

Gesture control car action group

Note: The VM and ROS-wifi image transfer module must be consistent with the microROS control board ROS_DOMAIN_ID and set the value to 20. You can check [MicroROS control board Parameter configuration] to set the microROS control board ROS_DOMAIN_ID. Check the tutorial Connecting to MicroROS Agents to see if the ids are the same.

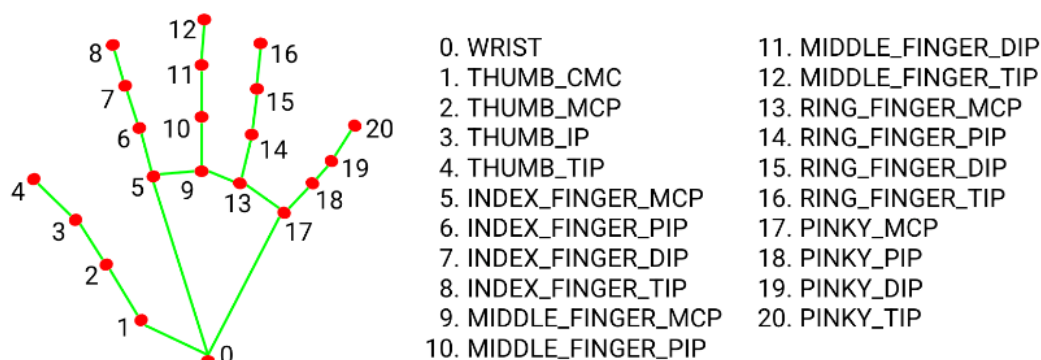
1、Program function specification

When the function is turned on, the camera captures images and recognizes relevant gestures to control the car's movement.

Gesture digit "5"	Car Stop
Gesture "yes"	Trolley square
Gesture "ok"	Trolley wheel
Gesture "rock" (Keep your pinky finger straight. Bend the rest)	The car takes the s shape
Gesture contempt (Clench your fists and extend your thumbs, thumbs down)	The car goes forward and then back

Here, when each gesture is finished, it will return to the initial position, and drop a sound, waiting for the next gesture recognition.

MediaPipe Hands The 3D coordinates of 21 hand-valued joints are inferred from a single frame.



2、Program code reference path

After entering the docker container, the location of the functional source code is located in,

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

3、 Program initiation

3.1、 Start command

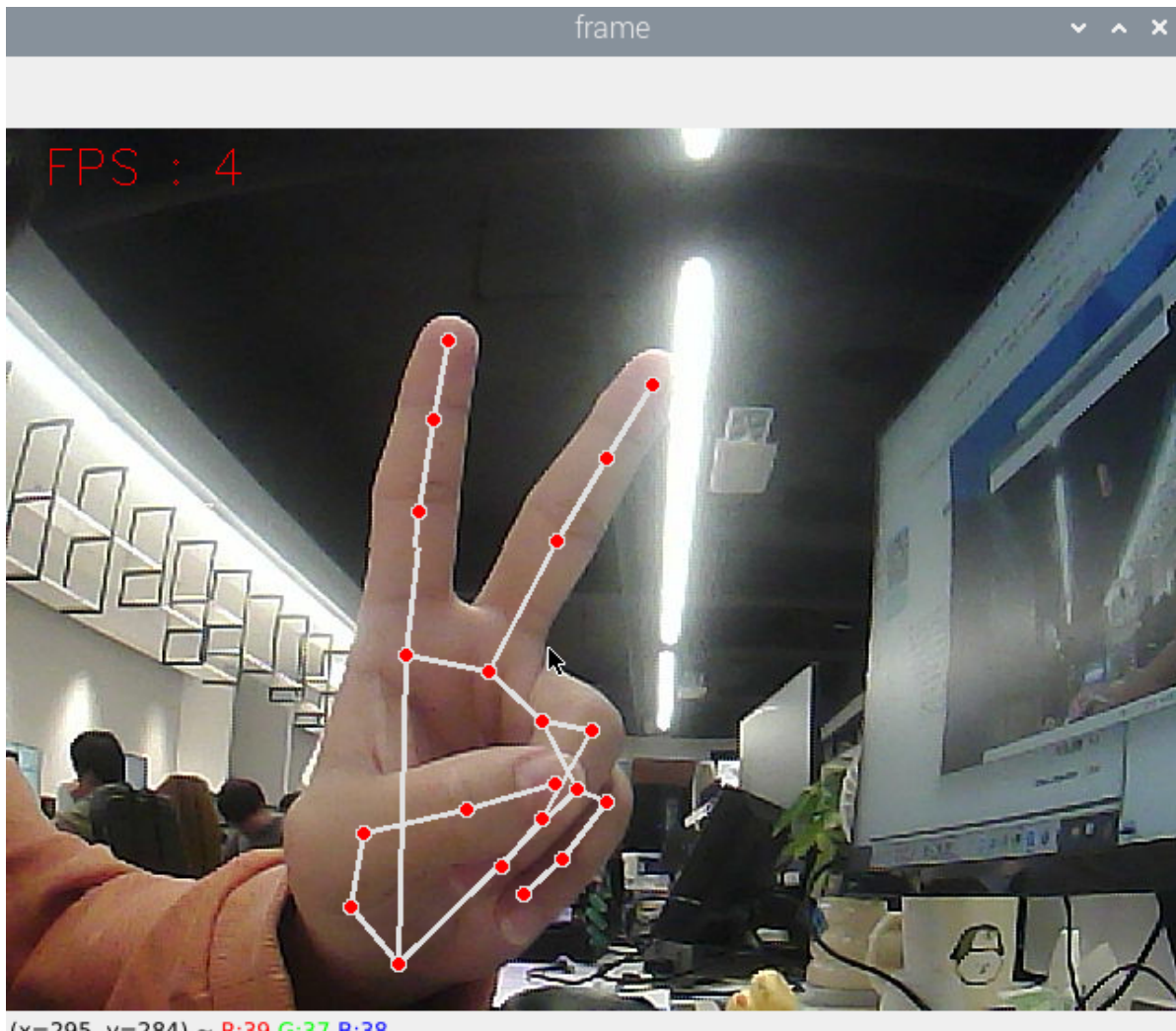
After entering the docker container, enter the terminal,

```
ros2 run yahboom_esp32ai_car FingerCtrl
```

If the Angle of the camera is not at this Angle, please press CTRL+C to end the program and run again, this is because the network delay causes the Angle of sending the steering gear to lose packets

If the camera picture image appears upside down, you need to see 3. Camera picture correction (must see) document itself correction, the experiment is no longer described.

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



4、 Core code

4.1、 FingerCtrl.py

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

The implementation process here is also very simple, the main function opens the camera to obtain data and passes it into the process function, which carries out "detect the palm "->" get finger coordinates "->" get gesture" in order, and then decides the action that needs to be executed according to the gesture results.

4.2、 Flow chart

