3. Voice control of the car line patrol automatic driving

This course needs to be combined with the Rosmaster-X3 car hardware, and only code analysis is done here. First, let's look at the built-in voice command,

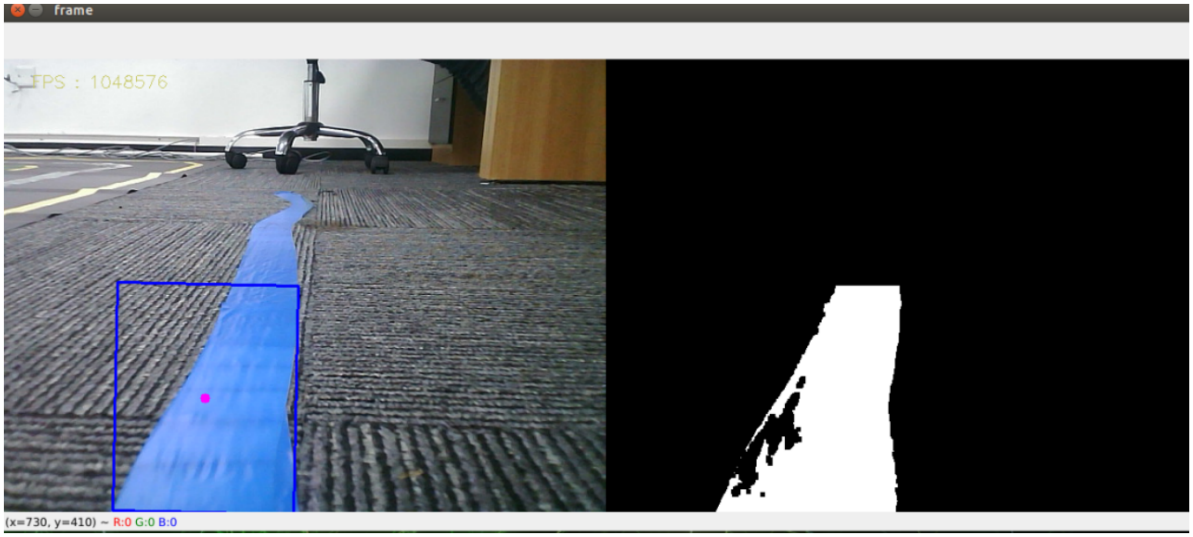
Function words	Speech recognition module results	Voice broadcast content
Turn off the line following function	22	OK, the line following function has been turned off
Turn on the red line patrol function	23	OK, the red line patrol function has been turned on
Turn on the green line patrol function	24	OK, the green line patrol function has been turned on
Turn on the blue line patrol function	25	OK, the blue line patrol function has been turned on
Turn on the yellow line patrol function	26	OK, the yellow line patrol function has been turned on

1. Program startup

Terminal input,

```
#Start the car chassis
ros2 run yahboomcar_bringup Mcnamu_driver_X3
#Start the voice patrol program
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_follow_line_a1_X3
#Start the handle control node
ros2 run yahboomcar_ctrl yahboom_joy_X3
ros2 run joy joy_node
```

Bend the camera of the car downward so that it can see the line, then wake up the module first ("Hi Yahboom"), and after getting a response, take patrolling the blue line as an example, you can say "patrol the blue line" to it,



Press the R2 button of the controller to start patrolling the line.

2. Core code

Code path:~/driver_ws/src/yahboomcar_voice_ctrl/yahboomcar_voice_ctrl/Voice_Ctrl_follow_line_a1_X3.py

```
def process(self, rgb_img, action):
    binary = []
    rgb_img = cv.resize(rgb_img, (640, 480))
    if self.img_flip == True: rgb_img = cv.flip(rgb_img, 1)
    #这里开始接收语音命令以及发布指令和加载hsv的值 Here we start receiving voice
commands, issuing instructions and loading hsv values
    self.command_result = self.spe.speech_read()
    self.spe.void_write(self.command_result)
    if self.command_result == 23:
    self.model = "color_follow_line"
    print("red follow line")
    #红色HSV Red HSV
    self.hsv_range = [(0, 84, 131), (180, 253, 255)]
#以下部分就是把hsv的值传进去，图像处理，得到一个self.circle的值，最后传入self.execute的函
数，计算速度 The following part is to pass the hsv value in, process the image, get
a value of self.circle, and finally pass it to the self.execute function to
calculate the speed
if self.model == "color_follow_line":
rgb_img, binary, self.circle = self.color.line_follow(rgb_img,self.hsv_range)
if len(self.circle) != 0:
threading.Thread(target=self.execute, args=
(self.circle[0],self.circle[2])).start()
```