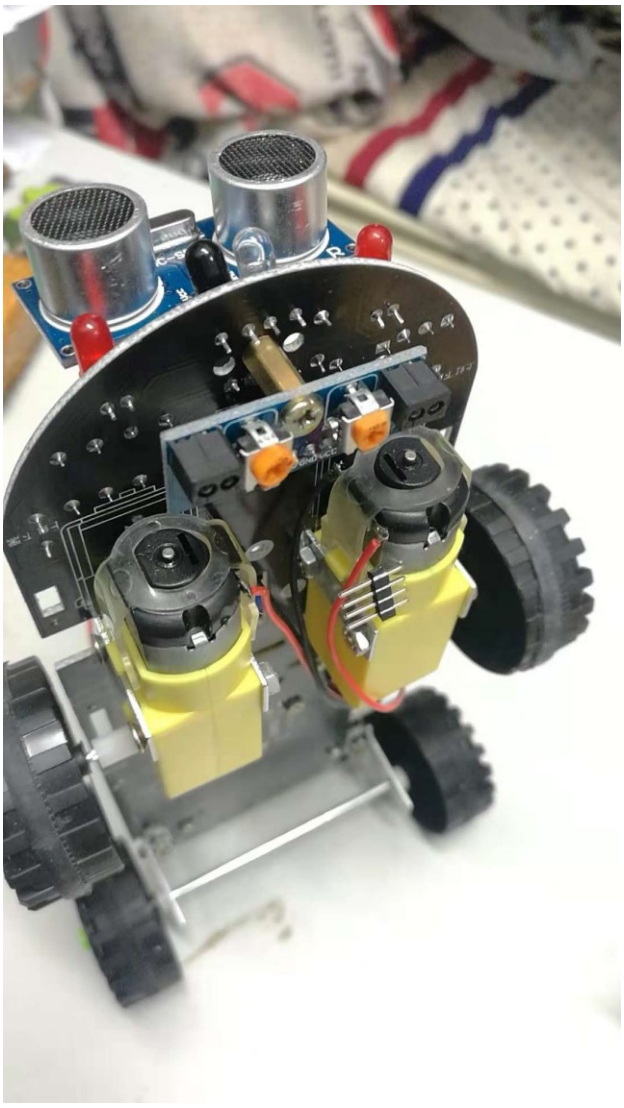
智能小车的项目，也是这门课程的重点项目。该项目一共分为三个小项目来实现。分别是 Random Walker、Sensored Walker、DeepAI Walker。

（1） Random Walker

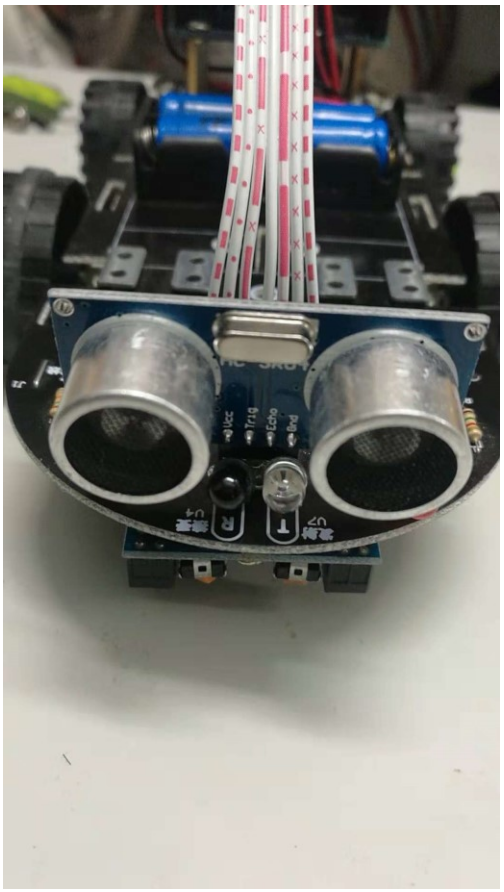
a）购买小车器件
b）熟悉 Arduino IDE
c）烧录程序

（2） Sensored Walker

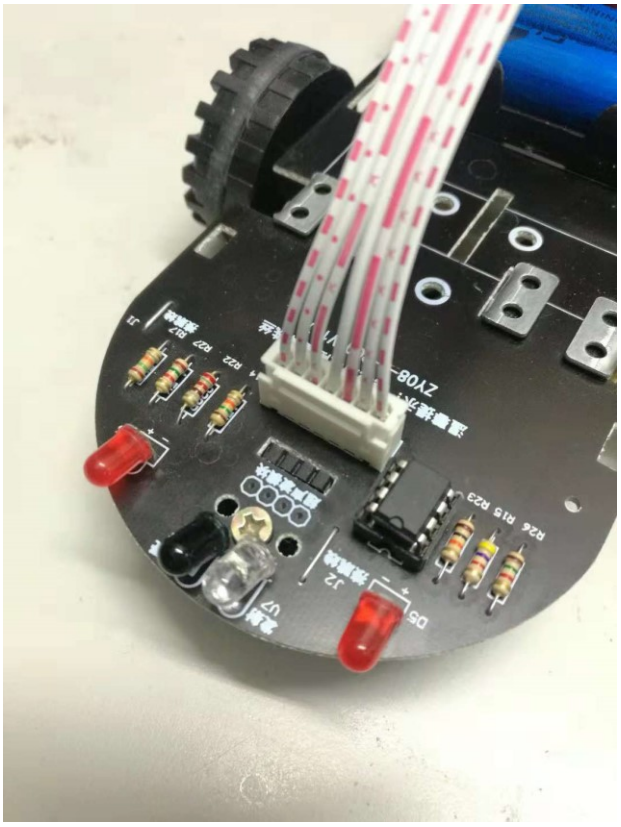
a）学习传感器原理
b）学习使用蓝牙串口 APP（蓝牙 SPP）
c）红外寻迹



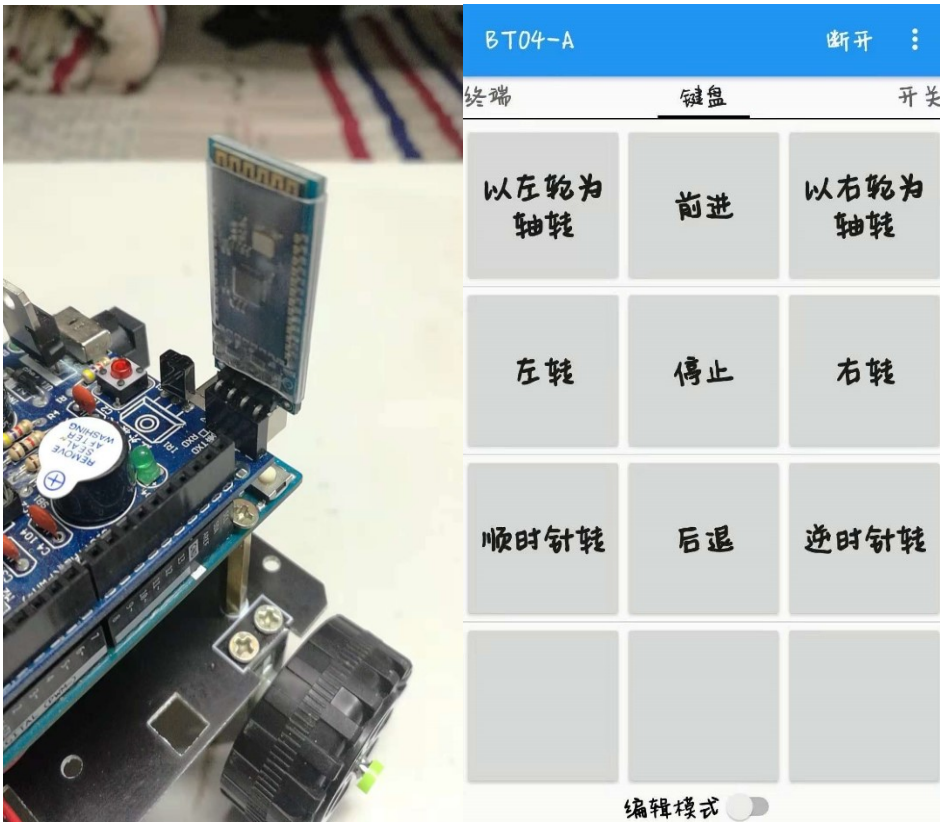
d） 超声波避障



e） 红外避障



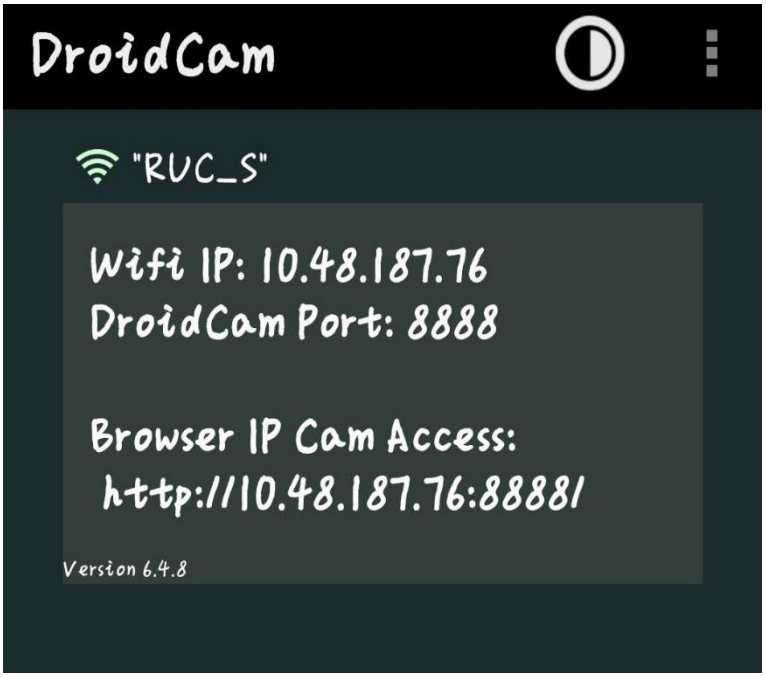
f） 手机蓝牙遥控

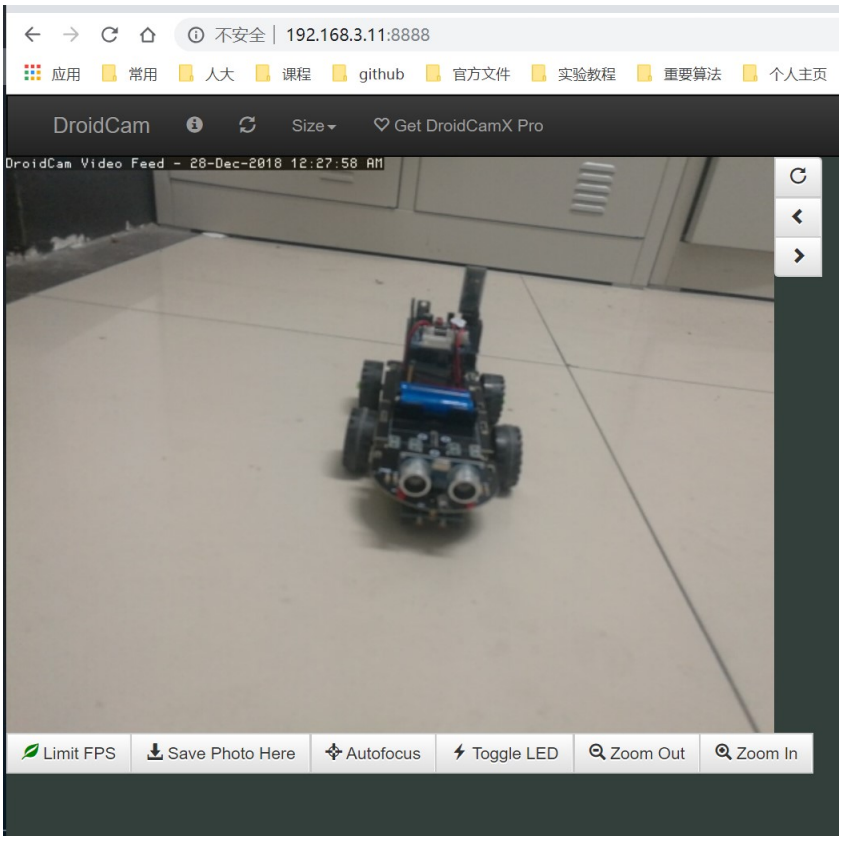


（3） DeepAI Walker

a） IP 摄像头

为了实现将手机放在小车上充当摄像头，并将拍摄到的内容传给 PC，需要下载一个 APP（DriodCam X Pro）。这样能够通过手机开一个服务端，只要和手机在同一个局域网下的 PC 端就能够通过访问特定的 IP 地址和端口号就能够获取手机摄像头拍摄出来的视频。





b) Opencv

通过 Opencv 提供的 API 接口，能够获取 DroidCam X Pro 的服务端视频流，并显示出来。

```
video= cv2.VideoCapture('http://10.48.187.76:8888/mjpegfeed')
```

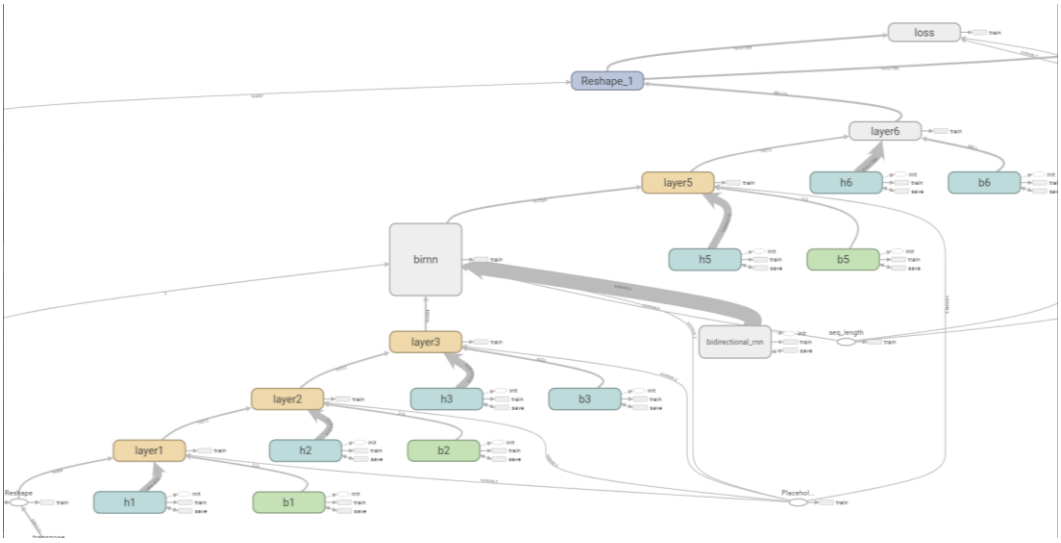
c) Python pySerial

为了让电脑指挥小车的行为，需要通过蓝牙将指令传输给小车。这时我们可以使用 python 的包 pySerial 来监听 PC 端的蓝牙端口，来发送和接收消息。

d）中文语音识别

训练集：THCHS30

模型结构：双向 RNN，从语音中提取 Mel 倒谱系数，通过三个。



e）交通标志检测

基于形状（矩形或圆形或三角形）检测的 ROI 检测算法：RGB 图片转 HSV 通道；提取图片中的蓝色，红色，黄色；模糊化；二值化；闭运算；去干扰；做裁剪。这样就能将交通标志从图片中标识并截取出来。

原图：

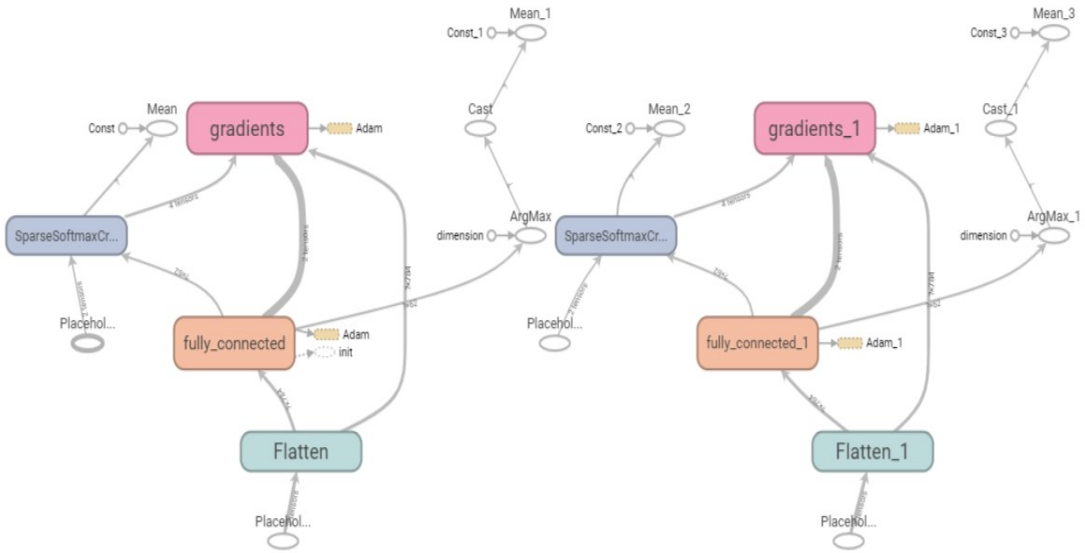
标记后的图片：

f）交通标志识别

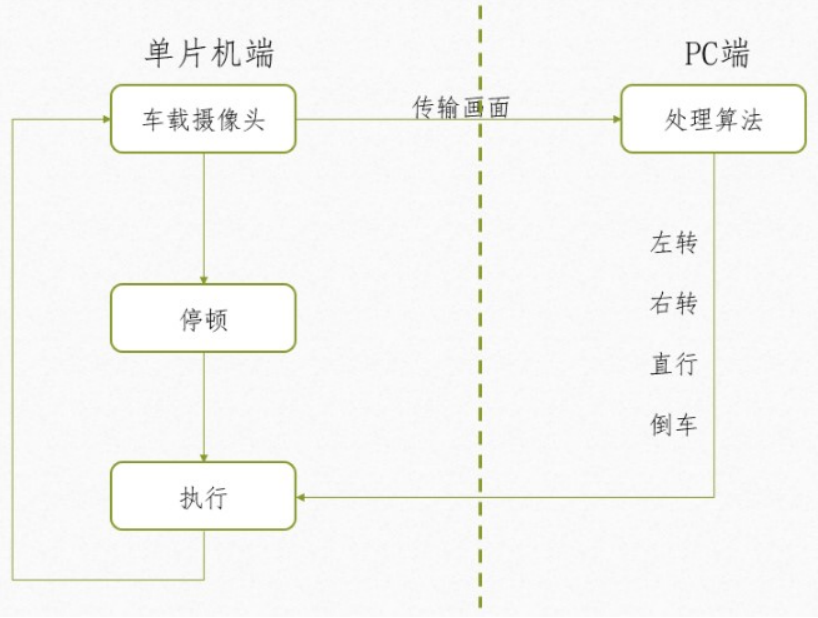
训练集：BelgiumTSC

模型结构：LeNet

Tensorboard 绘制的 Graph：



g） 小车与 PC 的框架



h） 实验结果

小车把车载摄像头拍摄到的前方图像通过服务端视频流的形式传输给 PC，PC 接收到图像后，对图像进行处理，得到小车下一步的响应（前进，后退，左转，右转），并将指令通过蓝牙发送给小车，效果比较理想。

但是，想通过语音控制小车，效果很不好，小车很难识别出前进、后退、左转、右转等指令，还需要进行进一步的改进。目前考虑到的改进方法有：

1、不做全部中文的语音识别，改为范围更小的孤立词语音识别系统，可以使用 DTW 算法，或者基于 GMM 和 HMM 来实现。

2、对录音进行预处理，比如去除噪音，归一化等。

三、本学期的收获

1、学会了使用 tensorflow，pytorch，kersa 等作为工具进行机器学习的工作。

2、同时学会了使用 tensorboard 画图，绘制 graph 来帮助理解模型结构。

3、了解了 Attention 机制在自然语言处理中的应用，能够解决输入长序列所带来的问题。

4、学习了 dropout 算法，来帮助避免出现过拟合情况。

5、提高了对于 GAN 的认识。

6、掌握了迁移学习的要领，养成了做 log 的习惯，利用训练好的网络，提取初级特征，之后只训练最后几层神经元，达到分类的效果。迁移学习已在文本分类、文本聚类、情感分类、图像分类、协同过滤等方面有了应用研究。

7、作为组长，提高了自己的管理能力，能够调动组员的积极性，针对组员的不同擅长之处来分配任务，协同工作，来共同完成项目。

四、展望

1、首先，使用更大的数据集，和加入 dropout 算法来优化中文语音识别。
2、通过基于 GMM 和 HMM 实现中文语音的孤立词识别系统，来实现对智能小车的简单语音控制，如：前进，后退，左转，右转等。
3、实践不同的项目，在实际动手的过程中，掌握更多的机器学习技能，最终能够发表相关论文，提高自己在行业中的竞争力。
4、其实自己对于机器人很感兴趣的，希望之后能参与到一个机器人制作的项目中来，做一个自己的机器人，这个机器人能够与人类进行智能交流，而且能够代替人类做很多事情。