Voice Controlled Car Tracking

Experimental Objective

Understand and learn to use fixed basic semantics to enable the car to track objects based on semantics.

Experimental Procedure and Results

1. First, enter the following command in the terminal:

```
cd /home/pi/project_demo/10.Basic_voice_control/5.Speech_Track_color_Face/
python3 Speech_Auto_Track.py
```

2. After entering this interface, wake up the car using the wake-up phrase: Hi, Yahboom in English.

```
pi@yahboom:~/project_demo/10.Basic_voice_control/5.Speech_Track_color_Face $ python3 Speech_Auto_
Track.py
Speech Serial Opened! Baudrate=115200
```

3. After successfully waking up, the car responds: "Hi, I'm here" in English.
4. Then, control the car using fixed commands to track the corresponding object.

Fixed Statement Table

Wake-up Statement (International Users)	Operational Performance	Car Answer (English Version)
Face Following	Car Tracking Face	OK, It Has Been Stopped
yellow following	Car will track the yellow object	OK, I found the yellow
red following	Car will track the red object	OK, I found the red
green following	Car will track the green object	OK, I found the green
blue following	Car will track the blue object	OK, I found this color
stop following	Car will stop tracking	OK, It Has Been Stopped

Main source code analysis

```
if __name__ == "__main__":
    tracker = ColorLineTracker()
    mySpeech = Speech()

    try:
        while True:
            time.sleep(0.2)

            num = mySpeech.speech_read()
            if num !=999 :
                #print(num)
                if num == 0:
                    mySpeech.void_write(num)
                if num == 71:
                    mySpeech.void_write(num)
                    print('Face')
                    tracker.start_Face()
                elif num == 72:
                    mySpeech.void_write(num)
                    print('yellow')
                    tracker.start('yellow')
                elif num == 73:
                    mySpeech.void_write(num)
                    print('red')
                    tracker.start('red')

                elif num == 74:
                    mySpeech.void_write(num)
                    print('green')
                    tracker.start('green')

                elif num == 75:
                    mySpeech.void_write(num)
                    print('blue')
                    tracker.start('blue')

                elif num == 76:
                    mySpeech.void_write(num)
                    print('stop!')
                    tracker.stop()

    except KeyboardInterrupt:
        tracker.stop()
        print('Speech Track end!')
```

speech_read: Identifies the color of the patrol line based on fixed semantics.
ColorLineTracker: Starts tracking. Can track red, yellow, blue, green, and faces.