# Palm control car movement

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.

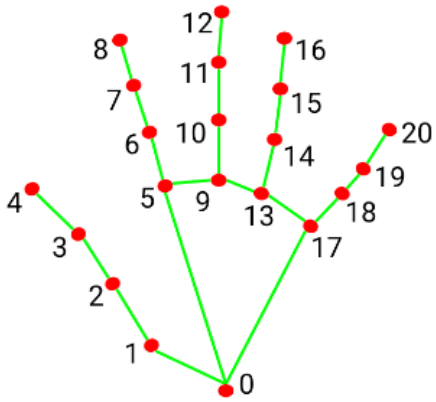
## 1、Introduction to mediapipe

MediaPipe is an open source and data stream processing machine learning application development framework developed by Google. It is a graph-based data processing pipeline for building data sources that use 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 GPU acceleration on mobile. 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 single frame。

After the palm detection of the entire image, the 21 3D hand joint coordinates in the detected hand region were accurately located by regression according to the hand marker model, that is, direct coordinate prediction. The model learns a consistent internal hand posture representation and is robust even to partially visible hands and self-occlusion.

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




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 specification

The case in this section may run at a very slow speed on the robot master, and the car can be tested first after recognizing the palm, so the effect will be better

The car will control the motion of the chassis according to the position of the hand in the picture.

Palm above the picture -> Car forward

Palm below the picture -> Car back

Palm on the left side of the screen -> Move the car left

Palm under the picture -> Car move right

### 2.1、Source path

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

## 3、Program initiation

### 3.1、Start command

Terminal input,

```
ros2 run yahboom_esp32ai_car RobotCtrl
```

**If the camera Angle is not at this Angle, please press CTRL+C to end the program and run it 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 screen will recognize the palm, the program recognizes the position of the palm, it will send the speed to the chassis, and then control the car movement.

## 4、Core code

#### 4.1、RobotCtrl.py

- Code reference position

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

- Code analysis

  1）、Import the corresponding library file

  ```
  from media_library import *
  ```

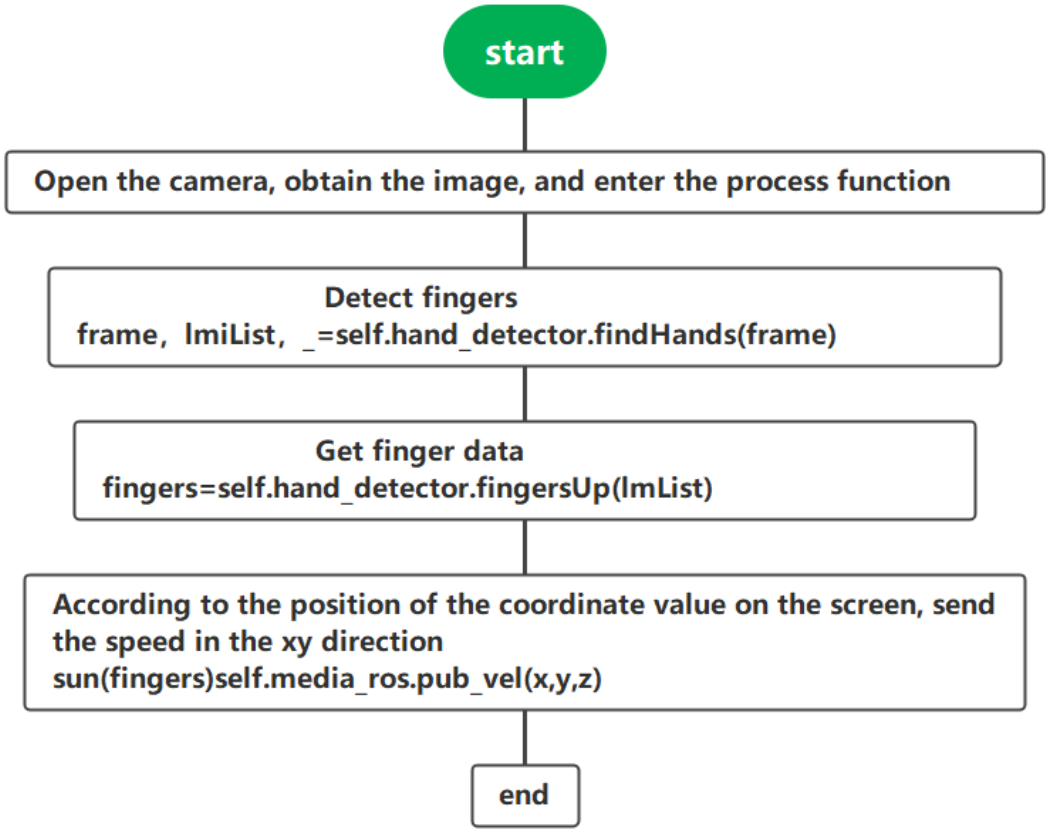
  This library file mainly consists of detecting the hand, fingers and obtaining the coordinates of each finger joint.

  2）、Check the palm, get the finger coordinates

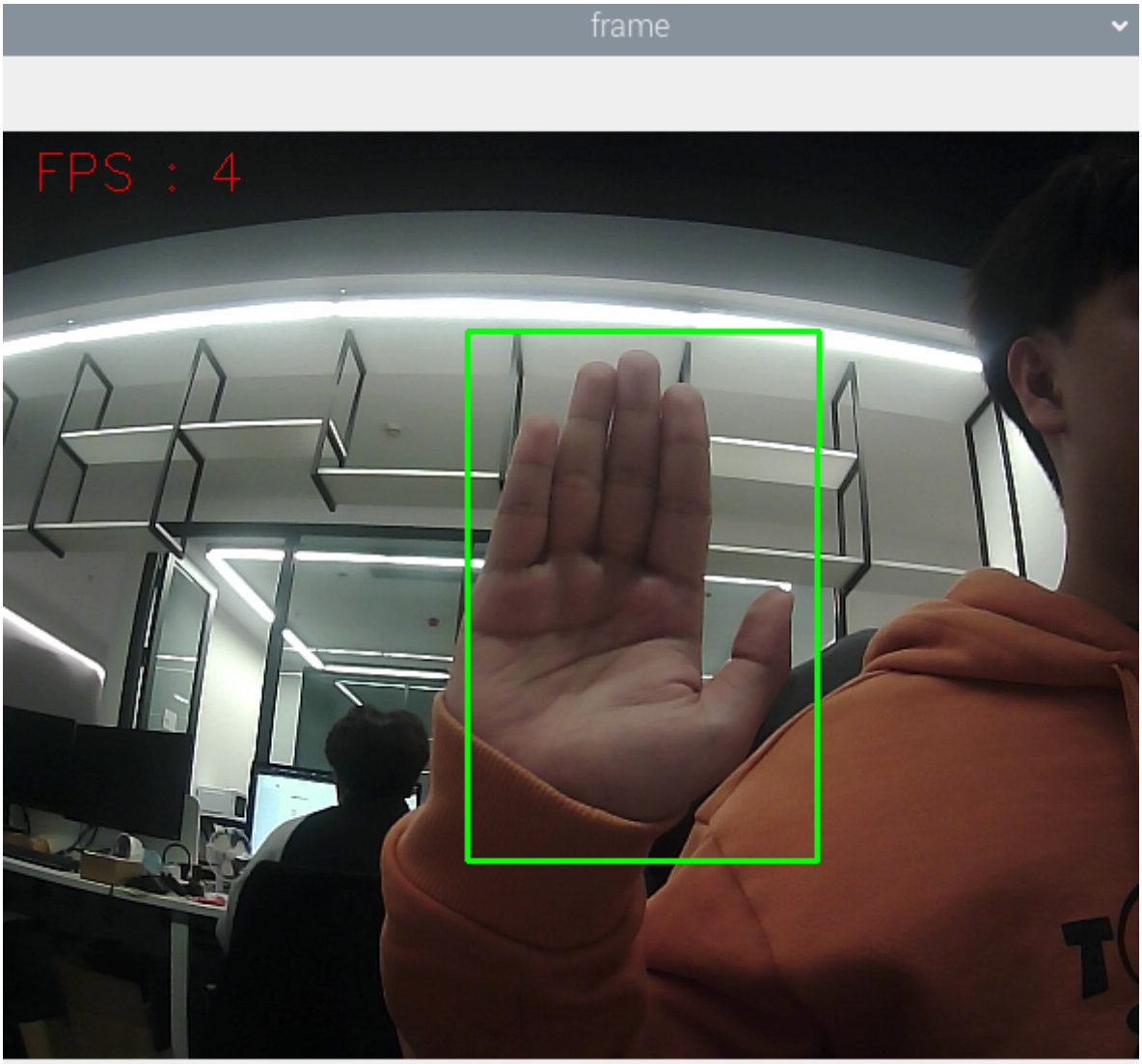
  ```
  fingers = self.hand_detector.fingersUp(lmList)
  point_x = lmList[9][1]
  point_y = lmList[9][2]
  ```

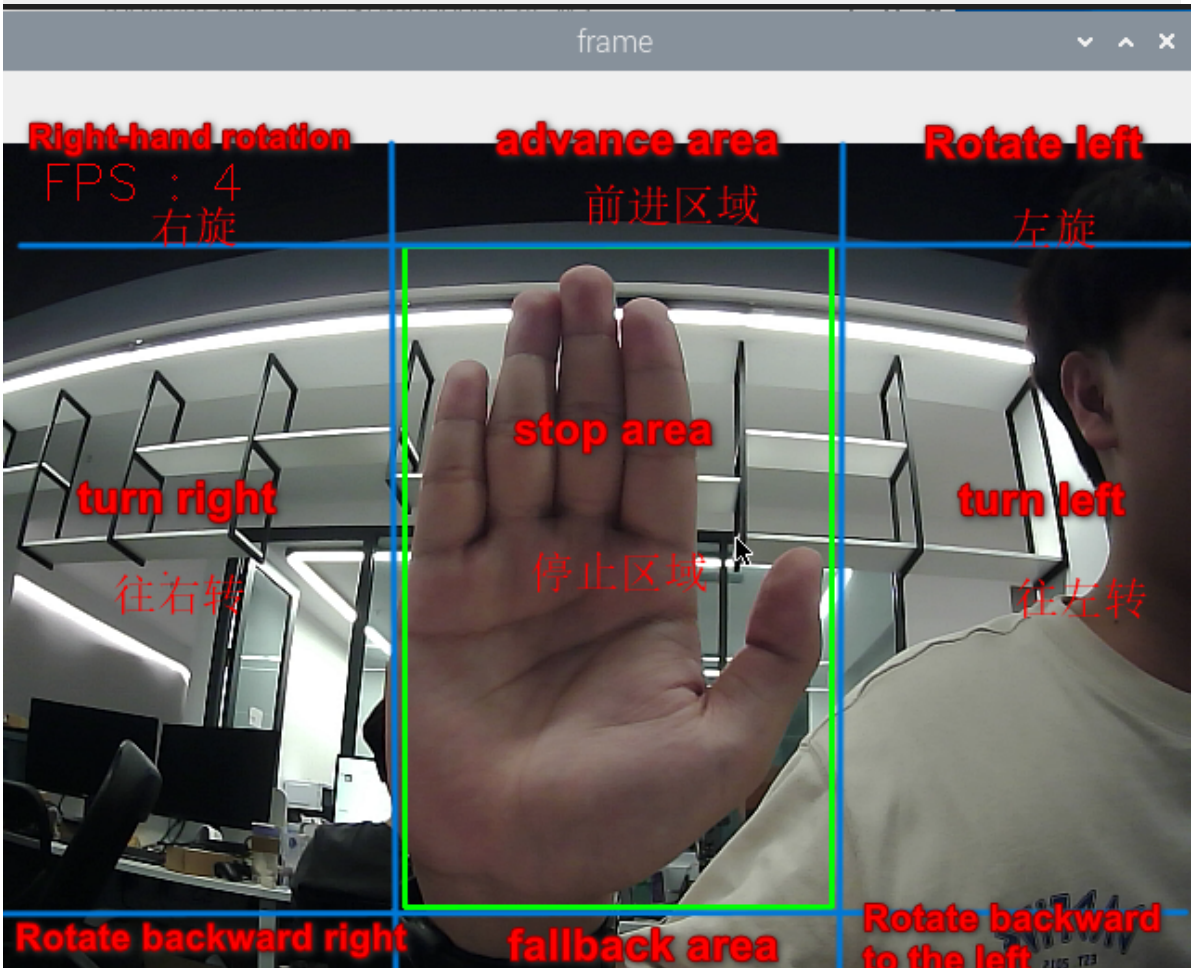
Combined with the picture introduced in 1. We can know that what we actually get is the coordinate ** of the first joint of the middle finger of our palm. By judging the position of this coordinate in the picture, we can send the speed in the xy direction to the chassis to achieve control.

## 4.2、Flow chart




## 4.3、Result graph


frame
FPS : 4


frame
Right-hand rotation
FPS : 4
右旋
advance area
前进区域
Rotate left
左旋
stop area
停止区域
turn right
往右转
turn left
往左转
Rotate backward right
fallback area
Rotate backward
to the left


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