
人工智能小车



实验报告

组员：李雅娴 杨宁宁 张宇宁

第二次展示

上台展示 李雅娴

报告撰写 张宇宁

PPT 准备 杨宁宁

一、实验目的

人工智能是指由人制造出来的机器所表现出来的智能，被称为“第四次工业革命”，我们要实现的小车基础功能是：

- 1、能够自动前进，不原地打转；
- 2、遇到障碍物时，能够通过超声波规避障碍物；
- 3、能够使用电脑或者手机蓝牙遥控小车，使其完成前进、左右转弯、后退、停止、启动；
- 4、能够向电脑或者手机传拍摄的图像以及视频；

在实现了小车基础功能后，我们要实现的小车高级功能是：

- 1、通过装人工智能软件，使小车能够沿固定的轨道行驶；
- 2、实现声控人工智能小车
- 3、图像识别，选用了 TSL1401CL 线性 CCD 图像采集模块，该模块采用串行通信方式与主控 CPU 连接，智能车能根据采集到的图像分析前方路径及障碍而实现智能驾驶。
- 4、小车可以自动追踪目标，前方物体跑得慢小车减速，前方物体跑得快小车加速。若前方物体距离小车很近，则停车。

二、实验步骤

1、元件准备

新的蓝牙模块*1

新的超声波探测器*1

新的扩展板*1

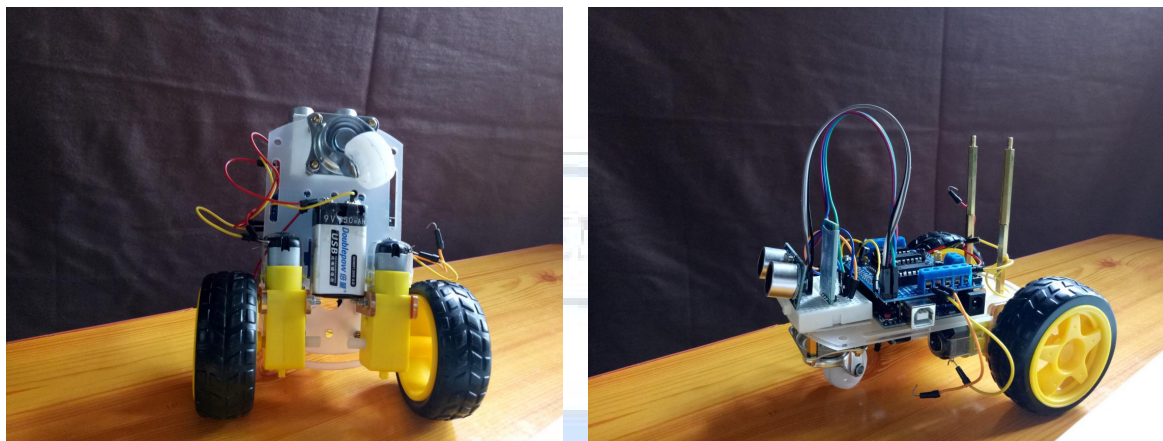
新的亚克力板车架*1（配有马达、轮胎）

Mp3 扩展板*1

发光二极管若干

2、小车拼装

(1) 拼装小车



(2) 调试小车

三、总结反思

1、经过多次实验，我们发现 **GND** 和 **+M** 口工作不稳定。比如：第一次调试小车时，**GND** 和 **+M** 接口不能正常工作。为了解决此问题，我们将两个马达的正负极各引出一根线分别与蓄电池的正负极相连。原理是：电池的正负极与扩展板的 **GND** 和 **+M** 口相连，扩展板 **M2** 和 **M3** 接口与马达的正负极相连，电池通过扩展板给马达供电。现在我们直接将电池正负极与两个马达正负极相连，实现电池直接给马达供电。

2、为了使小车显得长一点，我们将两个马达换了方向。调试小车时，我们发现小车倒着走。首先，我们对代码进行修改，将小车初始状态向前改为向后，发现并没有起作用。后来，我们尝试改变电池与马达的连线方式。定义马达上面的接口为 **a**，下面的接口为 **b**，我们开始将电池的正极接马达 **a** 口，负极接马达 **b** 口，现在我们调换一下顺序，将电池正极接马达 **b** 口，负极接马达 **a** 口。同时将代码修改回来，发现小车是向前走的，该问题解决。

3、在调试小车避障功能过程中，小车的右轮跑掉了两次，我们将它又安了上去。因为小车的轮子不能通过螺丝固定，小车撞上障碍物时发生碰撞，把轮子碰松了。同时我们发现，固定马达的螺丝钉会与车轮内侧发生摩擦。

4、刚开始小车没有动，经过多次尝试，我们发现新购买的扩展板的 M1、M2、M3、M4 接口工作不稳定。于是，我们使用二极管对它们进行测试，发现这些接口都不亮。我们又使用二极管对老的扩展板进行测试，发现它的 M1、M4 接口无法正常工作，M2、M3 接口正常工作。所以我们对接线和代码进行调整，将两个马达正负极分别与老扩展板的 M2、M3 接口相连，并将代码中 M1、M4 改为 M2、M4。通过不断尝试，小车可以前进了。

5、因为新扩展板 GND 和+M 接口无法正常工作，我们使用电池直接给马达供电。通过多次实验，发现有时候小车左侧马达的两极未与电池正负极相连时，右侧马达两极与电池正负极连接后，右轮并没有转动，反而左轮缓慢地转动。我们将左侧马达与电池相连，发现右轮依然没有转动，左轮急速地转动。但是，有时候左右轮正常转动，我们推测是因为连接马达同一极的两根导线接触而形成了回路，使右轮短路而无法正常运转。

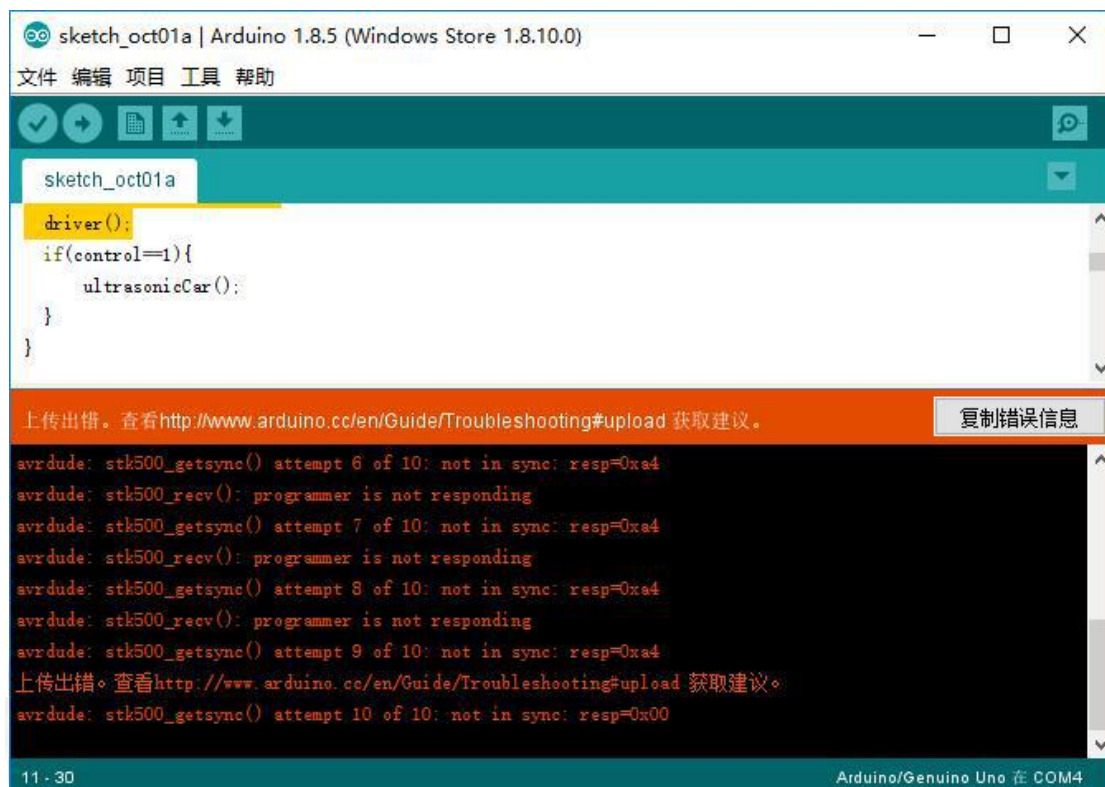
6、在测试小车避障功能时，我们发现小车无法识别障碍物。通过反复实验并与其他小组交流，我们发现面包板的前后两部分是不连通的。我们将杜邦线和传感器都插在了前半部分。

7、在测试小车避障功能时，为了方便控制小车，我们通过修改代码，将小车的速度从原来的 80 降至 50。多次实验中，我们发现小车有轻微停顿和小角度转弯，我们猜测是由于小车速度较低使转弯效果不明显。所以，我们将小车的速度提高至 60，并增加了转弯的时间。

8、在多次实验中，我们发现杜邦线的接触不良以及有些线是坏的，这对我们造成了困扰，我们通过换线、用胶带固定、焊接等方式进行改善。

9、在修改代码过程中，我们遇到了代码上传无法完成的情况，程序报错。我们对 UNO 板进行 reset 操作，重新上传发现还是不可以。后来我们分析后，如果 IDE 出现 avrdude: stk500_recv(): programmer is not responding 的提示的时候，可能是扩展板的数字 0 和数字 1 插着东西。如果数码管接到了数字 0 和数字 1 上，每次上传代码时，都需要要拔下来，因为数据交互借用了这两个脚。也就是说，如果你接线时用了串行，upload 时需要先拔下 0,1 口的接线，上传完成后再接回去。为了验证我们的想法，我们使用新的扩展板上传代码，新扩展板的 0 和 1

接口未进行焊接，发现代码正常上传，所以证实了我们的想法。我们发现自己的 UNO 开发板串口 RX/TX 是有其他连接或复用的（蓝牙连接），为了解决此问题，故直接摘下电机驱动扩展板，再次传输，成功。



10、实验过程中，我们发现传感器无法正常工作，经过多次分析和比对后，我们发现 rx 和 tx 接错了，正确的接法是 rx 接 tx，我们误将 rx 与 rx 连接，tx 与 tx 连接。重新连接后，传感器正常工作。

四、实验结果

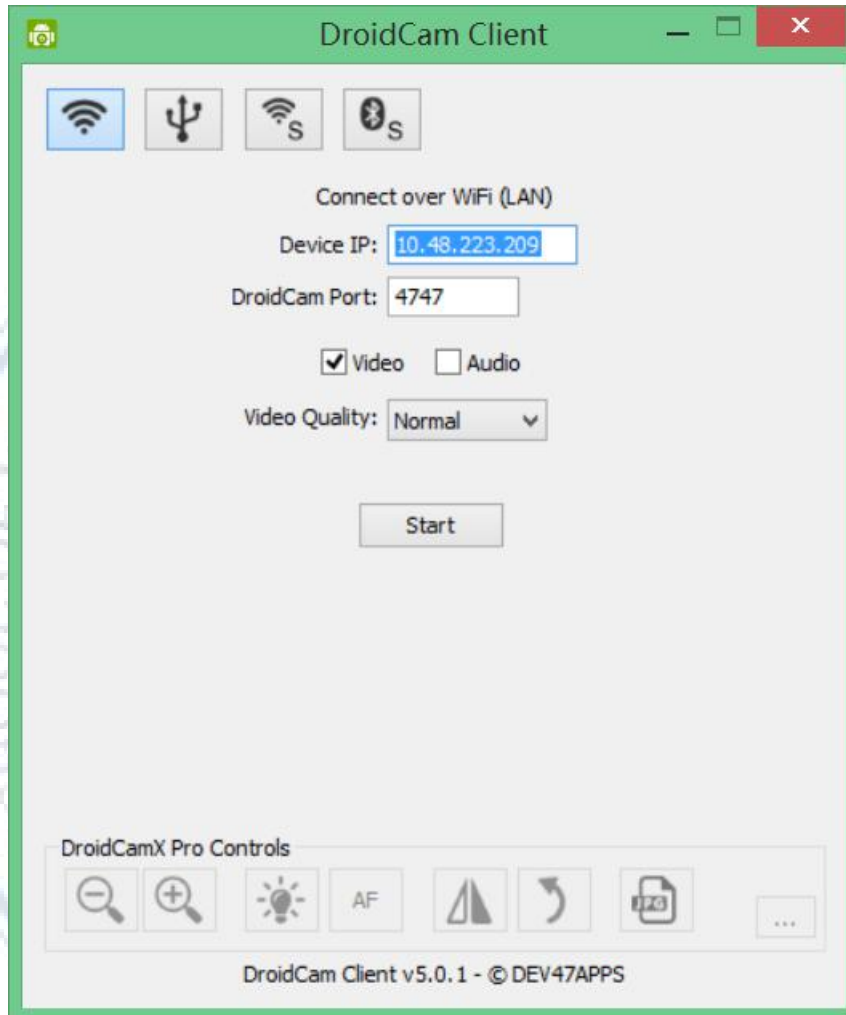
我们实现了以下功能：自动前进、超声波避障、蓝牙遥控（前进后退左右转停止等）、手机向电脑实时传图、自动追踪目标。完成了图像识别的初步准备工作，生成了测试集和训练集。

我们通过超声波测距，使小车随机转向 0.05 - 1 秒。人工控制中 1 表示前进，2 表示后退，3 表示向右转，4 表示向左转，5 表示停车；6 表示自动控制开始，7 表示自动控制结束，8 表示避障，9 表示自动追踪目标；+表示提升 RLRatio，使得右轮转速提升；-表示降低 RLRatio，使得右轮转速下降。

自动追踪目标时，最初小车会随机停等 0.1 - 0.5 秒，取得两个距离值。如果

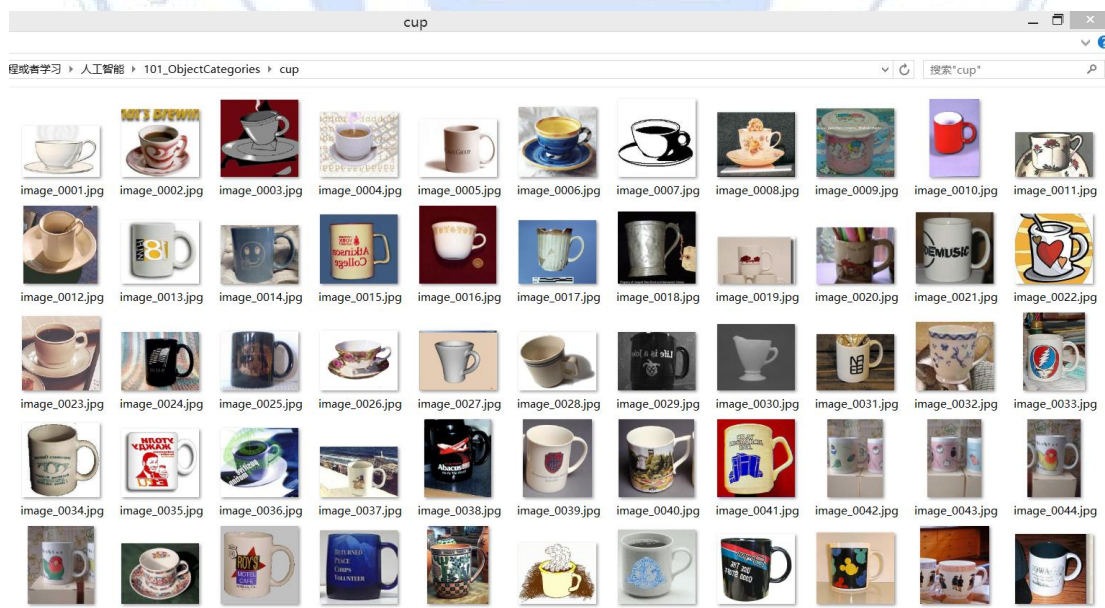
判断与前方物体距离太近（小于 20），停车；如果前方物体移动速度快（两次测距差值大），则小车加速；如果物体移动速度慢（两次测距差值小），则小车减速。如果是正常情形，小车跟着前方目标直行。

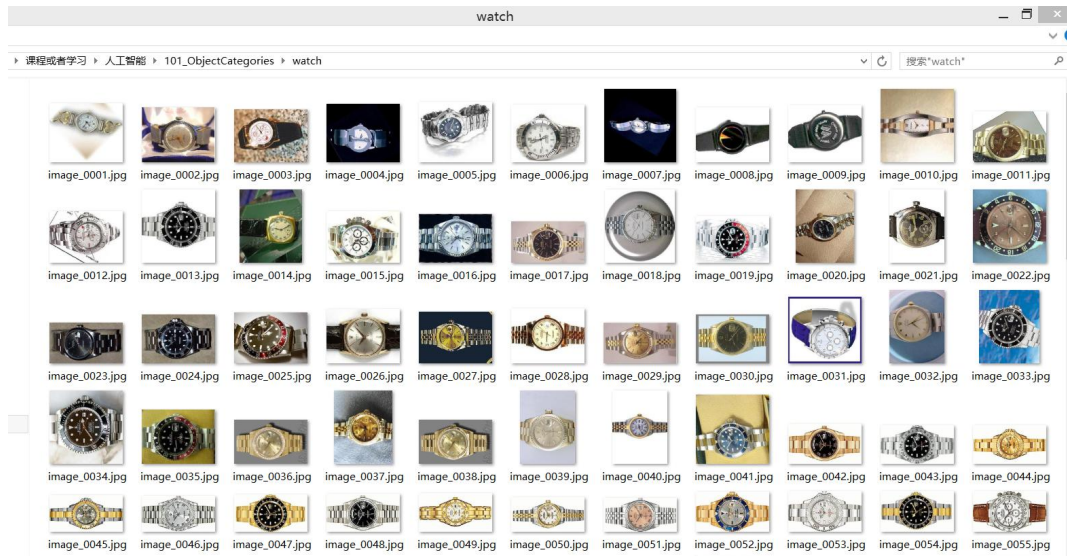
手机电脑实时视频传输截图：





图片分类测试集训练集部分展示如下：





以上数据将用于下一步深度学习的训练集。

五、代码

```
#include <AFMotor.h>

AF_DCMotor Rback_motor(4);
AF_DCMotor Lback_motor(1);

float RLRatio = 0.8;

int maxSpeed = 100;//最大速度

char getstr;//串口数据

int IntervalTime;//测距时间

int trig = 9;//sero1 保留端口

int echo = 10;//sero2 保留端口

int control=1;//0:人工控制;1:自动控制;2:追踪正前方物体

void setup() {

    // put your setup code here, to run once:
```



```
Serial.begin(9600);

Rback_motor.setSpeed(0);

Rback_motor.run(RELEASE);

Lback_motor.setSpeed(0);

Lback_motor.run(RELEASE);

delay(2000);

//超声波测距

pinMode(echo, INPUT);

pinMode(trig, OUTPUT);

}

void loop() {

    getstr = Serial.read();

    driver();

    if(control==1){

        ultrasonicCar();

    }

    else if(control==2){

        follow();

    }

}

void forward() {

    Rback_motor.run(BACKWARD);

    Lback_motor.run(BACKWARD);
```

```
Rback_motor.setSpeed(maxSpeed*RLRatio);

Lback_motor.setSpeed(maxSpeed);

}

void backward() {

    Rback_motor.run(FORWARD);

    Lback_motor.run(FORWARD);

    Rback_motor.setSpeed(maxSpeed*RLRatio);

    Lback_motor.setSpeed(maxSpeed);

}

void stopcar() {

    Rback_motor.setSpeed(0);

    Lback_motor.setSpeed(0);

    Rback_motor.run(RELEASE);

    Lback_motor.run(RELEASE);

}

void left(){

    Rback_motor.run(FORWARD);

    Lback_motor.run(BACKWARD);

    Rback_motor.setSpeed(maxSpeed*RLRatio);

    Lback_motor.setSpeed(maxSpeed);

}

void right(){

    Rback_motor.run(BACKWARD);
```

```
Lback_motor.run(FORWARD);

Rback_motor.setSpeed(maxSpeed*RLRatio);

Lback_motor.setSpeed(maxSpeed);

}

void driver() {

    if ((getstr == '-') && (RLRatio > 0.1)) {

        RLRatio -= 0.1;

        Serial.println("decrease right");

        Serial.println(RLRatio);

    }

    if (getstr == '+') {

        RLRatio += 0.1;

        Serial.println("increase right");

        Serial.println(RLRatio);

    }

    if (getstr == '5') {

        Serial.println("stopcar");

        stopcar();

        control=0;

    }

    if (getstr == '1') {

        Serial.println("forward");

        forward();

    }

}
```

```
control=0;

}

if (getstr == '2') {

    Serial.println("backward");

    backward();

    control=0;

}

if (getstr == '3') {

    Serial.println("right");

    right();

    control=0;

}

if (getstr == '4') {

    Serial.println("left");

    left();

    control=0;

}

if (getstr=='6'){

    control=1;

    Serial.println("Auto Control On");

}

if(getstr=='7'){

    control=0;
```

```
Serial.println("Auto Control Off");

}

if(getstr=='8'){

    float s;

    s = getdistance();

    Serial.println(s);

}

if(getstr=='9'){

    control=2;

    Serial.println("follow up....");

}

}

float getdistance() {

    digitalWrite(trig, 1);

    delayMicroseconds(15);

    digitalWrite(trig, 0);

    IntervalTime = pulseIn(echo, HIGH);

    return IntervalTime / 58.00;

}
```

```
void ultrasonicCar() {

    float s = 0;

    s = getdistance();
```



```
ultrasonicCar(s);  
}
```

```
void follow() {  
    //跟踪正前方物体  
  
    float s1 = 0;  
  
    float s2 = 0;  
  
    s1 = getdistance();  
  
    long n = random(100,500); //随机停等 0.1 - 0.5 秒  
    delay(n);  
    Rback_motor.setSpeed(0);  
    Lback_motor.setSpeed(0);  
    s2 = getdistance();  
    // 距离太近，停车  
    if(s2<20){  
        stopcar();  
    }  
  
    // 物体向前方移动，距离变化很大，加速  
    else if(s2-s1>=60){  
        if(maxSpeed<150){  
            maxSpeed+=5;  
        }  
  
        forward();  
    }  
}
```

```
}

// 物体向前方移动，距离变化很小，减速

else if(s2-s1<10){

    if(maxSpeed>=50){

        maxSpeed-=5;

    }

    forward();

}

// 正常情形，跟着前方目标直行就可以了

else{

    maxSpeed=100;

    forward();

}

}

void ultrasonicCar(float s) {

    if ((s < 40)&&(s>0)){//20cm 以内

        randomright();

    }

    else{

        forward();

    }

}
```

```
void randomright(){

    Rback_motor.run(BACKWARD);

    Lback_motor.run(FORWARD);

    Rback_motor.setSpeed(maxSpeed*RLRatio);

    Lback_motor.setSpeed(maxSpeed);

    long n = random(50,1000);//随机转向 0.05 - 1 秒

    delay(n);

    Rback_motor.setSpeed(0);

    Lback_motor.setSpeed(0);

}

void randomleft(){

    Rback_motor.run(FORWARD);

    Lback_motor.run(BACKWARD);

    Rback_motor.setSpeed(maxSpeed*RLRatio);

    Lback_motor.setSpeed(maxSpeed);

    long n = random(50,1000);//随机转向 0.05 - 1 秒

    delay(n);

    Rback_motor.setSpeed(0);

    Lback_motor.setSpeed(0);

}
```