Control the car movement with your palm

Note: The ROS_DOMAIN_ID of the Raspberry Pi and the microROS control board need to be consistent. 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.
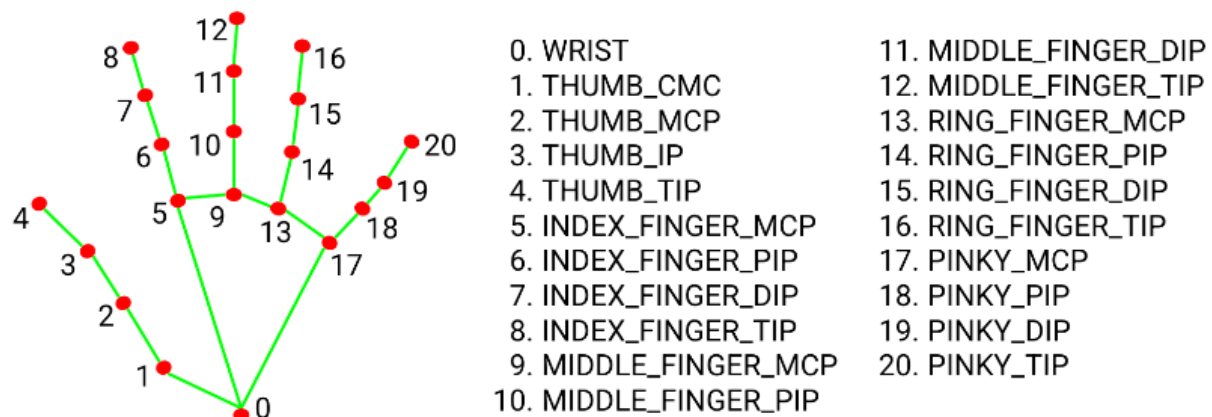
1、Introduction to mediapipe

MediaPipe is an open source data stream processing machine learning application development framework developed by Google. It is a graph-based data processing pipeline for building data sources using many forms, such as video, audio, sensor data, and any time series data. MediaPipe is cross-platform and can run on embedded platforms (Raspberry Pi, etc.), mobile devices (iOS and Android), workstations and servers, and supports mobile GPU acceleration. MediaPipe provides cross-platform, customizable ML solutions for live and streaming media.

MediaPipe Hands is a high-fidelity hand and finger tracking solution. It uses machine learning (ML) to infer the 3D coordinates of 21 hands from a frame.

After palm detection on the entire image, the 21 3D hand joint coordinates in the detected hand area are accurately positioned by regression according to the hand marking model, that is, direct coordinate prediction. The model learns a consistent internal hand pose representation that is robust even to partially visible hands and self-occlusion.

In order to obtain ground truth data, about 30K real-world images were manually annotated with 21 3D coordinates, as shown below (get the Z value from the image depth map, if there is a Z value for each corresponding coordinate). To better cover possible hand poses and provide additional supervision over the nature of the hand geometry, high-quality synthetic hand models in various backgrounds are also drawn and mapped to corresponding 3D coordinates.


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

2、Program Description

The case in this section may run very slowly on the robot main control. You can first set up the car to test after recognizing the palm, so the effect will be better.

The car will control the movement of the chassis based on the position of the palm of the hand in the screen.

Palm is at the top of the screen -> the car moves forward

The palm is at the bottom of the screen -> the car moves backwards

The palm is on the left side of the screen -> the car moves to the left

The palm is at the bottom of the screen -> the car moves to the right

2.1、Source code path

After entering the docker container, the source code location 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 in the terminal.

```
ros2 run yahboomcar_mediapipe RobotCtrl
```

Turn on this function, and then put your hand in front of the camera. The screen will recognize the palm. After the program recognizes the position of the palm, it will send the speed to the chassis to control the movement of the car.

(x=395, y=213) ~ R:230 G:231 B:228
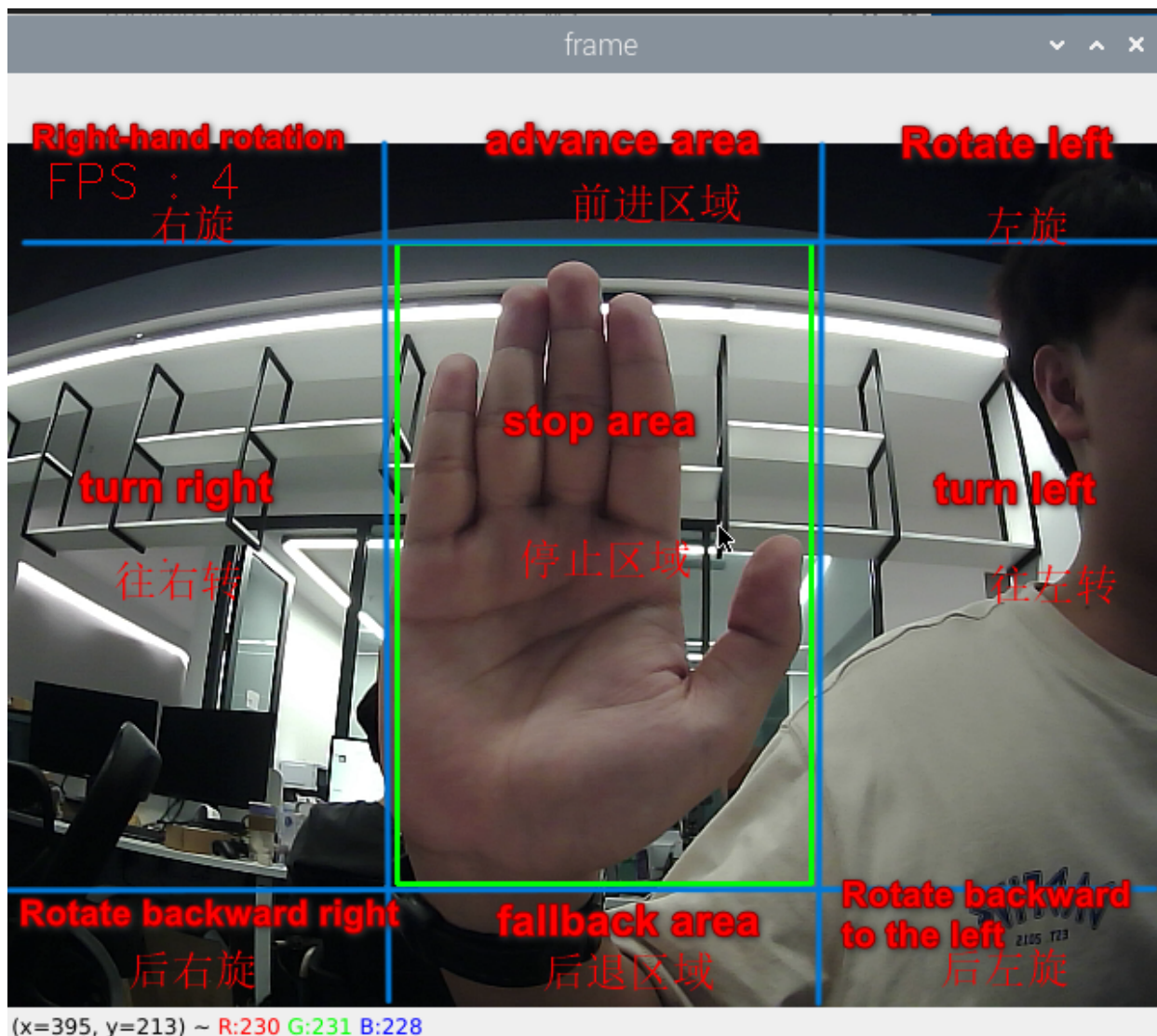
4、core code

4.1、RobotCtrl.py

- Code reference location

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

- code analysis

    1）、Import the corresponding library file

```
from media_library import *
```

This library file mainly includes detecting palms, fingers and obtaining the coordinates of each finger joint.

    2）、Detect palm and obtain finger coordinates

```
fingers = self.hand_detector.fingersUp(lmList)
point_x = lmList[9][1]  #x value
point_y = lmList[9][2]  #Y value
```

Combining the picture of 1. Introduction, we can know that what is actually obtained is the coordinate of the first joint of the middle finger of our palm. By judging the position of this coordinate in the picture, the speed in the xy direction of the chassis is sent to the chassis. Control can be achieved.

4.2、 flow chart

```
                    ┌──────────┐
                    │  start   │
                    └──────────┘
                          │
    ┌─────────────────────────────────────────────────────────────┐
    │ Open the camera, obtain the image, and enter the process function │
    └─────────────────────────────────────────────────────────────┘
                          │
    ┌─────────────────────────────────────────────────────────────┐
    │                      Detect fingers                          │
    │  frame,  lmiList,  _=self.hand_detector.findHands(frame)     │
    └─────────────────────────────────────────────────────────────┘
                          │
    ┌─────────────────────────────────────────────────────────────┐
    │                     Get finger data                          │
    │     fingers=self.hand_detector.fingersUp(lmList)             │
    └─────────────────────────────────────────────────────────────┘
                          │
    ┌─────────────────────────────────────────────────────────────┐
    │ According to the position of the coordinate value on the screen, send │
    │ the speed in the xy direction                                │
    │ sun(fingers)self.media_ros.pub_vel(x,y,z)                    │
    └─────────────────────────────────────────────────────────────┘
                          │
                    ┌──────────┐
                    │   end    │
                    └──────────┘
```