人工智能小车第三阶段报告

组员：袁国韬 2015201934

何忠恒2015201924

李瑞晨2015201932

1. **任务概述**

在前面的智能小车制作过程中，我们已经完成了小车的基本搭建，烧录了可以让小车完成基本避障的代码，实现自动驾驶，蓝牙连接，可以手动操控，加入了液晶显示屏功能。

在最后阶段，考虑到智能小车的现实背景->无人驾驶，在现实的交通中，小车会根据实际路况与交通标志作出自我调整，基于此背景我们实现了小车的智能语音识别与交通标志识别。

1. **任务目标**

本阶段任务目标分为以下二点：

1. 语音识别：具体来说是声音指令识别。也就是小车可以通过接受并识别声音指令，并根据正确的指令作出正确的动作。“前进”对应小车直行，“停止”对应小车停车，“左转”对应小车左转，“右转”对应小车右转。
2. 图像识别：具体来说是交通标志识别。通过识别特定的交通标志来做出特定的动作。
3. **完成过程**
4. **语音识别。**

（1）端点检测和录音

a.使用webrtcvad进行语音端点检测，pyaudio进行语音录制。

b.抽取20ms一帧，计算每一帧的能量大小，如果在400ms内90%的帧能量大于一个阈值，则判断语音开始。

c.语音开始后，若400ms内90%的帧能量低于语音帧能量的20%时，判断语音结束，结束录制。

d.语音开始后，若400ms内90%的帧能量低于语音帧能量的20%时，判断语音结束，结束录制。

（2）语音识别

a.将录制好的语音通过百度语音识别api进行识别，转换成utf-8字符串。

b.由于语音文件较小，在网速不差的情况下，识别速度较快。

（3）命令模糊匹配

使用fuzzywuzzy模块进行命令和识别语句的模糊匹配，例如语音“前行”可以成功识别为“前进”命令，提高了语音命令的准确率。

1. Pybluez

利用python的蓝牙包pybluez将电脑与小车的蓝牙模块连接起来，从而将电脑识别出来的语音指令传给小车并让小车作出相应的动作。

1. **交通标志识别。**
2. 数据集
3. 格式转换：.pmm转换为.jpg处理。
4. 大小统一 ：通过 skimage.transform.resize()函数将大小调整为32\*32
5. 卷积神经网络Tensorflow
6. 模型选取：采用最小可行性模型——单层网络，每个神经元将图片所有像素的RGB值作为输入。是一个完全链接层；
7. 构建 Tensorflow 图：设置Placeholder用来放置图片和标签。定义一个全连接层，使用 ReLU函数作为激活函数，交叉熵来作为损失函数
8. 训练循环：创建Session对象。初始化操作，循环次数设置成了200
9. 模型使用：测试集的准确度能达到90%；

（3）各个交通标志识别情况

[](stop.mp4)

停止：识别后小车停下

[](left.mp4)

左转

[](straight.mp4)

直行

[](right.mp4)

右转

[](passinger.mp4)

人行横道：小车停止3秒后继续前进

1. Opencv
2. IP摄像头

界面展示：

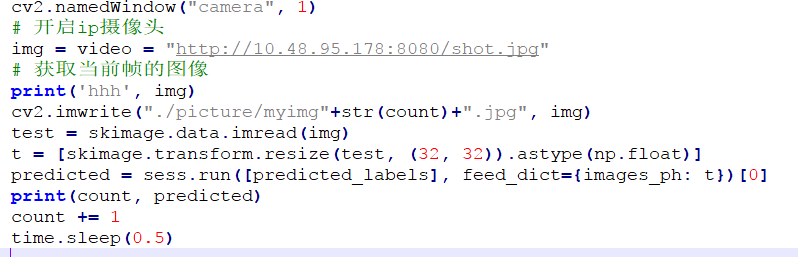


Ip地址及相关信息。（ip地址会在不同局域网下变化）



可以在网页上进行相关的调整设置

代码实现：opeCV获取ip摄像头当前帧图像，并传入神经网络模型中进行预测分析。



1. **遇到的问题与总结**
2. 数据集：不同类型的数据集差异较大，需要调整。
3. 环境搭建：Python版本不兼容，package安装失败
4. 硬件问题：使用蓝牙模块进行频繁的信息传输延迟较高，同时可能出现设备无法处理过多信息的情况
5. 小车电路接触不良：小车电路接触不良导致两个马达的电压波动较大。小车难以以直线前行，各种转弯命令容易出现偏差。

5.总结：组装小车过程中带来的各种电路问题给了全新的挑战，锻炼了动手能力。建立的简单的神经网络模型进行训练，对人工智能算法有了更好的了解。了解了基础的语音和图像的处理方法，增进了多媒体计算机技术的了解。对蓝牙模块和网络socket协议有了更加深入的了解。