

基于 Arduino 的人工智能小车

课程设计报告

课程名称：人工智能

报告人：杨文清

学号：2015202039

组员：赵元培（组长）、胡景文

摘要

本文介绍了基于 arduino 的人工智能小车的设计及制作实现过程。共分为三个阶段：1. 随机行走 2. 超声波传感器避障 3. 图像识别。第一阶段主要是组装小车，并通过 arduino 及驱动板控制小车进行随机行走；第二阶段加入了超声波测距，使得小车在靠近障碍物时能够及时调整方向，避开障碍；第三阶段接入了实时摄像，通过图像识别算法，小车做出相应的判断。三个阶段由浅到深，完成了从能够简单行动到具有实时分析能力的跨越。

小车的控制核心是 Arduino 单片机，机器学习算法也是由 Arduino 单片机指导小车行动从而体现的。在第一阶段中，由于没有外部传感器的输入数据，小车的行动完全随机。第二阶段，接入了超声波传感器的测距，arduino 通过判断前方障碍物距离，指导小车在与障碍物打到临界距离时进行转向。第三阶段，arduino 会判断当前行驶路段(红色)，并识别前方的标志(转弯、红绿灯)，指导小车作出相应的选择。

目录

一、 背景介绍.....	4
二、 随机行走的小车设计.....	5
三、 智能避障的小车设计.....	7
四、 基于图像识别的智能小车设计.....	10
五、 总结.....	14

一、背景介绍

1、时代背景

人工智能 (Artificial Intelligence)，英文缩写为 AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能的核心在于计算和数据，随着计算机的不断发展，计算能力不断提高，数据量不断增大。人工智能的快速发展已是大势所趋。日前，工信部印发《促进新一代人工智能产业发展三年行动计划(2018-2020 年)》，提出力争到 2020 年，一系列人工智能标志性产品取得重要突破，在若干重点领域形成国际竞争优势。其中“自动驾驶汽车”的发展便是其中之一。在这样的时代下，计算机专业的大学生学习一些人工智能的知识，并有所实践，无疑是非常好的。本次课程设计中，我们的目标便是使用 Arduino 单片机，并实现一些简单的机器学习算法，使得小车具有简单的“智能”。

2、Arduino 介绍

Arduino，是一个基于开放原始码的软硬件平台，构建于开放原始码 simple I/O 介面版，并且具有使用类似 Java, C 语言的 Processing/Wiring 开发环境。它包含两个主要的部分：硬件部分是可以用来做电路连接和 Arduino 电路板；另外一个则是 Arduino IDE，你的计算机中的程序开发环境。你只要在 IDE 中编写程序代码，将程序上传到 Arduino 电路板后，程序便会告诉 Arduino 电路板要做什么事了。

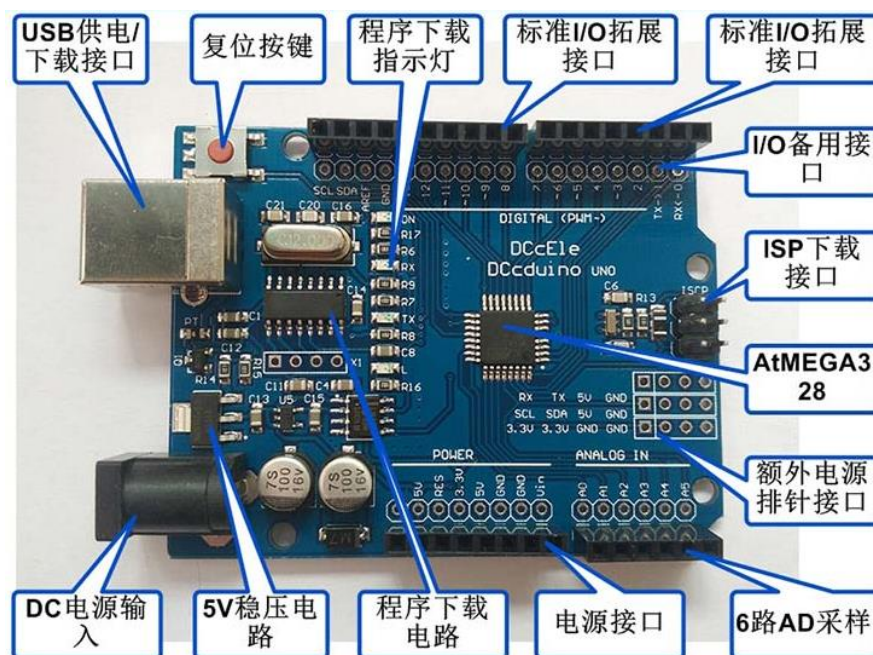


图 1 Arduino 实物图

Arduino 能通过各种各样的传感器来感知环境，通过控制灯光、马达和其他的装置来反馈、影响环境。板子上的微控制器可以通过 Arduino 的编程语言来编写程序，编译成二进制文件，收录进微控制器。对 Arduino 的编程是利用 Arduino 编程语言（基于 Wiring）和 Arduino 开发环境（based on Processing）来实现的。基于 Arduino 的项目，可以只包含 Arduino，也可以包含 Arduino 和其他一些在 PC 上运行的软件，他们之间进行通信（比如 Flash, Processing, MaxMSP）来实现。可以自己动手制作，也可以购买成品套装；Arduino 所使用到的软件都可以免费下载。硬件参考设计（CAD 文件）也是遵循 available open-source 协议，你可以非常自由地根据你自己的要求去修改他们。

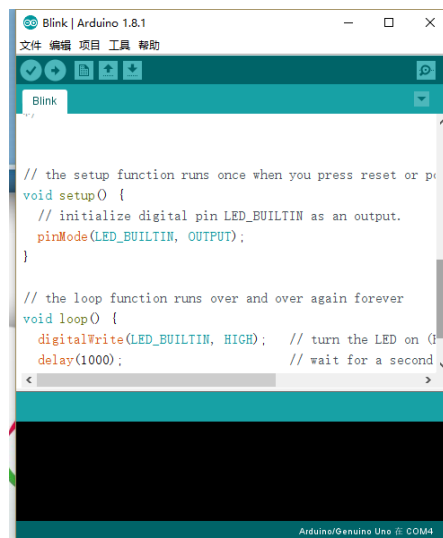


图 2 Arduino IDE

3、 课程设计主要内容

本次课程设计，主要基于 arduino，实现一些简单的驱动控制及机器学习算法，共分为三个阶段：1. 随机行走 2. 超声波传感器避障 3. 图像识别。第一阶段主要是组装小车，并通过 arduino 及驱动板控制小车进行随机行走；第二阶段加入了超声波测距，使得小车在靠近障碍物时能够及时调整方向，避开障碍；第三阶段接入了实时摄像，通过图像识别算法，小车做出相应的判断。三个阶段由浅到深，完成了从能够简单行动到具有实时分析能力的跨越。

二、 随机行走的小车设计

1、 阶段目标

- 完成硬件的组装搭建
- 熟悉软件（IDE 等）的安装使用
- 编写程序，控制小车随机行走。

2、 硬件结构说明

小车底板：左右 2 个轮子，分别由一个 motor 控制

Arduino uno 板：用于烧录程序，连接驱动板等设备

L293D 驱动板：连接 2 个 motor，连接蓝牙串口，连接超声波传感器

电池：连接驱动板，提供电源

电动机 2 个

面包板 1 个

连接线若干

3、 组装过程

将【Arduino UNO 主板】与【L293D 电机驱动扩展板】连接

将【小车底座、车轮】组装好，与【Arduino UNO 主板】连接

将【电动机 2 个】与【小车底座】组装好，与【L293D 电机驱动扩展板】连接

利用连接线，通过【面包板】，将【电源】接入【L293D 电机驱动扩展板】固定以及焊接各个部分

4、 接口说明

电动机接口：两个电动机分别接入【L293D 电机驱动扩展板】的【M1】【M4】接口

5、 源代码

```
#include "AFMotor.h"
//设置两个 motor 的接口和频率
AF_DCMotor motor_left(1,MOTOR12_1KHZ);
AF_DCMotor motor_right(4,MOTOR34_1KHZ);
//设置两个舵机接口
const int TrigPin = 9;
const int EchoPin = 10;
//距离变量
float cm;
void setup()
{
    Serial.begin(9600);
    pinMode(TrigPin, OUTPUT);
    pinMode(EchoPin, INPUT);
    //设置左右 motor 速度
    motor_left.setSpeed(180);
    motor_right.setSpeed(210);
    //设置左轮速度 180，右轮速度 210（由于 M1 和 M4 频率不同，需要微调）
}
```

6、 阶段成果

组装后的小车能够正常运动，由于两轮频率不一致，调整之后的小车基本能够按照直线进行运动。通过修改左右轮的速度，可以进行随机行走。

三、 智能避障的小车设计

1、 阶段目标

通过接入超声波传感器检测到的前方障碍物据小车的距离，小车进行智能判断：转弯或倒车。

通过蓝牙控制小车

2、 新增硬件

HC-SR04 超声波传感器：探测前方障碍物，将数据传回主板

BT06 蓝牙串口：提供远程控制接口

3、 硬件设计框架

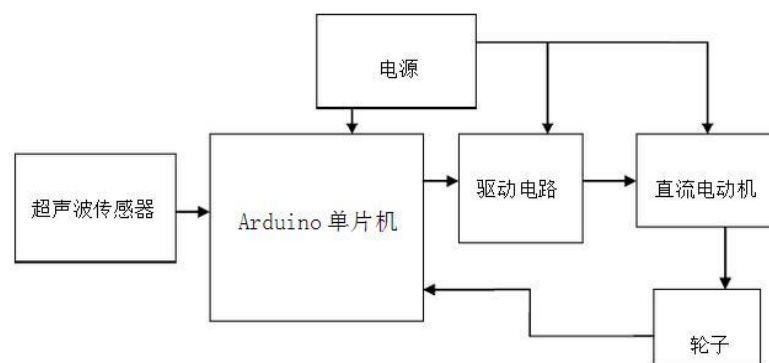


图 3 硬件结构图

4、 组装过程

利用连接线，将【蓝牙接收器】接入【L293D 电机驱动扩展板】的舵机接口

利用连接线，将【超声波距离探测器】接入【L293D 电机驱动扩展板】的舵机接口

5、 接口说明

蓝牙接口：

【蓝牙接收器】接入舵机接口，编号为【9】

- VCC：接 Arduino 的 5V

- GND: 接 Arduino 的 GND
- TXD: 一般表示为自己的发送端, 接 Arduino 的 RX
- RXD: 一般表示为自己的接收端, 接 Arduino 的 TX

距离探测器接口:

【超声波距离探测器】的【trig】端接入舵机【9】接口

【超声波距离探测器】的【echo】端接入舵机【10】接口

6、流程图

参考流程图如下

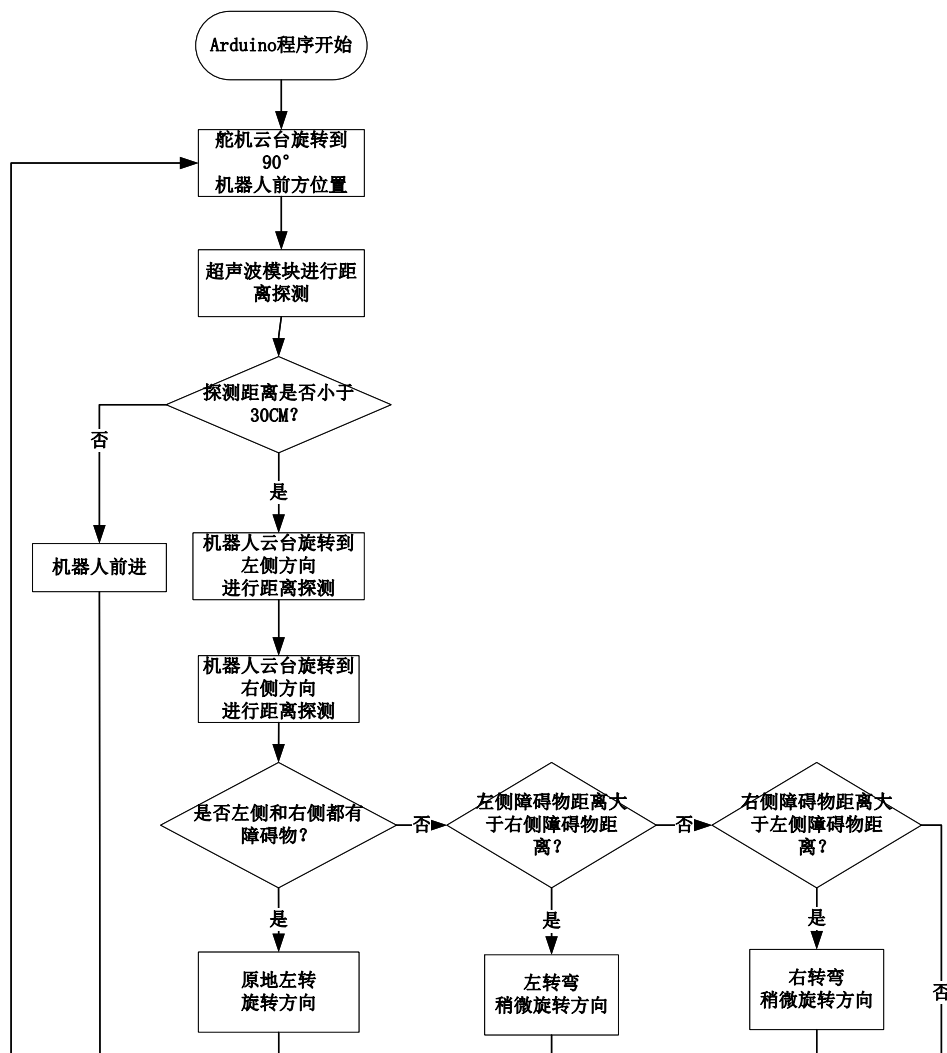


图 4 流程图

7、 源代码

7.1 距离检测及自动控制模块


```

void loop() //距离控制函数
{
    digitalWrite(TrigPin, LOW); //低高低电平发一个短时间脉冲去 TrigPin
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin, LOW);

    cm = pulseIn(EchoPin, HIGH) / 58.0; //将回波时间换算成 cm
    cm = (int(cm * 100.0)) / 100.0; //保留两位小数

    if(cm < 15){ //障碍物过近 (0~15cm) 反向旋转，实现倒车
        motor_left.setSpeed(180);
        motor_left.run(FORWARD);
        motor_right.run(FORWARD);
        delay(300);
    }
    //面对正常障碍物 (15~40cm): 设置左轮速度 0，实现左转
    else if(cm < 40){
        motor_left.setSpeed(0);
        motor_left.run(BACKWARD);
        motor_right.run(BACKWARD);
        delay(300);
    }
    else{
        motor_left.setSpeed(180);
        motor_left.run(BACKWARD);
        motor_right.run(BACKWARD);
    }
    delay(100);
}

```

7.2 蓝牙远程控制模块

```

    if(ch == 'F'){
//前进
        Serial.println("up");
        motor_left.setSpeed(210);
        motor_right.setSpeed(180);
        motor_left.run(BACKWARD);
        motor_right.run(BACKWARD);
    }
    else if(ch == 'B'){
//后退

```

```

        Serial.println("back");
        motor_left.setSpeed(210);
        motor_right.setSpeed(180);
        motor_left.run(FORWARD);
        motor_right.run(FORWARD);
    }

```

8、 阶段成果及不足

遇到远处障碍物能够转向
 无法转向时能够倒车
 超声波传感器位置较高，遇到较低障碍物无法检测并避开
 超声波传感器宽度不足，小车两侧边缘可能被卡住

四、 基于图像识别的智能小车设计

1、 阶段目标

完成手机与 PC 端视频讯号的传递 (DroidCamApp)
 利用 python 处理图片、视频
 实现简单的图像识别算法
 小车通过识别路标及路面信息进行运动

2、 视频讯号传递

使用软件: DroidCamApp
 传输延迟: USB < WLAN < Bluetooth
 控制逻辑: 通过 python 程序实时分析当前小车运转过程中摄像头捕获的图像, 将得出的结果通过蓝牙发送给小车, 最终由 Arduino 控制小车行动。

3、 图像识别算法调研

3.1 pyssim

Pyssim 是 python 中的一个工具包, 可以用来计算两张图片的相结构相似性。该方法能对两张图片的相似度进行一个大致的区分, 它的计算基于两张图片的亮度、对比度和结构。返回一个-1~1 之间的实数。

Pyssim 调用简单, 但是效果并不好, 仅可以用为一个补充, 而不能作为算法核心。

3.2 相似性度量

在做分类时常常需要估算不同样本之间的相似性度量 (Similarity Measurement), 这时通常采用的方法就是计算样本间的“距离” (Distance)。采用什么样的方法计算距离是很讲究, 甚至关系到分类的正确与否。常见的距离有

欧氏距离、夹角余弦、切比雪夫距离、闵可夫斯基距离、马氏距离等等。

在图片相似度分析中，常用的距离是夹角余弦距离，由于图像的本质是 RGB 值的数据组合，可以通过将两张图片缩减到相同的规格，向量每一维的值为对应像素点 RGB 值的平均，然后计算图片平均。但是这样的计算量过大，速度过慢，不适用于小车的实时判断。

3.3 色彩分布直方图

色彩分布直方图首先通过将图片转化成 RGB 对应的颜色分配图。然后判断直方图是否接近。

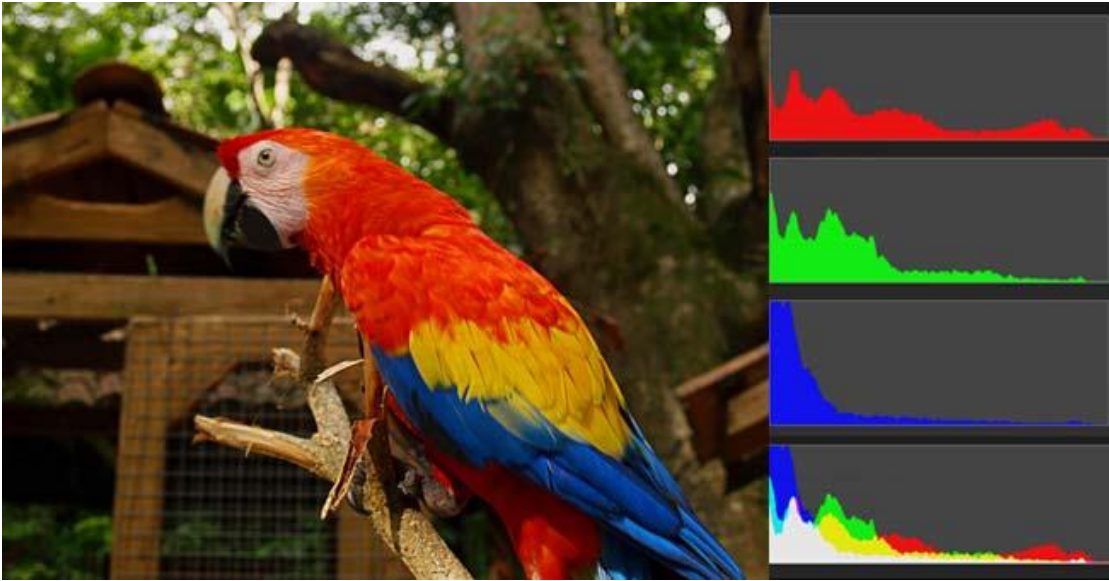


图 5 一张图片的色彩分布直方图

任何一种颜色都是由红绿蓝三原色（RGB）构成的，所以上图共有 4 张直方图（三原色直方图 + 最后合成的直方图）。

如果每种原色都可以取 256 个值，那么整个颜色空间共有 1600 万种颜色（256 的三次方）。针对这 1600 万种颜色比较直方图，计算量实在太大了，因此需要采用简化方法。可以将 0~255 分成四个区：0~63 为第 0 区，64~127 为第 1 区，128~191 为第 2 区，192~255 为第 3 区。这意味着红绿蓝分别有 4 个区，总共可以构成 64 种组合（4 的 3 次方）。

任何一种颜色必然属于这 64 种组合中的一种，这样就可以统计每一种组合包含的像素数量。

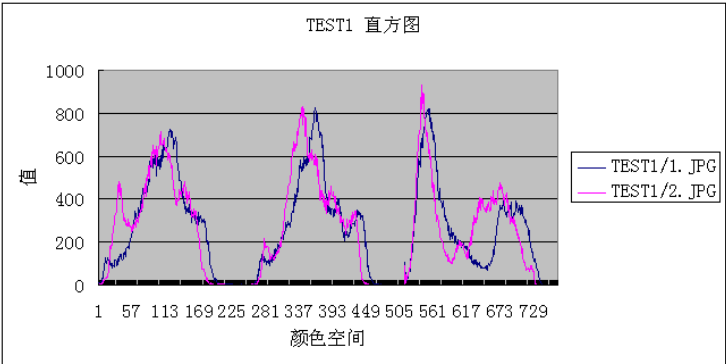


图 6 两张图片的对比

对于小车的行进过程中图片的分析，因为小车运行的道路为红色，所以对于道路的判断可以根据 RGB 的分布进行判断，非常适合当前场景。但由于 RGB 组合过多，计算依旧很复杂，考虑到需要实时判断，需要压缩延迟，所以再引入灰度直方图算法。

如果两张图片很相似，它们的黑白轮廓应该是相近的。对于路标的判断可以运用这种方法。

4、神经网络的应用

4.1 神经网络简介

神经网络的核心是通过大量的神经元加权得出结果，不断地进行逼近，经由界定阈值判断得出最终的决策结果。

神经网络的好坏，决定性因素就是权重及阈值的大小，通常初始值由人为给定，经过训练集的不断训练，得出一个足够贴近实际情况的权重及阈值，这样就得到了一个很好的现实模型。

因此神经网络的使用过程应该具备以下几步：

- 1、确定输入和输出
- 2、找到一种或多种算法，可以从输入得到输出
- 3、找到一组已知答案的数据集，用来训练模型，估算 w （权重）和 b （阈值）
- 4、一旦新的数据产生，输入模型，就可以得到结果，同时对 w 和 b 进行校正

需要注意的是，神经网络的运作过程需要海量的计算。

4.2 课程设计中的应用

依据上文中的步骤，具体到智能小车的实现中，应该分为以下几步：

1. 输入数据为摄像头捕获的实时图像，输出为小车的运动指令。
2. 依赖色彩分布直方图算法为核心，兼用相似性度量算法，由图像转换到运动指令。
3. 通过前期多角度地对实际场景拍照及摄像，提供训练集，提高权重及阈值的精度
4. 将实际运动过程中的图像提交给模型，得出小车的运动指令。并在此过程中继续修正权重和阈值。

4.3 训练集及实际场景展示



图 7 红色卡纸铺成的道路及转向箭头



图 8 红色纸板

红色的纸板为模拟斗牛场景，小车发现前方出现红色纸板会加速前进

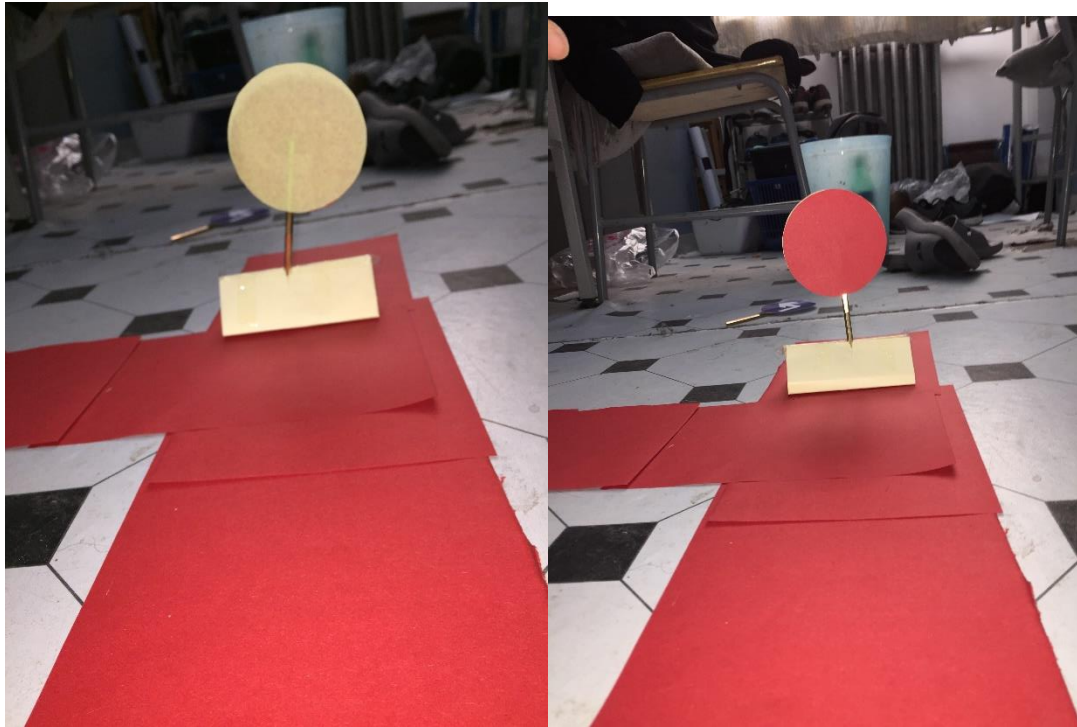


图 9 红绿灯

小车发现前方出现红色提示灯会停止行动，发现绿色提示灯会前进。

5、 阶段目标

编写神经网络模型，综合使用多种图像识别算法，通过训练集得出贴合实际情况的权重及阈值，构建出模型。处理实际的输入图像，指导小车进行正确的判断。

五、 总结

在本次课程设计中，我主要参与的部分为前期小车的制作，接通 Arduino 单片机，完善蓝牙串口等各个部件的组装，以及后期图像处理的算法调研，以及神经网络算法的实际运用。在这个过程中，一方面提高了我的动手能力，尤其是针对复杂的 Arduino 单片机以及引脚的焊接，我学习到了很多的电子知识，提高了我对于陌生硬件的学习能力。另一方面学习并实际运用了神经网络和图像识别的相关算法，对人工智能有了一个更加深入的理解。

人工智能时代已经到来，对于智能小车的设计制作是一次小小的尝试，从头到尾完整地参与一次课程设计使我受益良多，也让我感到了人工智能的魅力所在。相信随着科技的发展，在未来，人工智能会带给我们更多。