3、 Voice controlled automatic driving for car line inspection

This course needs to be combined with the hardware of the Rosmaster-X3 car, and only code analysis will be done here. Firstly, let's take a look at the built-in voice commands，

Directive word	Speech recognition results
Close tracking mode	22
track red line	23
track green line	24
track blue line	25
track yellow line	26

1、 Enter the startup program

Terminal input，

```
#Start the trolley chassis
ros2 run yahboomcar_bringup Mcnamu_driver_X3
#Enable line patrol function
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_follow_line_a1_X3
#Start handle control node
ros2 run yahboomcar_ctrl yahboom_joy_X3
ros2 run joy joy_node
```

Break down the camera of the car so that it can see the line, and then wake up the module ("Hi Yahboom") first. After receiving a response, taking the blue line patrol as an example, you can say "track blue line" to it,，
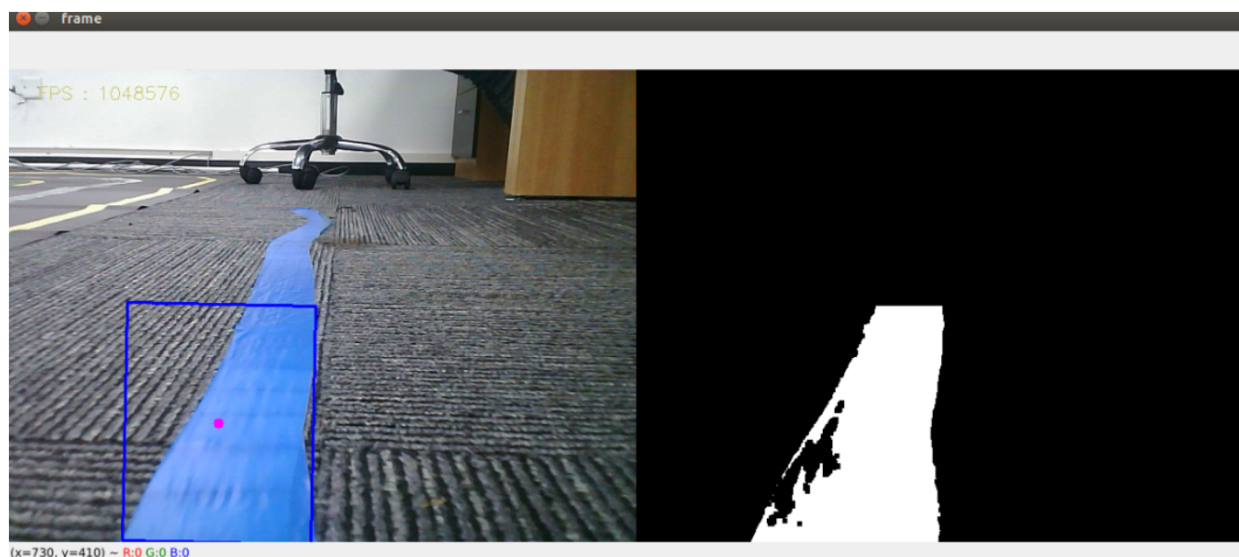
Press the R2 key on the handle to start line patrol。

2、Core code

code path:

~/driver_ws/src/yahboomcar_voice_ctrl/yahboomcar_voice_ctrl/Voice_Ctrl_follow_line_a1_X3.py

```python
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)
    #Start receiving voice commands, issuing commands, and loading hsv values here
    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")
    #Red HSV
    self.hsv_range = [(0, 84, 131), (180, 253, 255)]
#The following section is to pass in the value of hsv, process the image, obtain a
self. circle value, and finally pass in 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()
```