Palm control servo PTZ

1. Introduction

MediaPipe is a data stream processing machine learning application development framework developed and open sourced by Google. It is a graph-based data processing pipeline used to build data sources in various 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 real-time and streaming media.

2. Program Description

After the program is started, after the camera captures the image, the two-dimensional gimbal of the car will follow the movement of the palm in the picture. The movement speed of the palm here should not be too fast, otherwise the image processing cannot keep up, which will cause lag.

2.1. Program code reference path

After entering the docker container, the source code of this function is located at,

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

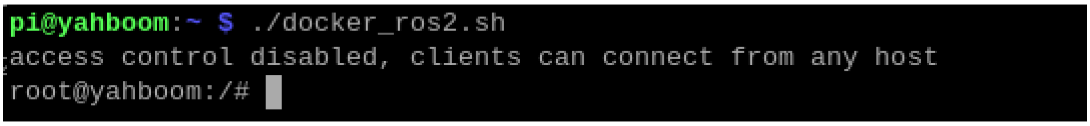
3. Program startup

3.1. Startup command

Open a terminal and enter the following command to enter docker,

```
./docker_ros2.sh
```

The following interface appears, which means that you have successfully entered docker



```
pi@yahboom:~ $ ./docker_ros2.sh
access control disabled, clients can connect from any host
root@yahboom:/#
```

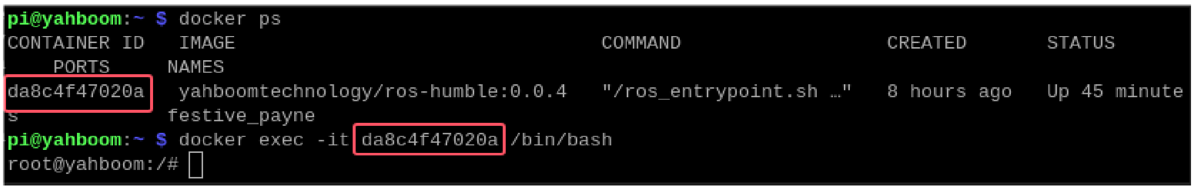
After entering the docker container, enter the terminal,

```
ros2 launch yahboomcar_bringup yahboomcar_bringup.launch.py
```

Open a new terminal, enter the same docker, and change the following da8c4f47020a to the ID displayed in the actual terminal

```
docker ps
```

```
docker exec -it da8c4f47020a   /bin/bash
```

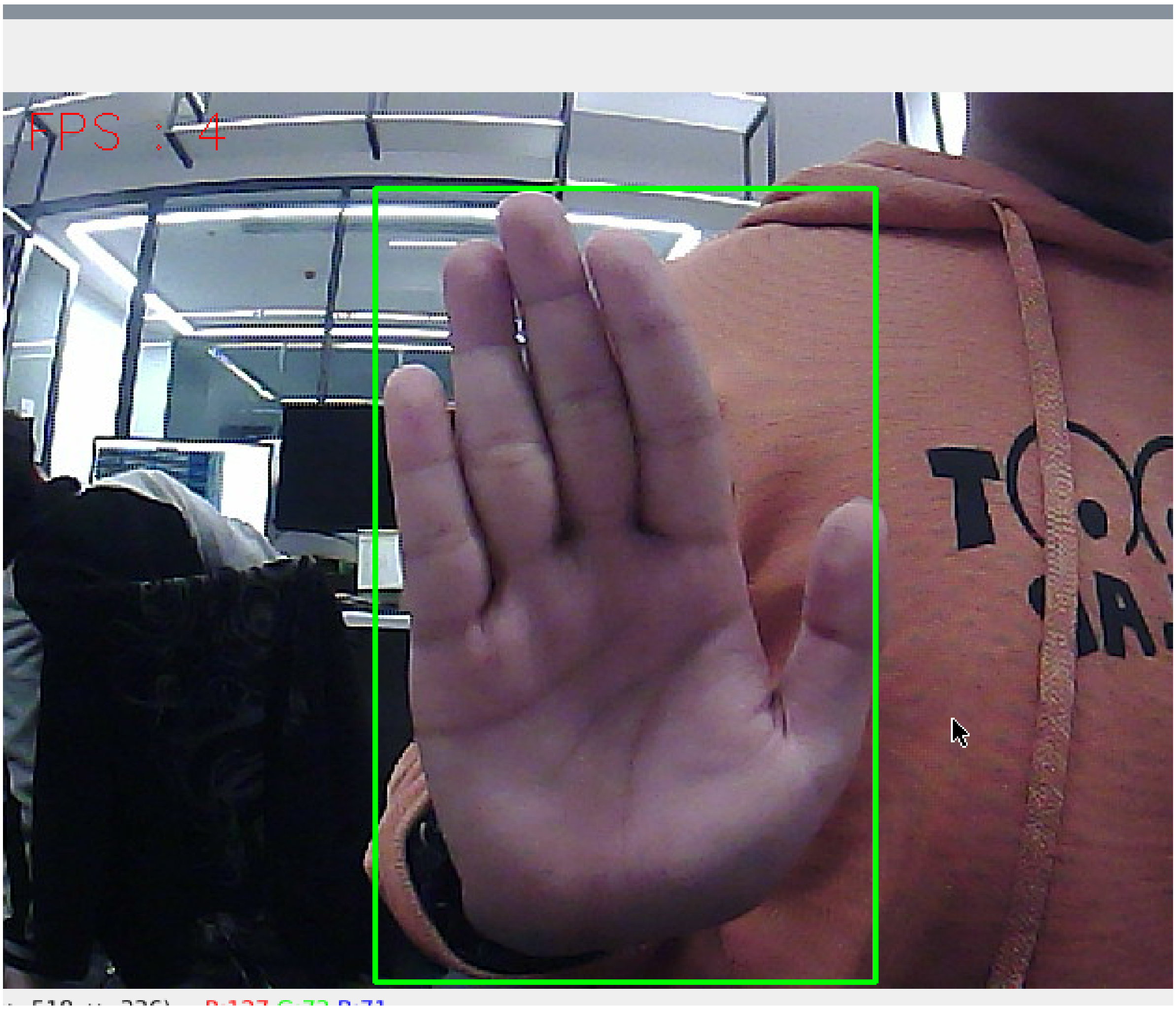


```
pi@yahboom:~ $ docker ps
CONTAINER ID   IMAGE                                COMMAND                   CREATED         STATUS
    PORTS     NAMES
da8c4f47020a   yahboomtechnology/ros-humble:0.0.4   "/ros_entrypoint.sh …"   8 hours ago    Up 45 minute
s             festive_payne
pi@yahboom:~ $ docker exec -it da8c4f47020a /bin/bash
root@yahboom:/# 
```

After entering the docker container, enter in the terminal,

```
ros2 run yahboomcar_mediapipe control_shape
```

After the function is turned on, the 2D gimbal of the car will follow the movement of the palm.



4. Code analysis

4.1、control_shape.py

- Code reference location

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

- Code Analysis

  1） Import important library files

  ```
  from media_library import *
  ```

  2）detect hands and obtain finger information

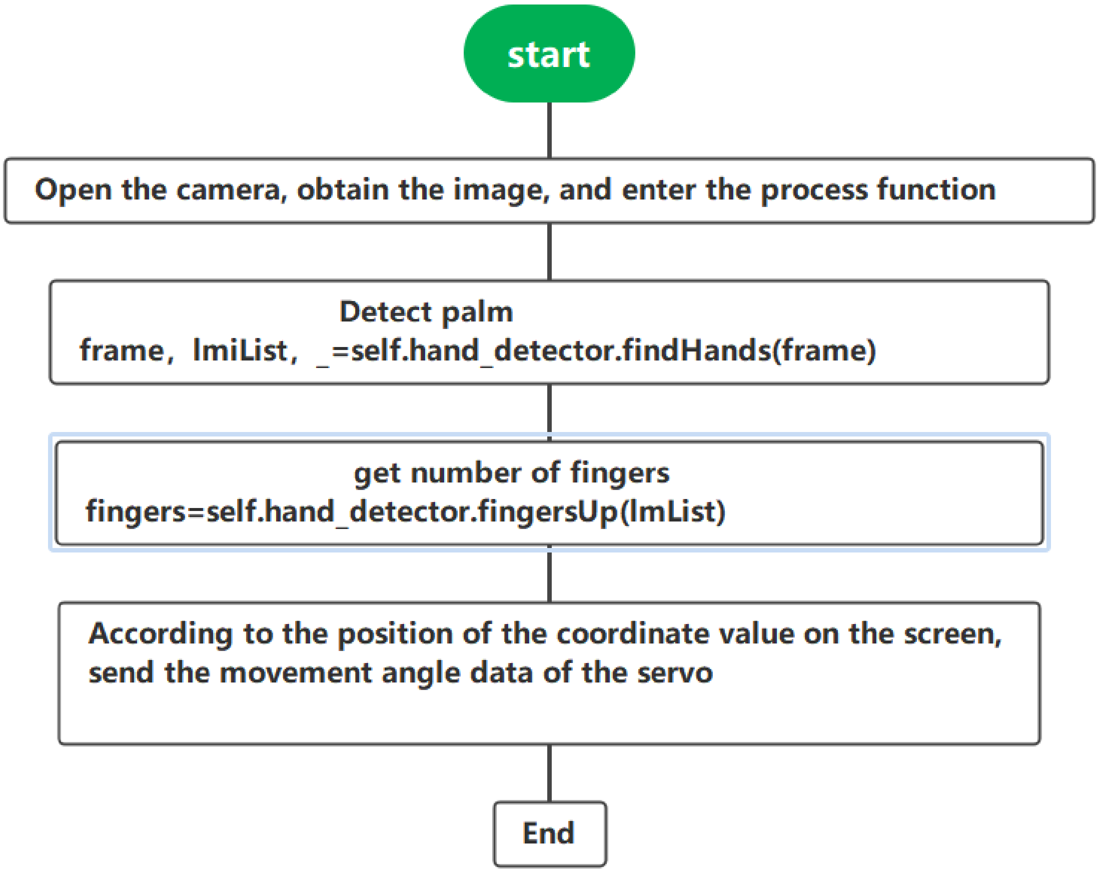
  ```
  frame, lmList, bbox = self.hand_detector.findHands(frame)
  #bbox is the minimum and maximum values ••of the xy frame of the detected hand. This value is very important. By calculating the center coordinates, the position of the palm in the picture can be determined. The source code is in media_library.py
  angle = self.hand_detector.ThumbTOforefinger(lmList)
  #Here, the bending angle of the thumb can be calculated to control the opening and closing of the gripper
  ```

  3）calculate the angle of each servo

  ```
              point_x = lmList[9][1]
              point_y = lmList[9][2]

              if point_y >= 270: self.y -= 2
              elif point_y <= 210: self.y += 2

              if point_x >= 350: self.x += 2
              elif point_x <= 290: self.x -= 2

              if self.x <= 0: self.x = 0
              elif self.x >= 180: self.x=180

              if self.y <= 0: self.y = 0
              elif self.y >=110: self.y = 110
  ```

  It can be seen that indexX and indexY are the coordinates of the center point of the picture frame. By judging, the angle that the servo should rotate is calculated. The parameters of each angle calculation method are calibrated according to the actual rotation angle and the visual range of the camera.

4.2 Flowchart


start
Open the camera, obtain the image, and enter the process function
Detect palm
frame，lmiList，_=self.hand_detector.findHands(frame)
get number of fingers
fingers=self.hand_detector.fingersUp(lmList)
According to the position of the coordinate value on the screen, send the movement angle data of the servo
End