

# AI-Car 项目报告

## 人工智能第五小组

### 一、小组成员

姓名	学号	邮箱	联系方式
江流洋	2016202185	jiangliuyang@ruc.edu.cn	18810877326
龙彦兵	2016202137	longyanbing@aliyun.com	15077256756
曹琛	2016202171	2016202171@ruc.edu.cn	18810931912
邓超	2016202184	dengchao@ruc.edu.cn	18810877396

#### 小组成员的分工：

**江流洋：**中文语音识别部分代码识别，交通标志检测与识别部分代码，视频剪辑，ppt 制作，PC 蓝牙监听代码，手机蓝牙控制代码，报告编写，组装小车，项目 github 的维护，项目整合，道路边缘检测代码。

**曹琛：**视频剪辑，ppt 制作，IP 摄像头代码，交通标志检测代码，购买小车零件，组装小车，小车直走、转弯代码。

**龙彦兵：**交通标志检测与识别部分代码，ppt 制作，红外避障、超声波避障、红外循迹代码，购买小车零件，组装小车。

### 二、项目简介

#### 项目背景

现如今，自动驾驶的研究已经成为了一个很重要的研究方向。越来越多的研究人员投入到自动驾驶的队列中来，一方面，生活水平的提高，使得自动驾驶的需求日益提高；另一方面，在自动驾驶方面的成果还远远不能够付诸实践。自动驾驶技术分为多个等级，目前国内外产业界采用较多的为美国汽车工程师协会（SAE）和美国高速公路安全管理局（NHTSA）推出的分类标准。按照 SAE 的标准，自动驾驶汽车视智能化、自动化程度水平分为 6 个等级：无自动化（L0）、驾驶支援（L1）、部分自动化（L2）、有条件自动化（L3）、高度自动化（L4）和完全自动化（L5）。

因为自动驾驶研究的重要性，所以我们小组最后选择的课题也是和自动驾驶有关的。不同的是，我们的项目的目标是实现小车的自动驾驶。项目的名称是“*AI-Car*”。在 github 上的项目地址是 <https://github.com/jaingmengmeng/AI-Car>。

# 项目内容

我们的项目一共可以分为三个部分，分别是 *Random Walker*、*Sensored Walker*、*DeepAI Walker*。下面将以表格的形式简单介绍下各个部分的项目内容。

Random Walker	明确小组分工，购买小车零件，组装小车，熟悉 Arduino IDE，学习 Arduino 编程，编写 Arduino 代码，烧录代码并测试，制作视频和 ppt。
Sensored Walker	明确小组分工，熟悉各个传感器原理（红外传感器、超声波传感器、蓝牙等），编写 Arduino 代码，烧录代码并测试，制作视频和 ppt。
DeepAI Walker	明确小组分工，探究 IP 摄像头、python 蓝牙通信等实现，寻找数据集，编写机器学习代码并调试，编写 Arduino 代码，烧录代码并测试，制作视频和展示 ppt。

## 三、项目过程

### 项目一：Random Walker

#### 1、小车的零件及工具列表：

- 智能小车底盘支架循迹遥控两驱三轮万向轮零配件。
- UNO R3 (CH340G) 数据线和排针。
- Arduino L293D 驱动模块 电机驱动扩展板 L293D 电机驱动马达。
- 超声波模块 超声波测距模块/超声波传感器/HC-SR04。
- BT06 蓝牙串口模块 无线透传数据 51 单片机 兼容 HC-06 DIY arduino。
- 20cm 长 40P 杜邦公头转母头 杜邦线 公对母 面包线 40 根一排。
- 面包线 面包线板 跳线 导线 转接线 面包板扎线。
- SYB-170 小板 迷你小小面包板 彩色面包板 170 孔面包板。
- 单头六角铜柱单通接线螺柱 主板隔离柱 M3\*5+6-M3\*50+6M3\*5+6-M3\*50+6。
- 航模四轴飞行器电池 3.7v 锂电池大容量玩具遥控飞机直升飞机充电器。
- 恒温电烙铁套装家用电焊笔锡焊老铁大功率焊锡枪电子维修焊接工具。

## 2、 ARDUINO IDE



## 3、 前进

```
void run(int time) // 前进  
{  
    digitalWrite(Right_motor_go, HIGH); // 右电机前进  
    digitalWrite(Right_motor_back, LOW);  
    analogWrite(Right_motor_go, 200); //PWM 比例 0~255 调速, 左右轮差异略增减  
    analogWrite(Right_motor_back, 0);  
    digitalWrite(Left_motor_go, LOW); // 左电机前进  
    digitalWrite(Left_motor_back, HIGH);  
    analogWrite(Left_motor_go, 0); //PWM 比例 0~255 调速, 左右轮差异略增减  
    analogWrite(Left_motor_back, 200);  
}  
  
void loop()  
{  
    delay(500);  
    run(10); //前进  
}
```

## 4、转弯

```
void run(int time) { // 前进
    digitalWrite(Right_motor_go,HIGH); // 右电机前进
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); //PWM 比例 0~255 调速，左右轮差异略增减
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,LOW); // 左电机前进
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0); //PWM 比例 0~255 调速，左右轮差异略增减
    analogWrite(Left_motor_back,200);
    delay(time * 100); //执行时间，可以调整
}

void brake(int time) { //刹车，停车
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,LOW);
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,LOW);
    delay(time * 100); //执行时间，可以调整
}

void left(int time) { //左转(左轮不动，右轮前进)
    digitalWrite(Right_motor_go,HIGH); // 右电机前进
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0); //PWM 比例 0~255 调速
    digitalWrite(Left_motor_go,LOW); //左轮不动
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,0); //PWM 比例 0~255 调速
    delay(time * 100); //执行时间，可以调整
}

void spin_left(int time) { //左转(左轮后退，右轮前进)
    digitalWrite(Right_motor_go,HIGH); // 右电机前进
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0); //PWM 比例 0~255 调速
    digitalWrite(Left_motor_go,HIGH); //左轮后退
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);
    analogWrite(Left_motor_back,0); //PWM 比例 0~255 调速
    delay(time * 100); //执行时间，可以调整
}

void right(int time) { //右转(右轮不动，左轮前进)
    digitalWrite(Right_motor_go,LOW); //右电机不动
```

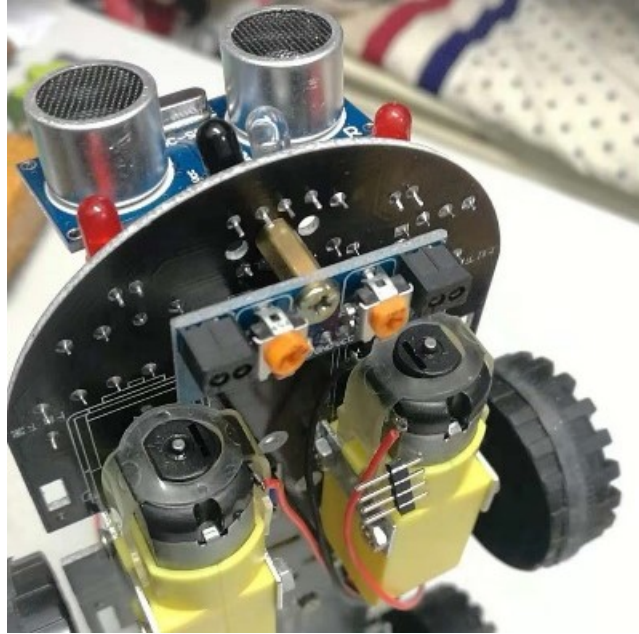
```

digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,0);//PWM 比例 0~255 调速
digitalWrite(Left_motor_go,LOW);//左电机前进
digitalWrite(Left_motor_back,HIGH);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,200);//PWM 比例 0~255 调速
delay(time * 100);    //执行时间，可以调整
}
void spin_right(int time) {          //右转(右轮后退，左轮前进)
digitalWrite(Right_motor_go,LOW);    //右电机后退
digitalWrite(Right_motor_back,HIGH);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,200);//PWM 比例 0~255 调速
digitalWrite(Left_motor_go,LOW);//左电机前进
digitalWrite(Left_motor_back,HIGH);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,200);//PWM 比例 0~255 调速
delay(time * 100);    //执行时间，可以调整
}
void back(int time) {                //后退
digitalWrite(Right_motor_go,LOW);    //右轮后退
digitalWrite(Right_motor_back,HIGH);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,150);//PWM 比例 0~255 调速
digitalWrite(Left_motor_go,HIGH);    //左轮后退
digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,150);
analogWrite(Left_motor_back,0);//PWM 比例 0~255 调速
delay(time * 100);    //执行时间，可以调整
}
void loop() {
delay(2000); //延时 2s 后启动
back(10);    //后退 1s
brake(5);    //停止 0.5s
run(10);     //前进 1s
brake(5);    //停止 0.5s
left(10);    //向左转 1s
right(10);   //向右转 1s
spin_right(20); //向右旋转 2s
spin_left(20); //向左旋转 2s
brake(5);    //停车
}

```

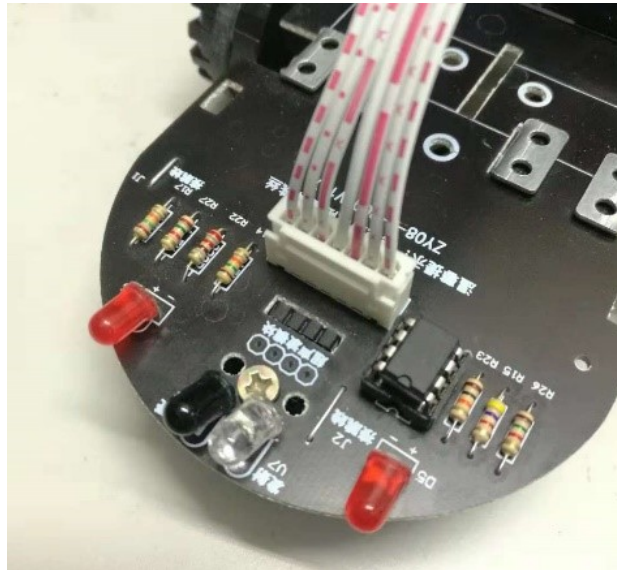
## 项目二：Sensored Walker

### 1. Infrared Following Track



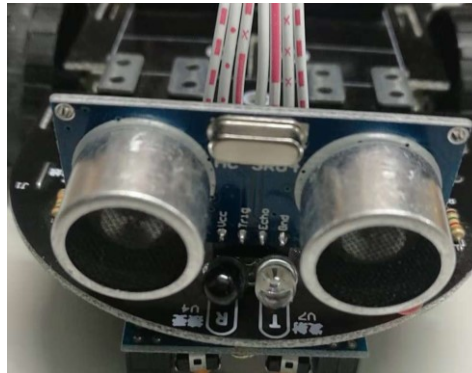
```
void loop()
{
    keyscan(); //调用按键扫描函数，当按键没被按下时处于忙等待状态；当按键被按下
    时，跳出死循环开始红外循迹。
    while (1)
    { // 检测到黑线输出高，信号为 HIGH，小车底部同侧 LED 灯亮；检测到白色区域输出
    低，信号为 LOW，同侧灯灭
        SR = digitalRead(SensorRight);
        SL = digitalRead(SensorLeft);
        if (SL == LOW && SR == LOW)
            run();
        else if (SL == HIGH & SR == LOW)
            // 左循迹红外传感器输出高，检测到黑色区域，车子向右偏离轨道，此时应该向左
            转
            left();
        else if (SR == HIGH & SL == LOW)
            // 右循迹红外传感器输出高，检测到黑色区域，车子向左偏离轨道，此时应该向右
            转
            right();
        else// 左右循迹红外传感器都输出高，车子处在黑色区域内，小车停止
            brake();
    }
}
```

## 2. Infrared Avoid Obstacles



```
void loop()
{
    keyscan(); //调用按键扫描函数，当按键没被按下时处于忙等待状态；当按键被按下
    时，跳出死循环开始红外避障。
    while (1)
    {
        //读取中间的红外避障传感器的状态。有信号为 LOW，表示前方有障碍物；没有信号
        为 HIGH，表示前方没有障碍物。
        S_avoid = digitalRead(Sensor_avoid);
        if (S_avoid == HIGH)
        //前面没有障碍物
        {
            run(); //调用前进函数
            digitalWrite(beep, LOW); //蜂鸣器不响
            digitalWrite(LED, LOW); //LED 灯不亮
        }
        else if (S_avoid == LOW)
        // 前面探测到有障碍物，有信号返回
        {
            digitalWrite(beep, HIGH); //蜂鸣器响
            digitalWrite(LED, HIGH); //LED 亮
            brake(3); //停止 300ms
            back(4); //后退 400ms
            left(5); //调用左转函数 延时 500ms
        }
    }
}
```

### 3. Ultrasonic Avoid Obstacles

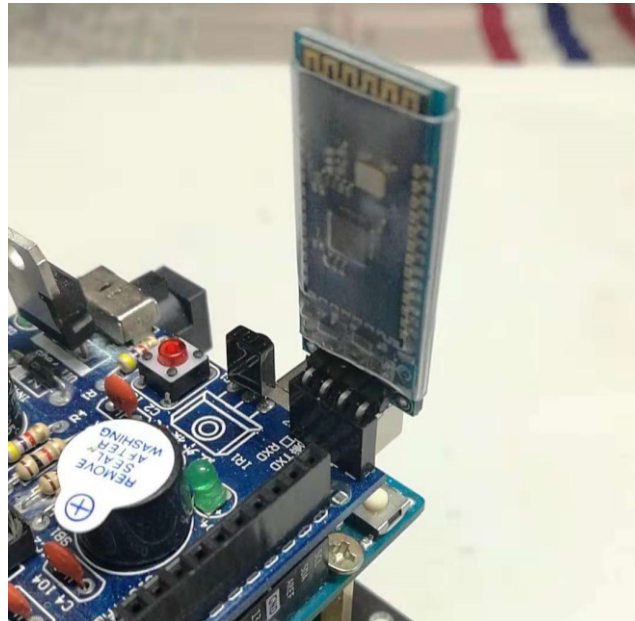


```
float Distance_test() { // 测量小车前方最近障碍物的距离
    digitalWrite(Trig, LOW); // 给触发脚低电平 2μs
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH); // 给触发脚高电平 10μs
    delayMicroseconds(10);
    digitalWrite(Trig, LOW); // 持续给触发脚低电
    float Total_time = pulseIn(Echo, HIGH);
    // 读取回声脚接收到高电平的总时间(单位: 微秒)。
    float Actual_distance = Total_time / 58;
    // 计算实际障碍物的距离(单位: 厘米)。
    Serial.print("Distance:"); //输出距离(单位: 厘米)。
    Serial.print(Actual_distance); //显示距离(单位: 厘米)。
    return Actual_distance;
}

void loop() {
    keyscan(); //调用按键扫描函数, 当按键没被按下时处于忙等待状态; 当按键被按下
    时, 跳出死循环开始超声波避障。
    while (1) {
        Distance = Distance_test(); //调用测距函数测量前方最近障碍物的距离
        if (Distance < 10) {
            //当前方的障碍物的距离小于某个阈值时, 开始后退并向右转一个角度
            back(2);
            right(1); //右转
            digitalWrite(beep, HIGH); //蜂鸣器响起
            brake(1); //停车
        }
        Else {
            run(); //无障碍物, 直行
            digitalWrite(beep, LOW);
        }
    }
}
```



## 4. Bluetooth Telecontrol



```
#define STOP 48           //30h      // 0 0
#define FORWARD 49       //31h      // 1 1
#define BACKWARD 50      //32h      //-1 -1
#define TURNLEFT 51      //33h      // 0 1
#define TURNRIGHT 52     //34h      // 1 0
#define ROTATELEFT 53    //35h      // 0 -1
#define ROTATERIGHT 54   //36h      //-1 0
#define SPINCLOCK 55      //37h      // 1 -1
#define SPINCOUNTERCLOCK 56 //38h     //-1 1

const int LeftMotorBack = 8;
const int LeftMotorGo = 9;
const int RightMotorGo = 10;
const int RightMotorBack = 11;

void setup()
{
    // put your setup code here, to run once:
    Serial.begin(9600);
    pinMode(LeftMotorGo, OUTPUT);
    pinMode(LeftMotorBack, OUTPUT);
    pinMode(RightMotorGo, OUTPUT);
    pinMode(RightMotorBack, OUTPUT);
}
```

```

void loop()
{
    // put your main code here, to run repeatedly:
    //usart read
    if (Serial.available() > 0)
    {
        int cmd = Serial.read(); //读取蓝牙模块发送到串口的数据
        Serial.println(cmd);
        motorRun(cmd);
    }
}

void motorRun(int cmd)
{
    switch (cmd)
    {
        case FORWARD:
            //1 1 左轮和右轮都前进
            Serial.println("FORWARD"); //输出状态
            digitalWrite(LeftMotorGo, HIGH);
            digitalWrite(LeftMotorBack, LOW);
            digitalWrite(RightMotorGo, HIGH);
            digitalWrite(RightMotorBack, LOW);
            break;
        case BACKWARD:
            //-1 -1 左轮和右轮都后退
            Serial.println("BACKWARD"); //输出状态
            digitalWrite(LeftMotorGo, LOW);
            digitalWrite(LeftMotorBack, HIGH);
            digitalWrite(RightMotorGo, LOW);
            digitalWrite(RightMotorBack, HIGH);
            break;
        case TURNLEFT:
            //0 1 左轮不动，右轮前进
            Serial.println("TURN LEFT"); //输出状态
            digitalWrite(LeftMotorGo, LOW);
            digitalWrite(LeftMotorBack, LOW);
            digitalWrite(RightMotorGo, HIGH);
            digitalWrite(RightMotorBack, LOW);
            break;
        case TURNRIGHT:
            //1 0 左轮前进，右轮不动
            Serial.println("TURN RIGHT"); //输出状态
            digitalWrite(LeftMotorGo, HIGH);
            digitalWrite(LeftMotorBack, LOW);
    }
}

```

```

        digitalWrite(RightMotorGo, LOW);
        digitalWrite(RightMotorBack, LOW);
        break;
    case ROTATELEFT:
        //0 -1 左轮不动，右轮后退
        Serial.println("ROTATE LEFT"); //输出状态
        digitalWrite(LeftMotorGo, LOW);
        digitalWrite(LeftMotorBack, LOW);
        digitalWrite(RightMotorGo, LOW);
        digitalWrite(RightMotorBack, HIGH);
        break;
    case ROTATERIGHT:
        //-1 0 左轮后退，右轮不动
        Serial.println("ROTATE RIGHT"); //输出状态
        digitalWrite(LeftMotorGo, LOW);
        digitalWrite(LeftMotorBack, HIGH);
        digitalWrite(RightMotorGo, LOW);
        digitalWrite(RightMotorBack, LOW);
        break;
    case SPINCLOCK:
        //1 -1 左轮前进，右轮后退
        Serial.println("SPIN CLOCKWISE"); //输出状态
        digitalWrite(LeftMotorGo, HIGH);
        digitalWrite(LeftMotorBack, LOW);
        digitalWrite(RightMotorGo, LOW);
        digitalWrite(RightMotorBack, HIGH);
        break;
    case SPINCOUNTERCLOCK:
        //-1 1 左轮后退，右轮前进
        Serial.println("SPIN COUNTERCLOCKWISE"); //输出状态
        digitalWrite(LeftMotorGo, LOW);
        digitalWrite(LeftMotorBack, HIGH);
        digitalWrite(RightMotorGo, HIGH);
        digitalWrite(RightMotorBack, LOW);
        break;
    default:
        //0 0 左轮和右轮都不动
        Serial.println("STOP"); //输出状态
        digitalWrite(LeftMotorGo, LOW);
        digitalWrite(LeftMotorBack, LOW);
        digitalWrite(RightMotorGo, LOW);
        digitalWrite(RightMotorBack, LOW);
    }
}

```

蓝牙模块的实现依赖于安卓蓝牙串口软件，为了避免自己开发，我们选择使用了 [蓝牙串口 SPP](#) 软件（安卓操作系统）。我们可以设置按钮，来向蓝牙 BT04-A 串口发送对应指令来控制小车的前进，后退，停止，左转，右转。编辑按钮的时候可以自己设置 ASCII 码或则 16 进制两种形式，注意要和写的 Arduino 代码匹配起来，只有保持了代码和按钮的编码一致，才能做到让小车按照自己的意愿进行响应。

软件的按钮设置截图以及按钮编辑器截图如下：



**注：**因为 Arduino UNO 板子每次只能烧录一个 Arduino 程序，所以我们在使用小车循迹功能时候，使用的是小车循迹的代码；红外避障又一套代码；超声波避障又一套代码。每次更换小车功能之后都要重新烧录，这样子就很麻烦，但如果用蓝牙遥控的话，就可以随意的切换模式，减少了多次烧录程序的麻烦。

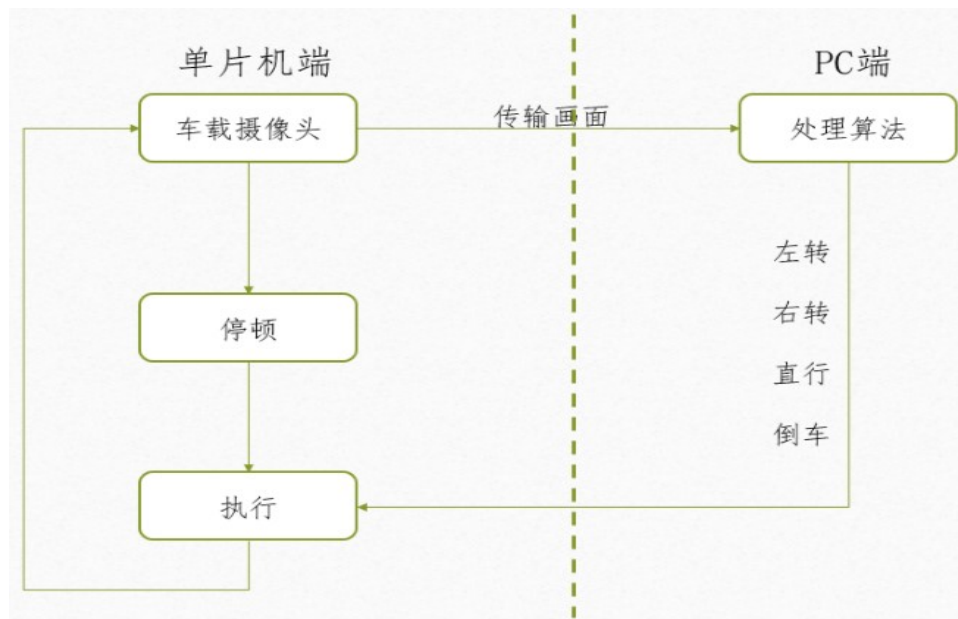


## 项目三：DeepAI Walker

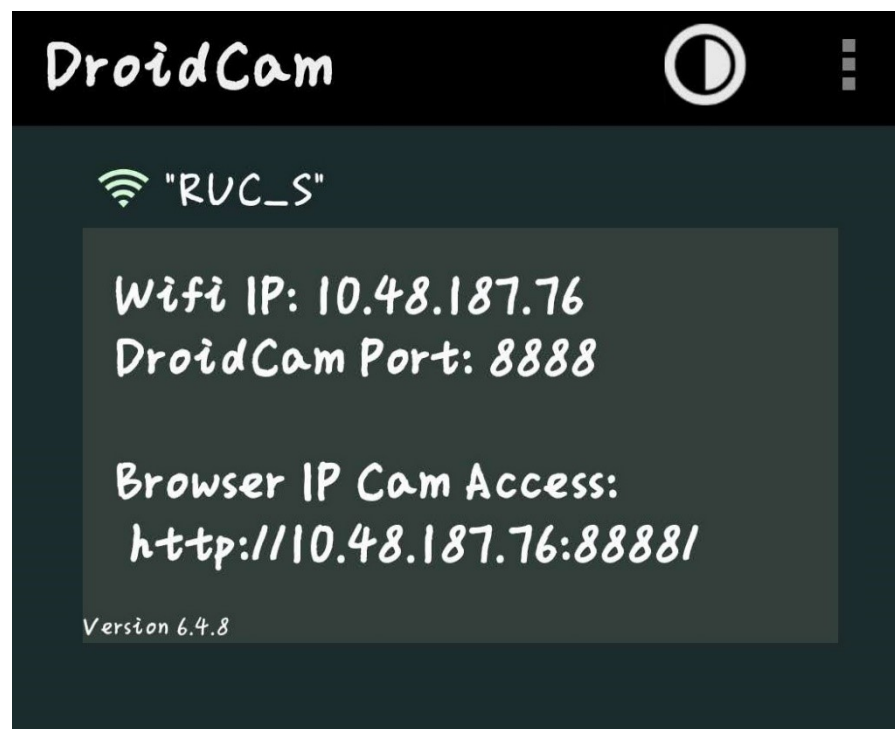
### 1、解决通信问题

将手机放在小车上，在小车上下载 DroidCam 软件作为 IP 摄像头，然后在 PC 端使用 opencv 库来调用接口获取服务端的视频流（也可以用浏览器直接获取服务端视频流）。然后通过截取的视频流，作为输入，传到训练好的机器学习模型之中进行评估，然后根据评估的结果通过用 PySerial 监听蓝牙串口的方式返回到小车，让小车做出相应的响应。

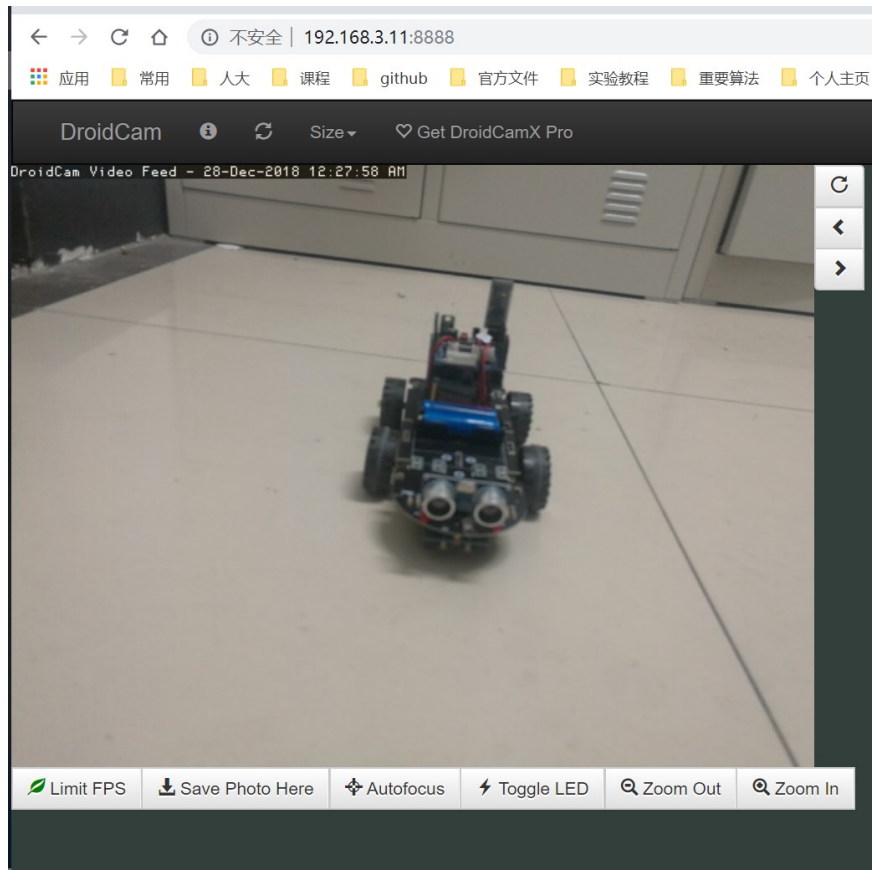
系统的整体框架如图：



DroidCam 手机软件截图：



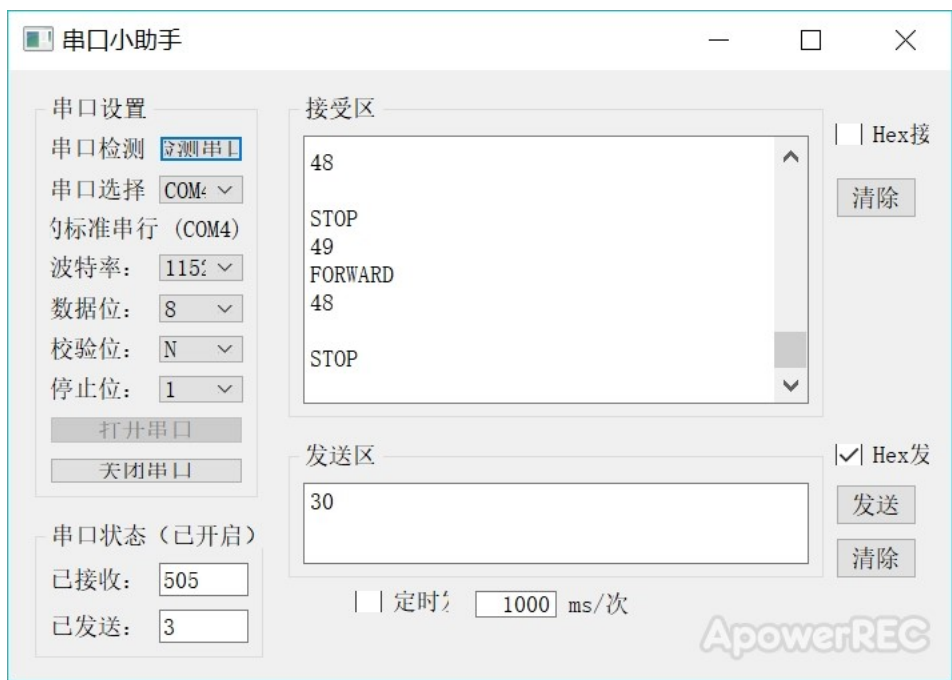
浏览器下获取视频流:



Python DroidCam.py:

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
# cap = cv2.VideoCapture(0)
cap = cv2.VideoCapture('http://10.48.187.76:8888/mjpegfeed')
while(True):
    # read every frame
    ret, frame = cap.read()
    if False == ret:
        break
    # RGB to gray
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    fig, axs = plt.subplots(1, 2)
    axs[0].imshow(frame)
    axs[1].imshow(gray)
    plt.subplot_tool()
    plt.show()
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

## 蓝牙串口监听工具（基于 python PySerial）



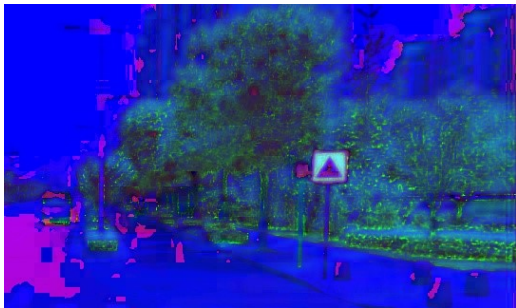
## 2、交通标志识别

交通标志的识别包括两部分，一是交通标志的检测，二是交通标志的识别。

### a) 交通标志的检测

交通标志的检测采用的是基于形状（矩形或圆形或三角形）检测的 ROI 检测算法。

1、将 RGB 转换为 HSV 通道

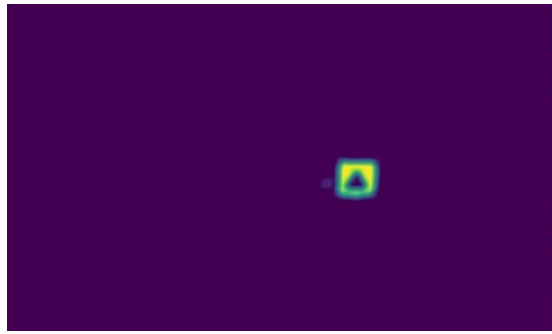


2、颜色过滤（因为交通标志的颜色为蓝色、黄色、红色，均是比较明显的颜色）

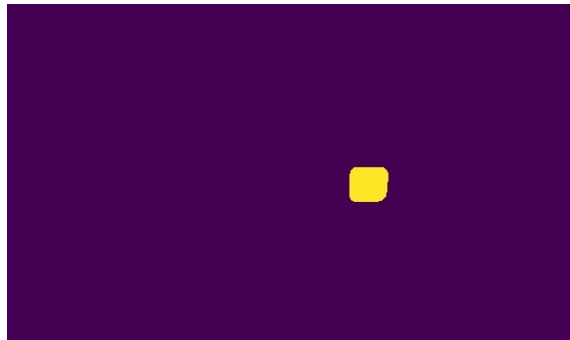




### 3、模糊化处理



### 4、二值化和做闭运算



### 5、得到检测后的图片

检测前的图片——检测后的图片：



### b) 交通标志的识别

1) 数据集：BelgiumTSC（包含 62 种交通信号）

2) 模型结构：LeNet

3) 数据规模变化：

$32 \times 32 \times 1 \rightarrow 30 \times 30 \times 32 \rightarrow 15 \times 15 \times 32 \rightarrow 13 \times 13 \times 64 \rightarrow 6 \times 6 \times 64 \rightarrow 4 \times 4 \times 128 \rightarrow 2 \times 2 \times 128$

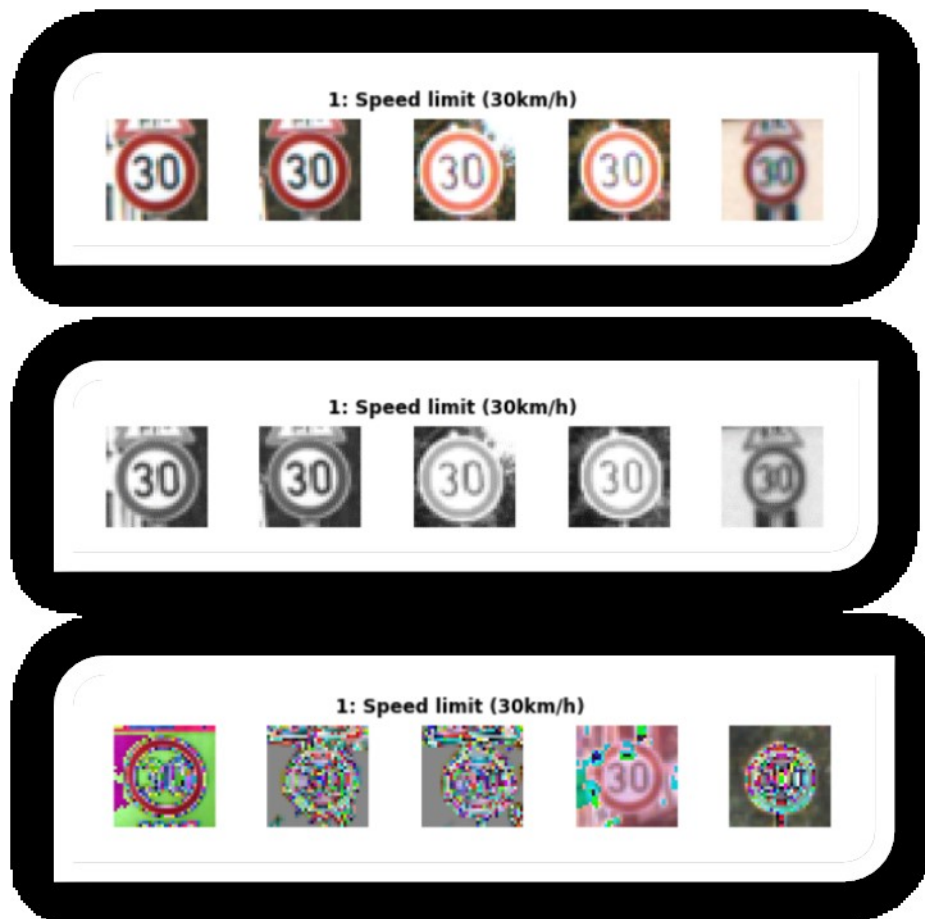
4) 数据预处理过程：

1. 格式转换

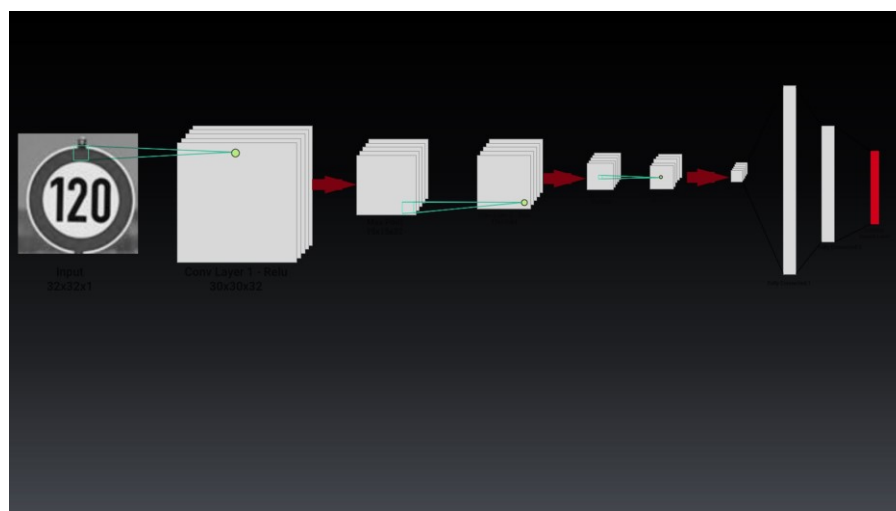


2. 将 RGB 图片转成灰度图

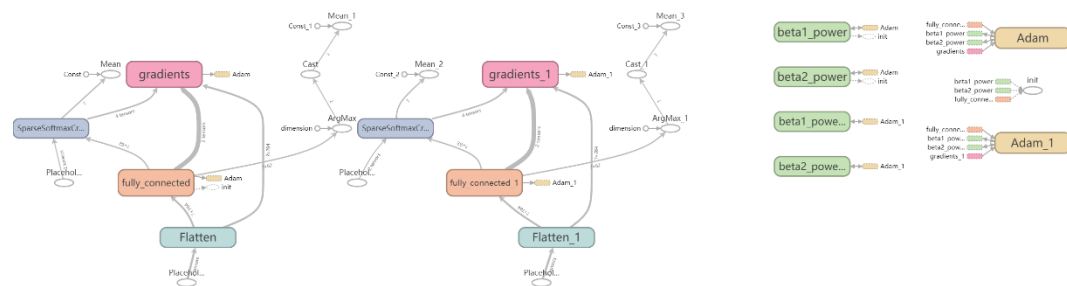
3. 归一化



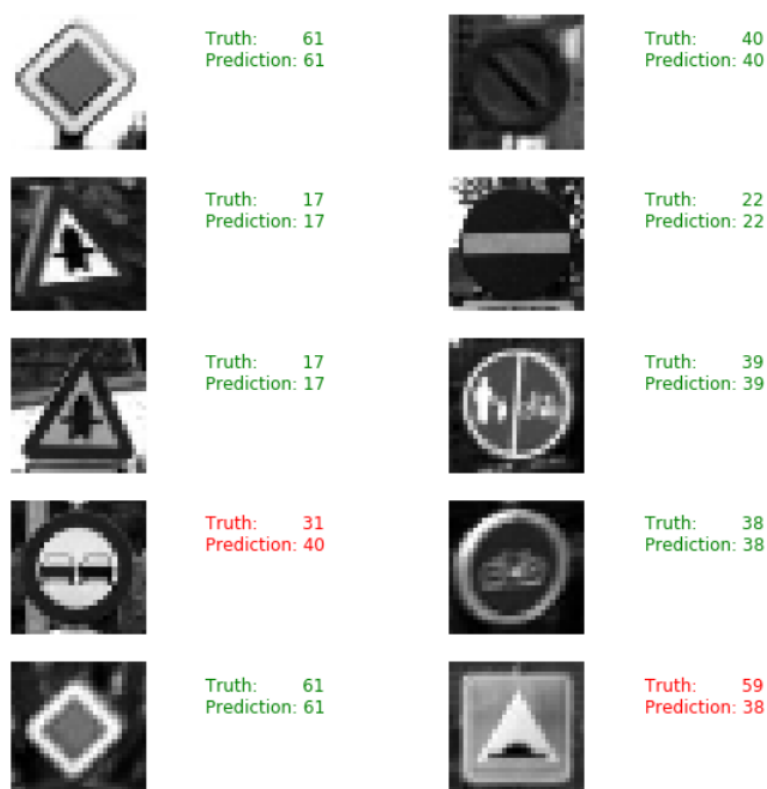
5) LeNet:



## Tensorboard 描绘的 Graph:



## 6) 实验结果



## 7) 准确率: 0.688

## 8) 小车响应

我们的小车对于其他的交通标志只能识别，但是并不会做出响应。但如果识别到前进，后退，左转，右转，停止时，会根据交通标志做出对应的动作。

获取视频流并识别交通标识的 python 代码:

```
import numpy as np
import cv2

# cap = cv2.VideoCapture(0)
```

```

# cap = cv2.VideoCapture('http://10.48.187.76:8888/mjpegfeed')
cap = cv2.VideoCapture('http://192.168.3.54:4747/mjpegfeed')

while(True):
    # read every frame
    ret, img = cap.read()
    img = cv2.resize(img, (500, 500))
    if False == ret:
        break

    hsv=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)

    #提取蓝色区域
    blue_lower=np.array([100,50,50])
    blue_upper=np.array([124,255,255])
    mask=cv2.inRange(hsv,blue_lower,blue_upper)

    #模糊
    blurred=cv2.blur(mask,(9,9))
    #二值化
    ret,binary=cv2.threshold(blurred,127,255,cv2.THRESH_BINARY)

    #使区域闭合无空隙
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (21, 7))
    closed = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel)

    #腐蚀和膨胀
    ...

    腐蚀操作将会腐蚀图像中白色像素，以此来消除小斑点，
    而膨胀操作将使剩余的白色像素扩张并重新增长回去。
    ...

    erode=cv2.erode(closed,None,iterations=4)
    dilate=cv2.dilate(erode,None,iterations=4)

    # 查找轮廓
    image,contours, hierarchy=cv2.findContours(dilate.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    # print('轮廓个数: ',len(contours))
    i=0
    res=img.copy()
    for con in contours:
        #轮廓转换为矩形
        rect=cv2.minAreaRect(con)

```

```

        #矩形转换为 box
        box=np.int0(cv2.boxPoints(rect))
        #在原图画出目标区域
        cv2.drawContours(res,[box],-1,(0,0,255),2)
        #计算矩形的行列

h1=max([box][0][0][1],[box][0][1][1],[box][0][2][1],[box][0][3][1])

h2=min([box][0][0][1],[box][0][1][1],[box][0][2][1],[box][0][3][1])

l1=max([box][0][0][0],[box][0][1][0],[box][0][2][0],[box][0][3][0])

l2=min([box][0][0][0],[box][0][1][0],[box][0][2][0],[box][0][3][0])
        #加上防错处理，确保裁剪区域无异常
        if h1-h2>0 and l1-l2>0:
            #裁剪矩形区域
            temp=img[h2:h1,l2:l1]
            i=i+1
            #显示裁剪后的标志
            # cv2.imshow('sign'+str(i),temp)

h_merge = np.hstack((img, res)) #水平拼接
cv2.imshow('h_merge', h_merge)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

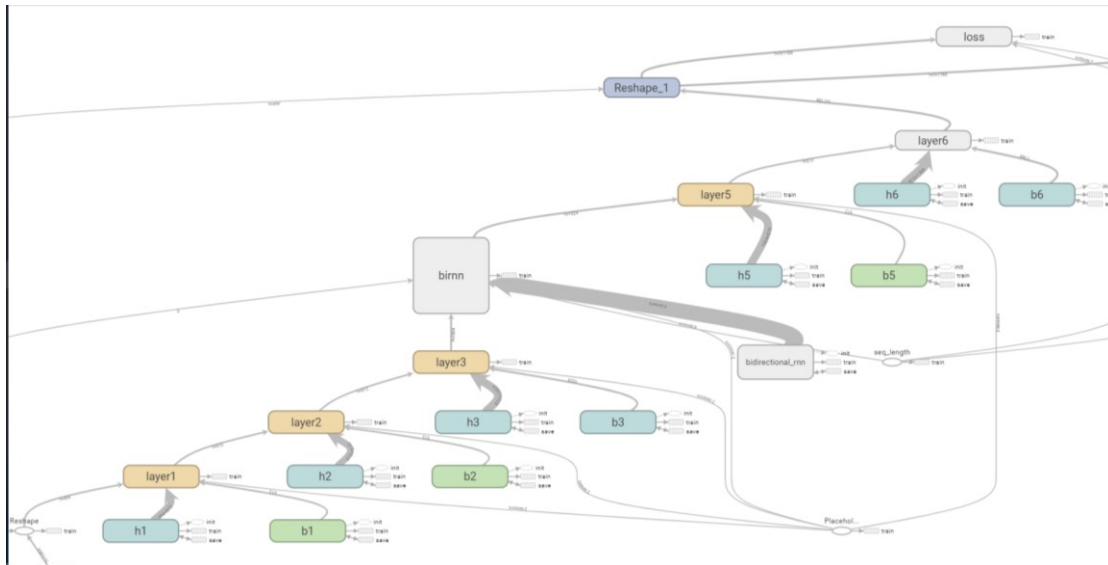
cap.release()
cv2.destroyAllWindows()

```

### 3、中文语音识别

- 1) 数据集：清华大学中文语料库（THCHS30）
- 2) 分词：清华大学中文分词系统（THULAC）VS jieba
- 3) 词向量：FastText
- 4) 模型：双向 RNN
- 5) 目标：中文语音的识别，希望小车能听懂人的所有会话（区别于孤立词的语音识别）

## Tensorboard 描绘的 Graph:



## 训练过程中的测试阶段（跑的数据集中的测试集）:

```
标记C:\Windows\System32\cmd.exe - python test.py
Microsoft Windows [版本 10.0.17134.472]
(c) 2018 Microsoft Corporation. 保留所有权利。

D:\Young\Young\Projects\AI\speech_recognition>python test.py
# Config
# Load waves and labels
wav: 2132
label: 2132
# Create dictionary
字表大小: 1787
# Build Model
# Begin to test
ckpt: ./opt/tf/speechtest/speech.cpkt-101
startepo: 101

读入语音文件: ./opt/wav/test/D11/D11_773.wav
开始识别语音数据.....
语音的原始文本: 台湾 婴幼儿 营养 专家 还 与 厦门市 同行 就 婴幼儿 的 生长 发育 进行 专题 探讨
识别出来的文本: 台湾 婴幼儿 营养 专家 还 厦门市 同行 就 婴幼儿 的 生长 发育 进行 专题 探讨
读入语音文件: ./opt/wav/test/D11/D11_774.wav
开始识别语音数据.....
语音的原始文本: 像 张 王村 八十年代 四百 户 拥有 耕牛 七百 多头 而 今 五百 五十多 农户 仅 拥有 耕牛 十一 头
识别出来的文本: 像 张 王村 八十年代 四百 户 拥有 耕牛 七百 多头 而 今 五百 五十多 农户 仅 拥有 耕牛 十一 头
读入语音文件: ./opt/wav/test/D11/D11_775.wav
开始识别语音数据.....
语音的原始文本: 看 羊 狗 跑前跑后 一只 惊 飞 的 山雀 惹得 它 汪汪汪 咬 几 声 省 喻 喻 喻 的 在 山 间 回 荡
识别出来的文本: 看 羊 狗 跑前跑后 一只 惊 飞 的 山雀 惹得 它 汪汪汪 咬 几 声 省 喻 喻 喻 的 在 山 间 回 荡
读入语音文件: ./opt/wav/test/D11/D11_776.wav
开始识别语音数据.....
语音的原始文本: 承运人 有权 要求 托运 人 填写 航空 货运单 托运 人 有权 要求 承运人 接受 该 航空 货运单
识别出来的文本: 承运人 有权 要求 托运 人 填写 航空 货运单 托运 人 有权 要求 承运人 接受 该 航空 货运单
读入语音文件: ./opt/wav/test/D11/D11_779.wav
开始识别语音数据.....
语音的原始文本: 当然 他们 不 像 鳄鱼 那样 吞下 石块 当 做 压 舱 石 而 是 磨 成 粉 以 后 才 服 用
识别出来的文本: 当然 他们 不 像 鳄鱼 那样 吞下 石块 当 做 压 舱 石 而 是 磨 成 粉 以 后 才 服 用
读入语音文件: ./opt/wav/test/D11/D11_780.wav
开始识别语音数据.....
语音的原始文本: 刀 光 枪 影 白 骨 如 麻 我 仿 佛 听 到 五 十 年 前 三 十 万 冤 魂 的 呐 喊 痛 贯 心 肝
识别出来的文本: 刀 光 枪 影 白 骨 如 麻 我 仿 佛 听 到 五 十 年 前 三 十 万 冤 魂 的 呐 喊 痛 贯 心 肝
读入语音文件: ./opt/wav/test/D11/D11_781.wav
开始识别语音数据.....
语音的原始文本: 不 仅 要 宣 传 少 生 还 要 宣 传 晚 婚 晚 育 优 生 优 育 宣 传 生 男 生 女 都 一 样
识别出来的文本: 不 仅 要 宣 传 少 生 还 要 宣 传 晚 婚 晚 育 优 生 优 育 宣 传 生 男 生 女 都 一 样
```

## 训练过程中的测试阶段（跑的自己录制的后退“二字”的音频）:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.472]
(c) 2018 Microsoft Corporation. 保留所有权利。

D:\Young\Young\Projects\AI\speech_recognition>python test_one.py
# Config
# Load waves and labels
wav: 2132
label: 2132
# Create dictionary
字表大小: 1787
# Build Model
# Begin to test
ckpt: ./opt/tf/speechtest/speech.cpkt-101
startepo: 101

读入语音文件: 后退.wav
开始识别语音数据.....
WARNING:root:frame length (1103) is greater than FFT size (512), frame will be truncated. Increase NFFT to avoid.
识别出来的文本: 尽 不 运 的
```

## 6) 实验结果:

训练出来的模型，在数据集的测试集上的效果很好，但是在自己生成的测试用例上的效果很差劲，以致于都不能识别出最简单的“前进”、“后退”、“左转”、“右转”、“停止”等指令。

分析其原因，一是算法中没有加入 dropout 算法，可能导致训练结果过拟合；二是本身因为硬件原因，训练集并没有选用整个 THCHS30 数据集的训练集，而是选取了其中的一部分做为训练集，也可能导致最后结果的不准确；三是本身训练集对于多音字的处理不够完善；四是录制的音频由于录音环境的问题包含了很多噪音。所以，最后的实验效果并不是很好，原本希望通过说出简单的语音来控制小车，但最后因为语音识别的效果太差，所以没能达到预期的目标。期望的改进方法是做中文语音的孤立词语音识别，将语音的范围固定在小车的那些指令里，期望提高效率。

## 四、项目总结

通过这一次的项目，我们学会了很多东西。

### 一、 Arduino 编程

三个项目都需要进行 Arduino 编程,进而达到控制小车动作的目的。一开始的小车,只能前进后退转弯等;后来加入了传感器,又实现了通过传感器来控制小车的动作;后来又加入了蓝牙模块,就能通过手机蓝牙来控制。其中蓝牙模块的控制选用的是[蓝牙 SPP](#) 软件,能够监听蓝牙串口的消息,通过消息来控制小车动作。又因为 Arduino 的 UNO 板子每次只能烧录一个程序,所以对于不同的控制程序,需要不同的 Arduino 代码,多次烧录会比较麻烦,所以我们小组研制了遥控来更改小车的控制程序的方法,只需要烧录一套代码就可以了,如,可以用不同的蓝牙遥控按钮来对应更改小车的不同控制程序,一键超声波避障,一键红外避障,一键红外循迹。

### 二、 用 Python 实现和小车通信

在项目的第三阶段,由于需要将小车和电脑进行通信,一是获取手机摄像头视频流,二是电脑通过监听蓝牙串口来给小车发控制消息。对于获取手机 IP 摄像头,在手机端安装 DroidCam 软件,然后手机端和电脑端连同一个局域网,就可以在电脑使用 opencv 库的 API 来获取服务端视频流。对于电脑端监听蓝牙串口,使用的是 PySerial 包来监听串口,接收来自小车信息并发送信息到小车。

### 三、 Tensorflow

在 DeepAI Walker 阶段,我们使用了 tensorflow 编写机器学习的代码,参照一些论文和博客,构建了机器学习的模型,模型的选用包括 LeNet、BiRNN,进行模型的训练。其中,交通标志的识别部分,我们加入了交通标志的检测,可以将小车前方的图像中的交通标志检测并提取出来,作为模型的输入进行评估,得到返回结果,再通过蓝牙将指令发送给小车。然后在中文语音识别部分,最终的实验结果并不好,不能达到预期的用语音控制小车的效果,我们希望做出进一步的改进,是做中文语音的孤立词语音识别。同时在项目过程中,使用了 tensorboard 绘制模型的 Graph,可以帮助我们更好的理解模型的结构。