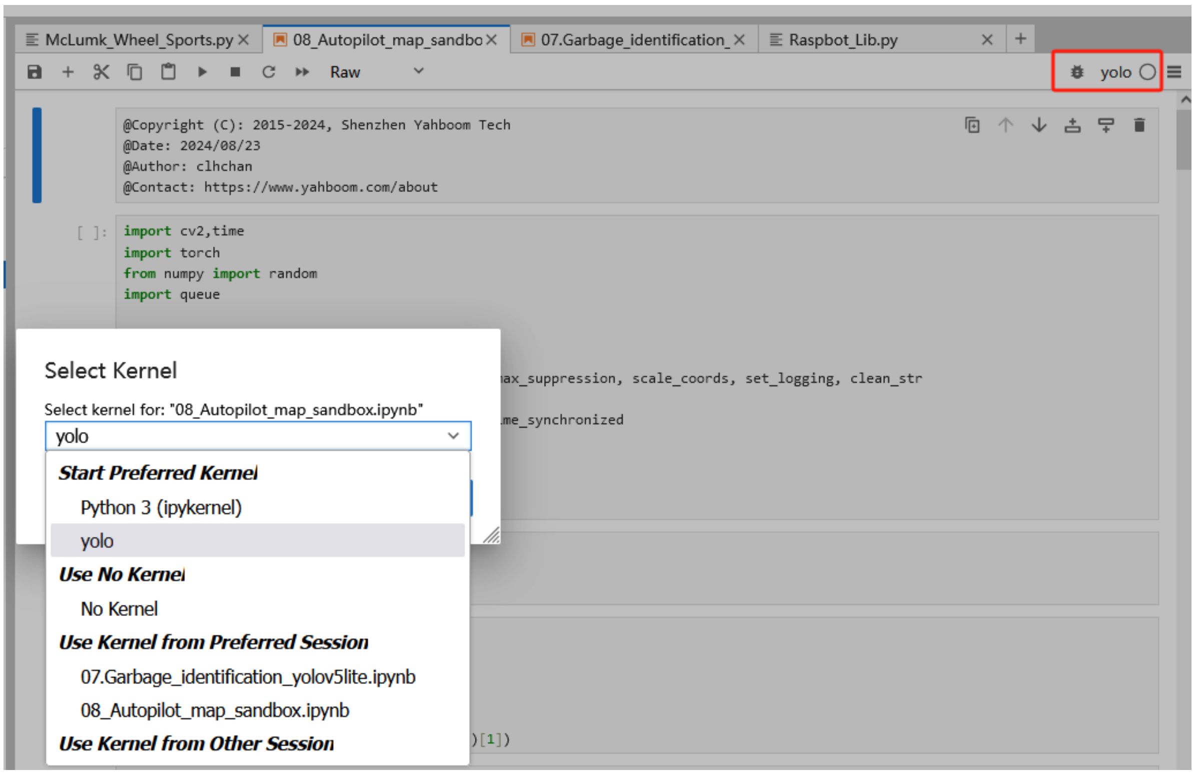
8. Track map Autopilot(purchased separately)

Must do before running

1. Close the startup program. For how to close it, please refer to "**00. Preparation before development**" in [**Basic course for small cars**]
2. Adjust the four-way patrol module. For how to adjust it, please refer to [**Basic tutorial for small cars**]"**07. Printing the status of the four-way patrol module**"
3. Because the camera is greatly affected by the ambient light, ensure that the ambient light is sufficient and uniform. Dim ambient light will affect road sign recognition.
4. Switch the jupyter leb kernel environment to **yolo environment** before starting



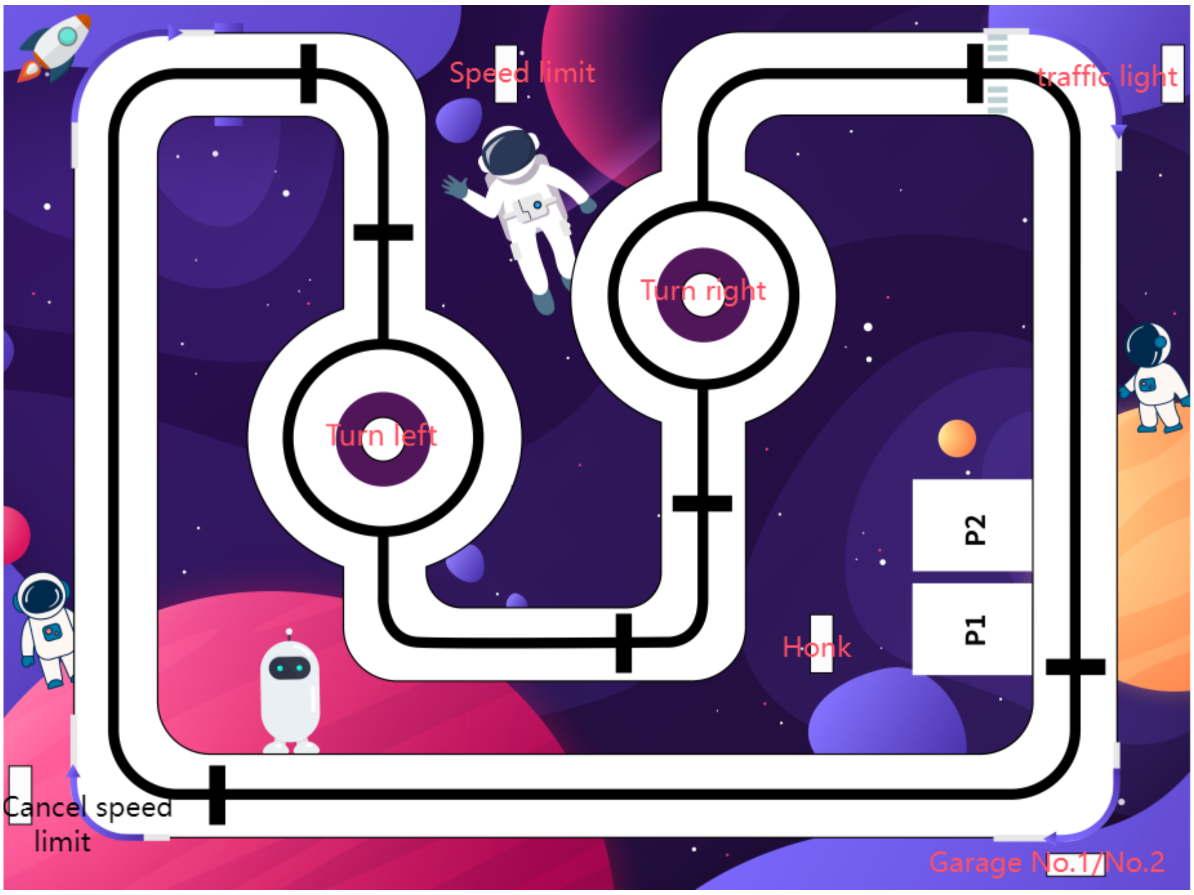
McLumk_Wheel_Sports.py
08_Autopilot_map_sandbo
07.Garbage_identification_
Raspbot_Lib.py
Raw
yolo
Select Kernel
Select kernel for: "08_Autopilot_map_sandbox.ipynb"
yolo
Start Preferred Kernel
Python 3 (ipykernel)
yolo
Use No Kernel
No Kernel
Use Kernel from Preferred Session
07.Garbage_identification_yolov5lite.ipynb
08_Autopilot_map_sandbox.ipynb
Use Kernel from Other Session

```
@Copyright (C): 2015-2024, Shenzhen Yahboom Tech
@Date: 2024/08/23
@Author: clhchan
@Contact: https://www.yahboom.com/about
```

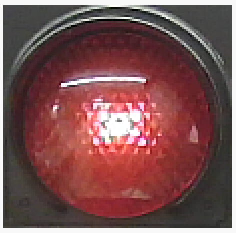
```
[ ]: import cv2,time
     import torch
     from numpy import random
     import queue
```

```
ax_suppression, scale_coords, set_logging, clean_str
me_synchronized
)[1])
```

Road sign placement instructions



Road sign indication function description

Garage No. 1	Garage No. 2	Honk	Speed Limit
			
Cancel speed limit	Turn left	Turn right	traffic light

Explanation of road sign functions

Garage No. 1: The car reverses into garage No. 1

Garage No. 2: The car reverses into garage No. 2

Honk: The car buzzer sounds once

Speed limit: The car slows down

Cancel speed limit: The car speeds up

Turn left: The car executes the left turn command

Turn right: The car executes the right turn command

Traffic light: The car stops when the red light is on, and the car moves forward when the red light is off

Key program analysis

Image processing function

```
def detect(weights='weights/best.pt', source='0', img_size=320, conf_thres=0.45,
iou_thres=0.35, device=''):
    #Default: best.pt yolov5 model
    #best1.pt yolov5lite model
    global classes
    # Initialize
    set_logging()
    device = select_device(device)
    half = device.type != 'cpu'  # half precision only supported on CUDA

    # Load model
    model = attempt_load(weights, map_location=device)  # load FP32 model
    stride = int(model.stride.max())  # model stride
    imgsz = check_img_size(img_size, s=stride)  # check img_size
    if half:
        model.half()  # to FP16

    # Set Dataloader
    dataset = LoadStreams(source, img_size=imgsz, stride=stride)

    # Get names and colors
    names = model.module.names if hasattr(model, 'module') else model.names
    colors = [[random.randint(0, 255) for _ in range(3)] for _ in names]

    frame_count = 0
    start_time = time.time()

    # Run inference
    if device.type != 'cpu':
        model(torch.zeros(1, 3, imgsz,
imgsz).to(device).type_as(next(model.parameters())))  # run once
    for path, img, im0s, vid_cap in dataset:
        img = torch.from_numpy(img).to(device)
        img = img.half() if half else img.float()  # uint8 to fp16/32
        img /= 255.0  # 0 - 255 to 0.0 - 1.0
        if img.ndimension() == 3:
            img = img.unsqueeze(0)

        # Calculate FPS before processing the frame
        frame_count += 1
        elapsed_time = time.time() - start_time
        fps = frame_count / elapsed_time
        # Display FPS on the frame
        cv2.putText(im0s[0], f"FPS: {fps:.2f}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

        # Inference
        pred = model(img, augment=False)[0]

        # Apply NMS
        pred = non_max_suppression(pred, conf_thres, iou_thres)

        # Process detections
        for i, det in enumerate(pred):  # detections per image
            if det is not None and len(det):
                # Rescale boxes from img_size to im0 size
                det[:, :4] = scale_coords(img.shape[2:], det[:, :4],
im0s[i].shape).round()

                # Print results
                for *xyxy, conf, cls in reversed(det):
                    label = f'{names[int(cls)]} {conf:.2f}'
                    classes = f'{names[int(cls)]}'
                    #classes = f'{names[int(cls)]}'

                    #print(classes)  # 打印识别的标签和置信度 Print the recognized
labels and confidence
                    plot_one_box(xyxy, im0s[i], label=label,
color=colors[int(cls)], line_thickness=3)

        # Stream results
        #cv2.imshow('Video0', im0s[0])
        global image_widget
        image_widget.value = bgr8_to_jpeg(im0s[0])

        # if cv2.waitKey(1) == ord('q'):  # Press 'q' to quit
        #     break

    #cv2.destroyAllWindows()
```

Car automatic driving function

```
def tracking_control():
    global classes,speed,flag,leftflag,rightflag \
    ,pack1flag,pack2flag,stopflag,classestemp,rightrunflag,leftrunflag,classflag
    while True:  # 连续检测 Continuous detection
        # 从I2C读取巡线传感器数据 Read line sensor data from I2C
        track_data = bot.read_data_array(0x0a, 1)
        track = int(track_data[0])

        # 解析巡线传感器的状态 Analyze the status of the line patrol sensor
        x1 = (track >> 3) & 0x01
        x2 = (track >> 2) & 0x01
        x3 = (track >> 1) & 0x01
        x4 = track & 0x01
        """
        x2 x1 x3 x4
        |  |  |  |
        L1 L2 R1 R2
        """
        lineL1=x2
        lineL2=x1
        lineR1=x3
        lineR2=x4

        if classes != None:
            if(classes == 'stop'):
                classestemp=classes
                classflag=2
                flag=0
                if(lineL2 == 1 or lineR1 == 1):
                    move_backward(5)
                    time.sleep(0.1)
                    stop()
            elif(classes == 'run'):
                classestemp=classes
                classflag=0
                '''-----根据实际情况调整速度-----
                -Adjust speed according to actual situation-'''
                speed=5
                '''-----根据实际情况调整速度-----
                -Adjust speed according to actual situation-'''
                flag=1
                classes=None
        #flag=0
        if flag:
            #if (lineL1 == 0 and lineL2 == 0 and lineR1 == 0 and lineR2 == 0):
            if (lineL1 == 0 and lineL2 == 0 and lineR1 == 0 and lineR2 == 0) or
(lineL1 == 0 and (lineR2 == 0 or lineR1 == 0)) or ( (lineL2 == 0 or lineL1 == 0)
and lineR2 == 0):
            #if (lineL1 == 0 or lineR2 == 0):
                #根据识别结果调整行为 Adjust behavior based on recognition results
                if classes != None and classes !='run' and classes !='stop':
                    stop()
                    time.sleep(0.025)
                    bot.Ctrl_BEEP_Switch(1)
                    time.sleep(0.1)
                    bot.Ctrl_BEEP_Switch(0)
                    if classes=='right' :# 路标右转动作 Road sign right turn action
                        #print("right!!!!!!!!!!!!")
                        bot.Ctrl_Servo(2, 5)
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                        move_forward(int(speed)) # 前进 go ahead
                        time.sleep(0.75)
                        rotate_right(int(speed*1.5))# 右转 Turn right
                        time.sleep(0.5)
                        bot.Ctrl_Servo(2, 25)
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                        rightrunflag=0
                        leftflag=0
                        classflag+=1
                    elif classes=='left' :# 路标左转动作 Road sign left turn action
                        #print("left!!!!!!!!!!!!!")
                        bot.Ctrl_Servo(2, 5)
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                        move_forward(int(speed))
                        time.sleep(0.75)
                        rotate_left(int(speed*2))
                        time.sleep(0.5)
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                        bot.Ctrl_Servo(2, 25)
                        leftrunflag=0
                        leftflag=1
                        classflag+=1
                    elif(classes == 'freeSpeed'):# 解除限速 Lift speed limit
                        speed=15#30
                    elif(classes == 'limitSpeed'):# 限制速度 Speed Limit
                        rightrunflag=1
                        leftrunflag=1
                        speed=5#20
                    elif(classes == 'beep'):# 鸣笛三声 Three blasts
                        rightrunflag=1
                        leftrunflag=1
                        bot.Ctrl_BEEP_Switch(1)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(0)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(1)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(0)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(1)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(0)
                        #print("beep")
                    elif(classes == 'one' and pack1flag ==0):# 倒入1库 Pour into
1 warehouse
                        bot.Ctrl_BEEP_Switch(1)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(0)
                        pack1flag=1
                        pack2flag=0
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                        move_backward(speed)
                        time.sleep(0.3)
```

```
                        rotate_left(int(speed*4))
                        time.sleep(0.8)
                        move_backward(speed)
                        time.sleep(0.4)
                        stop()
                        time.sleep(2)
                        move_forward(speed)
                        time.sleep(0.75)
                        rotate_right(int(speed*4))
                        time.sleep(0.8)
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''

                    elif(classes == 'two' and pack2flag ==0):# 倒入2库 Pour into
2 warehouses
                        bot.Ctrl_BEEP_Switch(1)
                        time.sleep(0.1)
                        bot.Ctrl_BEEP_Switch(0)
                        pack1flag=0
                        pack2flag=1
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                        move_backward(speed)
                        time.sleep(0.9)
                        rotate_left(int(speed*4))
                        time.sleep(0.8)
                        move_backward(speed)
                        time.sleep(0.4)
                        stop()
                        time.sleep(2)
                        move_forward(speed)
                        time.sleep(0.75)
                        rotate_right(int(speed*4))
                        time.sleep(0.8)
                        '''-----根据实际情况调整速度，延时-----
                        -Adjust speed and delay according to actual conditions-
'''
                classes=None
                stopflag=0
            if leftflag:# 左转优先 Left turn priority
                if(lineL1 == 0 and lineL2 == 0 and lineR1 == 0 and lineR2 == 0):
                    rightrunflag=1
                    leftrunflag=1
                    move_forward(int(speed))
                    stopflag=0
                elif lineL1 == 0 and (lineR2 == 0 or lineR1 == 0):  # 左锐角或左大
弯 Left sharp angle or left sharp bend
                    #print("3")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_left(int(speed*1.5))
                    time.sleep(0.15)
                elif( (lineL2 == 0 or lineL1 == 0) and lineR2 == 0):#右锐角：右大
弯,0表示检测到黑线 Right acute angle: right big bend, 0 means black line is
detected
                    #print("2")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_right(speed)
                    time.sleep(0.05)
                elif lineL1 == 0:  # 左最外侧检测 Left outermost detection
                    #print("4")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_left(speed)
                    time.sleep(0.02)
                elif lineR2 == 0:  # 右最外侧检测 Right outermost detection
                    #print("5")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_right((int(speed)))
                    time.sleep(0.05)
                elif lineL2 == 0 and lineR1 == 1:  # 中间黑线上的传感器微调车左转 The
sensor on the middle black line fine-tunes the car to turn left
                    if leftrunflag==0:
                        classes=None
                    if rightrunflag==0:
                        classes=None
                    #print("6")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_left(int(speed))
                elif lineL2 == 1 and lineR1 == 0:  # 中间黑线上的传感器微调车右转 The
sensor on the middle black line fine-tunes the car to turn right
                    if leftrunflag==0:
                        classes=None
                    if rightrunflag==0:
                        classes=None
                    #print("7")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_right(int(speed))
                elif lineL2 == 0 and lineR1 == 0:  # 都是黑色，加速前进 All black,
speed up
                    if(classflag)>1:
                        '''-----根据实际情况调整速度-----
                        -Adjust speed according to actual situation-'''
                        move_forward(5)
                        '''-----根据实际情况调整速度-----
                        -Adjust speed according to actual situation-'''
                    else:move_forward(speed)
            else:# 默认右转优先巡线 # Default right turn priority patrol
                if(lineL1 == 0 and lineL2 == 0 and lineR1 == 0 and lineR2 == 0):
                    rightrunflag=1
                    leftrunflag=1
                    move_forward(int(speed))
                    stopflag=0
                elif( (lineL2 == 0 or lineL1 == 0) and lineR2 == 0):#右锐角：右大
弯,0表示检测到黑线 Right acute angle: right big bend, 0 means black line is
detected
                    #print("2")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_right((int(speed*2)))
                    time.sleep(0.05)
                elif lineL1 == 0 and (lineR2 == 0 or lineR1 == 0):  # 左锐角或左大
弯 Left sharp angle or left sharp bend
                    #print("3")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_left(int(speed*1.5))
                    time.sleep(0.15)
                elif lineR2 == 0:  # 右最外侧检测 Right outermost detection
                    #print("5")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_right((int(speed)))
                    time.sleep(0.05)
                elif lineL1 == 0:  # 左最外侧检测 Left outermost detection
                    #print("4")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_left(int(speed*1.5))
                    time.sleep(0.05)
                elif lineL2 == 0 and lineR1 == 1:  # 中间黑线上的传感器微调车左转 The
sensor on the middle black line fine-tunes the car to turn left
                    if leftrunflag==0:
                        classes=None
                    if rightrunflag==0:
                        classes=None
                    #print("6")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_left(int(speed))
                elif lineL2 == 1 and lineR1 == 0:  # 中间黑线上的传感器微调车右转 The
sensor on the middle black line fine-tunes the car to turn right
                    if leftrunflag==0:
                        classes=None
                    if rightrunflag==0:
                        classes=None
                    #print("7")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    rotate_right(int(speed)) #右转
                elif lineL2 == 0 and lineR1 == 0:  # 都是黑色, 加速前进 All black,
speed up
                    #print("8")
                    #print(lineL1,lineL2,lineR1,lineR2)
                    if(classflag)>1:
                        '''-----根据实际情况调整速度-----
                        -Adjust speed according to actual situation-'''
                        move_forward(5)
                        '''-----根据实际情况调整速度-----
                        -Adjust speed according to actual situation-'''
                    else:move_forward(speed)
            # 等待一段时间再进行下一次检测 Wait for a while before the next test
            #print("\n")
            time.sleep(0.01)
        elif flag==0:
            stop()
```

Notes

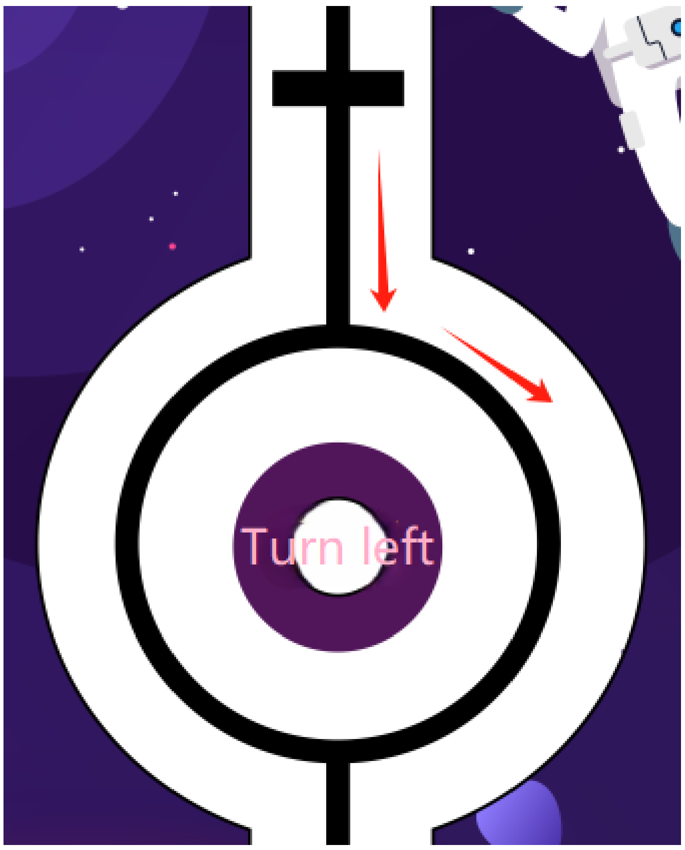
1. Due to the difference in the hardware of the motor itself, the speed is different and the effect is also different. This program needs to be fine-tuned according to the personal use environment, not the best effect
2. The battery must be fully charged before running to ensure that the speed is reached, otherwise the line patrol cannot be performed normally
3. Road sign recognition provides two models, which can be tried according to the actual situation
4. Under normal circumstances, the motor is relatively smooth. If the motor is stuck, please consider charging
5. The line patrol map must be kept clean and there should not be too many stains, otherwise it will affect normal use
6. The environment should not be too bright or too dark when patrolling the line, normal lighting is enough

How to adjust the movement effect of the car

Example 1

Recognize the road sign and turn left. There are two actions: **forward**, **turn left**

The same applies to turning right.



```
elif classes=='left' :# 路标左转动作 Road sign left turn action
...
    '''-----根据实际情况调整速度，延时-----
    -Adjust speed and delay according to actual conditions-'''
    move_forward(int(speed))# 前进 go ahead
    time.sleep(0.75)
    rotate_left(int(speed*2))# 左转 Turn left
    time.sleep(0.5)
    '''-----根据实际情况调整速度，延时-----
    -Adjust speed and delay according to actual conditions-'''
...
```

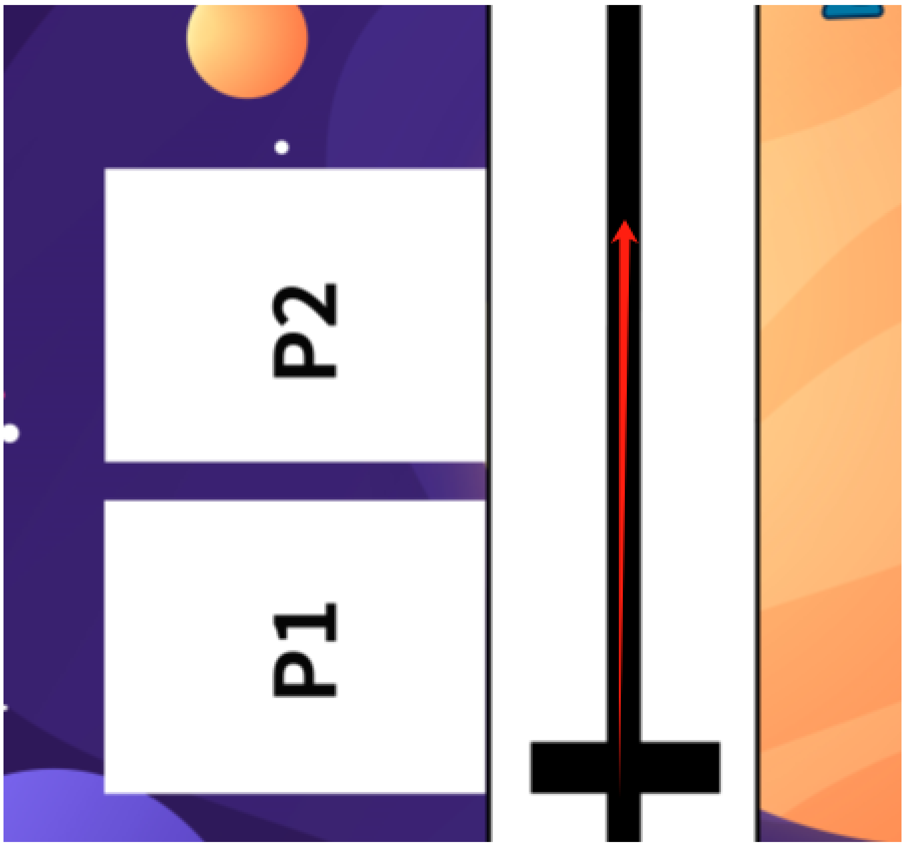
If the action does not reach the position, you need to adjust the speed and action delay time.sleep here

Example 2

The recognition of the reverse parking sign is four actions, **entering** (backward, left turn, backward), **exiting** (forward, right turn)

(Reversing the same parking lot cannot be repeated, you can switch the recognition parking lot sign and eliminate the sign)

The first reverse of the reverse parking lot 2 is more than the first reverse parking lot 1, and the same applies to the reverse parking lot 2



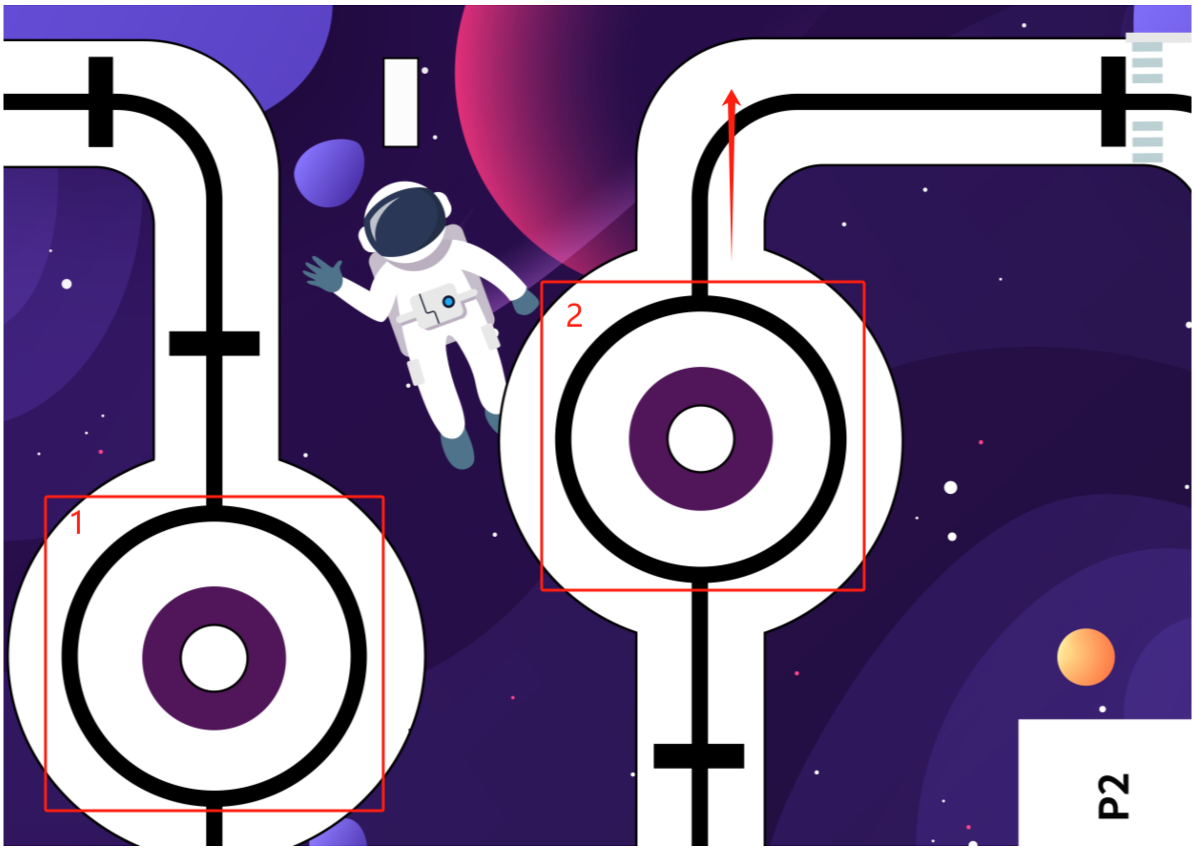
```
elif(classes == 'one' and pack1flag ==0):# 倒入1库 Pour into 1 warehouse
...
    '''-----根据实际情况调整速度，延时-----
    -Adjust speed and delay according to actual conditions-'''
    move_backward(speed)# 后退 Back
    time.sleep(0.3)
    rotate_left(int(speed*4))# 左旋 Left Rotation
    time.sleep(0.8)
    move_backward(speed)# 后退 Back
    time.sleep(0.4)
    stop()# 停止
    time.sleep(2)
    move_forward(speed)# 前进 go ahead
    time.sleep(0.75)
    rotate_right(int(speed*4))# 右旋 Right Rotation
    time.sleep(0.8)
    '''-----根据实际情况调整速度，延时-----
    -Adjust speed and delay according to actual conditions-'''
...
```

If the action fails to reach the position, the speed and action delay time.sleep need to be adjusted

Example 3

Recognition model effect reason, entering the section of **traffic light**, the car needs to drive very slowly to leave space and time for the car to recognize the red light

The program logic is to recognize the left/right turn road sign, and when the number of times is greater than one, the flag classflag is set to 1, and the speed limit is imposed on straight driving.



```
elif lineL2 == 0 and lineR1 == 0:  # 都是黑色，加速前进 All black, speed up
    if(classflag)>1:
        '''-----根据实际情况调整速度-----
        -Adjust speed according to actual situation-'''
        move_forward(5)
        '''-----根据实际情况调整速度-----
        -Adjust speed according to actual situation-'''
    else:move_forward(speed)
```

If the action does not reach the position, you need to adjust the speed and action delay time.sleep