Gesture control of the car to make shapes

Note: The virtual machine, ROS-wifi image transmission module and ESP32 communication board ROS_DOMAIN_ID need to be consistent, and both must be set to 20. You can view [ESP32 communication board parameter configuration] to set the ESP32 communication board ROS_DOMAIN_ID, and view the tutorial [Connecting MicroROS Agent] to determine whether the ID is consistent.

Before running the experiment, please make sure that the microros balance car and ROS-wifi image transmission module have correctly enabled the agent on the virtual machine (linux with humbleROS2 system)

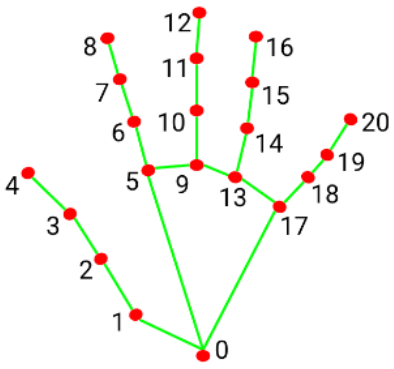
1. Program function description

After the function is enabled, the camera captures the image and recognizes the relevant gestures to control the movement of the car.

Gesture number "5"	Car stops
Gesture "yes"	Car moves in a square
Gesture "ok"	Car turns in a circle
Gesture "rock" (index finger and pinky finger are straight, and the others are bent)	Car moves in an S shape
Gesture contempt (clenched fist, thumb extended, thumb facing down)	Car moves forward and then backward

Here, after each gesture is completed, it will return to the initialization position and beep, waiting for the next gesture recognition.

MediaPipe Hands infers the 3D coordinates of 21 hand value joints from a frame.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

2. Program code reference path

The source code of this function is located at,

```
/home/yahboom/yahboomcar_ws/src/yahboom_esp32ai_car/yahboom_esp32ai_car/FingerCtrl.py
```

3. Program startup

3.1. Start command

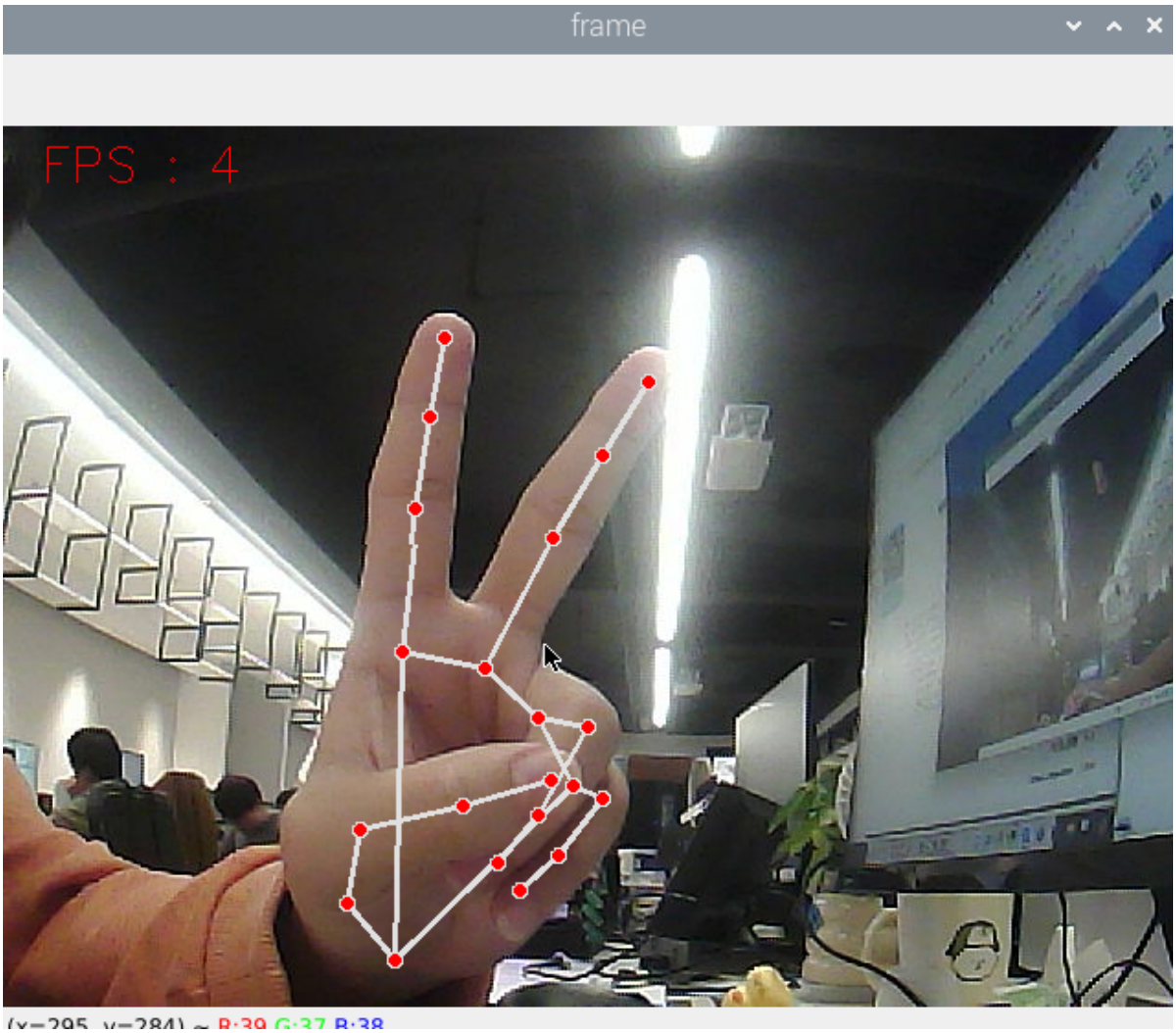
Input in terminal,

```
#Start chassis driver
ros2 launch yahboomcar_bringup yahboomcar_bringup_launch.py
```

```
#Start gesture control car action group program
ros2 run yahboom_esp32ai_car FingerCtrl
```

If the camera image is inverted, you need to read **3. Camera Image Correction (Select View)** document to correct it yourself. This experiment will not be described.

Turn on this function, then put your hand in front of the camera, the screen will draw the shape of the finger, and after the program recognizes the gesture, it will send the speed to the chassis, thereby controlling the movement of the car.



4. Core code

4.1. FingerCtrl.py

```
frame, lmList, bbox = self.hand_detector.findHands(frame) #Detect palm
fingers = self.hand_detector.fingersUp(lmList) #Get finger coordinates
gesture = self.hand_detector.get_gesture(lmList) #Get gesture
#For the specific implementation process of the above three functions, please
refer to the content in media_library.py
```

The implementation process here is also very simple. The main function opens the camera to obtain data and then passes it to the process function. It performs "detect palm" -> "get finger coordinates" -> "get gesture" in sequence, and then determines the action to be performed based on the gesture results.

4.2. Flowchart