Gesture control car to make shapes

Note: The ROS_DOMAIN_ID of the Raspberry Pi needs to be consistent with that of the microROS control board. You can check [MicroROS Control Board Parameter Configuration] to set the microROS control board ROS_DOMAIN_ID. Check the tutorial [Connect MicroROS Agent] to determine whether the IDs are consistent.

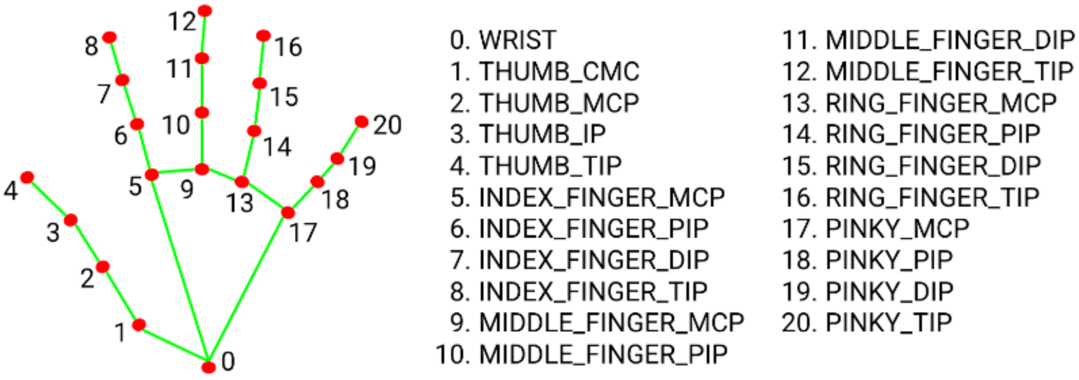
1、Program function description

After the function is turned on, the camera captures images and recognizes relevant gestures to control the movement of the car.

Gesture number "5"	Car stops
Gesture“yes”	The car moves in a square
Gesture “ok”	The car spins in circles
Gesture “rock”（The index finger and little finger are straight, the other fingers are bent)	The car takes an S shape
Gesture of contempt（Clench your fist, extend your thumb, and point your thumb down)	The car goes forward and then backward

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

MediaPipe Hands infers the 3D coordinates of 21 hand-valued 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

After entering the docker container, the location of the source code of this function is as follows.

```
/root/yahboomcar_ws/src/yahboomcar_mediapipe/yahboomcar_mediapipe/
```

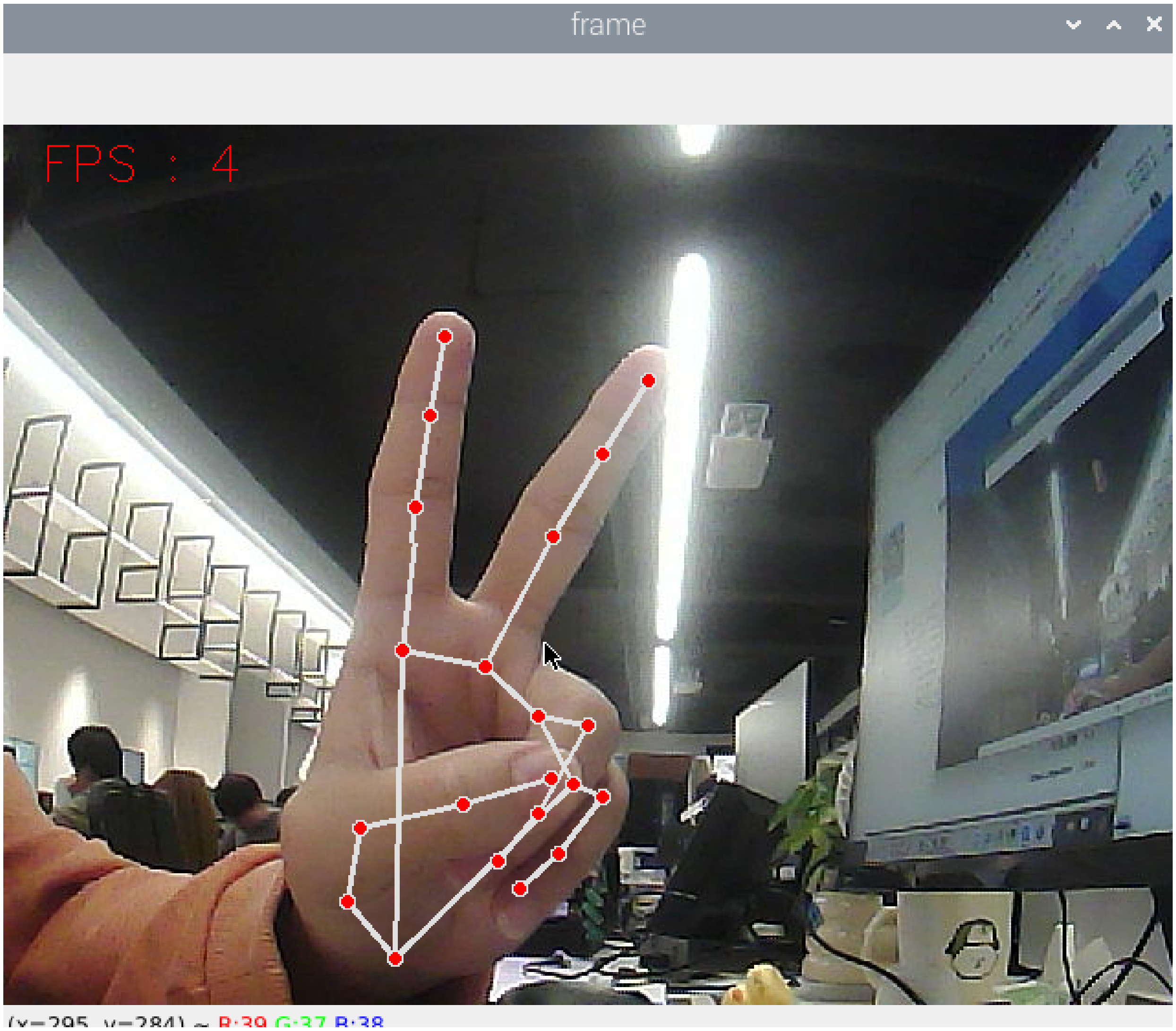
3、Program start

3.1、Start command

After entering the docker container, enter the following command in the terminal

```
ros2 run yahboomcar_mediapipe FingerCtrl
```

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



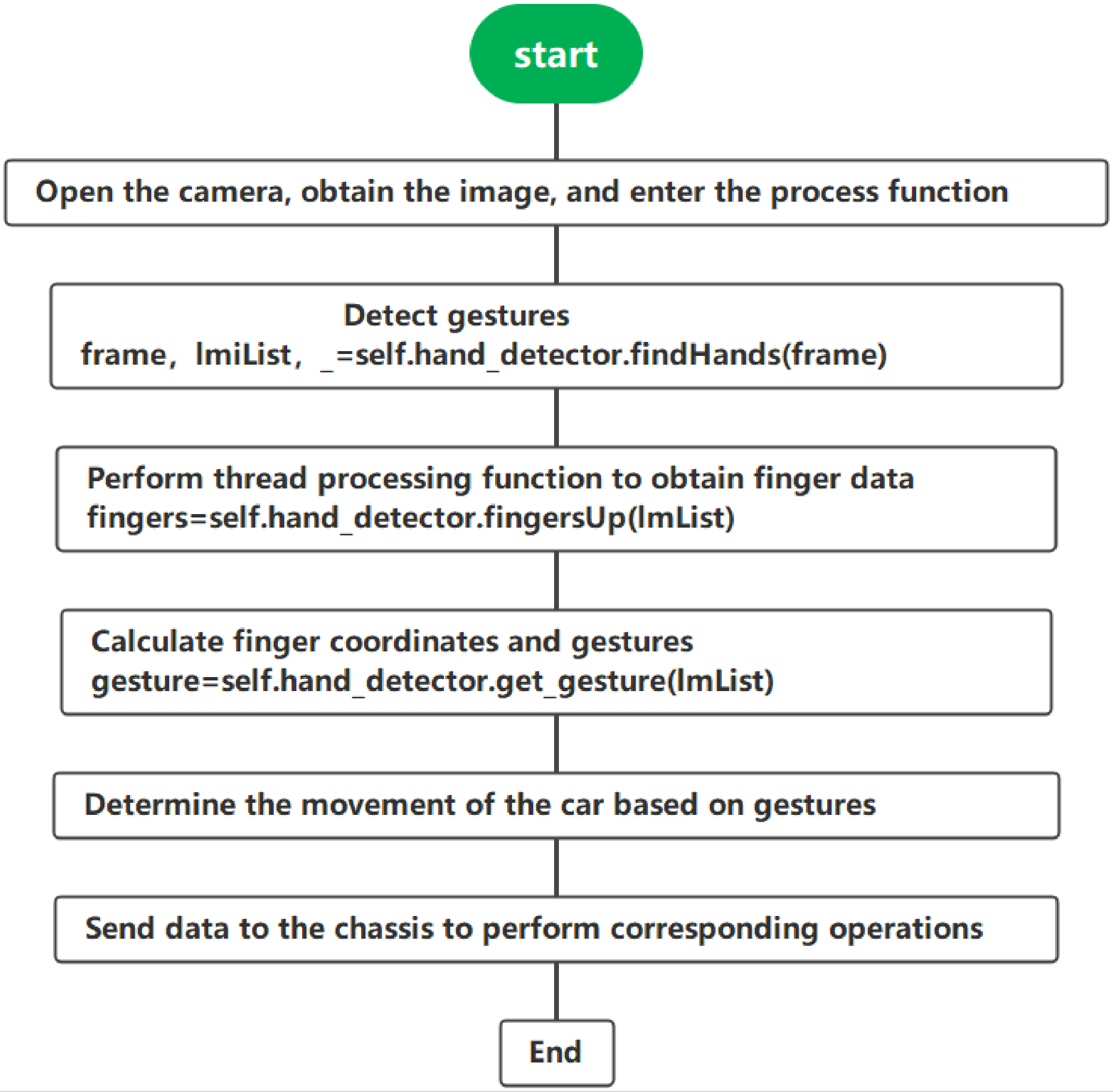
4、core code

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, which performs "detect palm" -> "get finger coordinates" -> "get gesture" in order, and then determine the needs based on the gesture results. The action performed.

4.2、flow chart



Open the camera, obtain the image, and enter the process function

Detect gestures
frame，lmiList，_=self.hand_detector.findHands(frame)

Perform thread processing function to obtain finger data
fingers=self.hand_detector.fingersUp(lmList)

Calculate finger coordinates and gestures
gesture=self.hand_detector.get_gesture(lmList)

Determine the movement of the car based on gestures

Send data to the chassis to perform corresponding operations