Palm recognition tracking

Note: The virtual machine, ROS-wifi image transmission module and ESP32 communication board ROS_DOMAIN_ID need to be consistent, and both must be set to 20. You can view [ESP32 communication board parameter configuration] to set the ESP32 communication board ROS_DOMAIN_ID, and view the tutorial [Connecting MicroROS Agent] to determine whether the ID is consistent.

Before running the experiment, please make sure that the microros balance car and ROS-wifi image transmission module have correctly enabled the agent on the virtual machine (linux with humbleROS2 system)

1. Introduction

MediaPipe is an open source data stream processing machine learning application development framework developed and opened by Google. It is a graph-based data processing pipeline for building 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, the camera captures the image, and the car will follow the left and right movement of the palm in the picture. The movement speed of the palm should not be too fast, otherwise the image processing cannot keep up, which will cause lag.

2.1, Program code reference path

The location of the function source code is,

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

3, Program startup

3.1, Start command

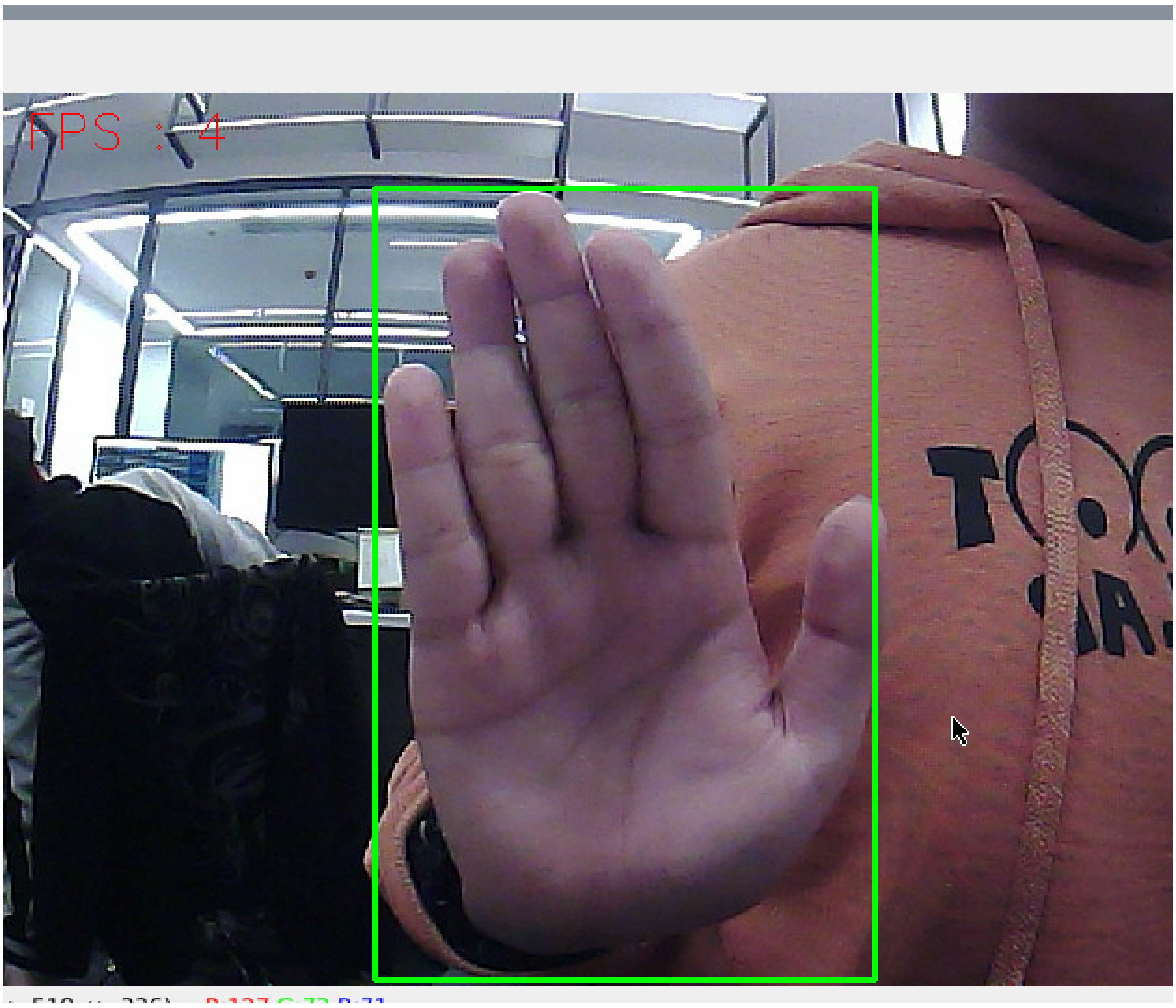
Input in the terminal,

```
#Start chassis driver
ros2 launch yahboomcar_bringup yahboomcar_bringup_launch.py
```

```
#Start palm recognition tracking program
ros2 run yahboom_esp32ai_car control_shape
```

If the camera image is inverted, you need to see **3. Camera image correction (select)** document to correct it yourself, this experiment will not be described.

After the function is turned on, the car will move left and right with the palm.



3.2, parameter adjustment

```
#angular PID
self.declare_parameter("angular_Kp",45.0)
self.angular_Kp =
self.get_parameter('angular_Kp').get_parameter_value().double_value
self.declare_parameter("angular_Ki",0.0)
self.angular_Ki =
self.get_parameter('angular_Ki').get_parameter_value().double_value
self.declare_parameter("angular_Kd",5.0)
self.angular_Kd =
self.get_parameter('angular_Kd').get_parameter_value().double_value
```

Mainly adjust the three PID parameters to make the car tracking more sensitive, modify the source code, and then recompile and update.

4. Code analysis

4.1. control_shape.py

- Code reference location

```
/home/yahboom/yahboomcar_ws/src/yahboom_esp32ai_car/yahboom_esp32ai_car/control_s
hape.py
```

- Code analysis

1), Import important library files

```
from media_library import *
```

2), Detect hands and get finger information

```
frame, lmList, bbox = self.hand_detector.findHands(frame)
#bbox is the minimum and maximum values of the xy frame that frames the detected
hand. This value is very important. By calculating the center coordinates, the
position of the palm on the screen can be determined. The source code is in
media_library.py
```

3), Some core codes are as follows,

```
#point_x value and area value are calculated based on linear velocity and angular
velocity
[z_Pid, x_Pid] = self.PID_controller.update([(point_x - 320)/10, (area -
self.area)/2])
if area == 0:
    print("Not Found Hand")
    self.pub_cmdVel.publish(Twist())
if area >self.area and x_Pid >0:
    self.linearx = -x_Pid
else:
    self.linearx = +x_Pid
if point_x > 320 and z_Pid > 0:
    self.angularz = +z_Pid
else:
    self.angularz = -z_Pid
if abs(point_x-320) < 40:
    self.angularz = 0.0
if self.area-20 < area < self.area or abs(x_Pid)>0.8:
    self.linearx = 0.0
self.linearx = 0.0
print("z_Pid:{:.2f}, x_Pid:{:.2f},point_x:{:.2f},area:
{:.2f}".format(z_Pid,x_Pid,point_x,area))
self.media_ros.pub_vel(self.linearx, 0.0,self.angularz)
```