13. Mediapipe recognition and tracking palm control robotic arm

13.1. Introduction

MediaPipe is an open source data stream processing machine learning application development framework developed by Google. It is a graph-based number

Data processing pipelines for building data sources that use multiple 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.

13.2. Use

After the program is started and the camera captures the image, the robotic arm will follow the movement of the palm in the screen. The palm movement speed here should not be too fast, otherwise the image processing will not be able to keep up, which will cause lag.

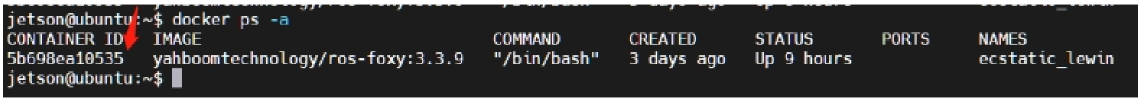
```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS/07, Docker tutorial
~/run_docker.sh
```

```
roslaunch arm_mediapipe mediaArm.launch # robot
```

<PI5 needs to open another terminal to enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

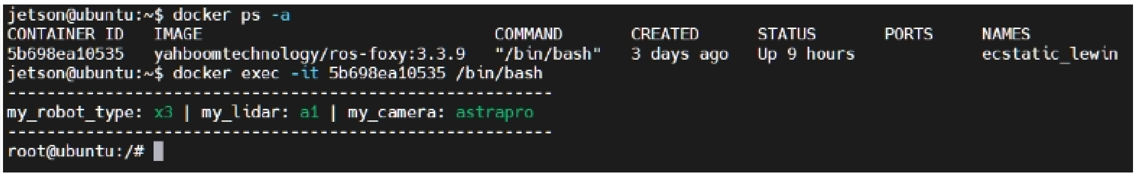
```
docker ps -a
```



```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                             COMMAND       CREATED      STATUS      PORTS     NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9  "/bin/bash"   3 days ago   Up 9 hours            ecstatic_lewin
jetson@ubuntu:~$
```

2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```



```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                             COMMAND       CREATED      STATUS      PORTS     NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9  "/bin/bash"   3 days ago   Up 9 hours            ecstatic_lewin
jetson@ubuntu:~$ docker exec -it 5b698ea10535 /bin/bash
-------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-------------------------------------------------------
root@ubuntu:/#
```

After successfully entering the container, you can open countless terminals to enter the container.

```
rosrun arm_mediapipe ArmCtrl.py #robot
```

After starting, press R2 on the handle to turn on the function.

After the function is turned on, the robotic arm will move with the palm, and at the same time, the end clamp will also change with the opening and closing of the palm.

13.3. Core code analysis ArmCtrl.py

- Code reference location

  ```
  ~/yahboomcar_ws/src/arm_mediapipe/scripts
  ```

- Code analysis

  1). Import important library files

  ```
  from media_library import *
  ```

  2. Detect hands and obtain finger information

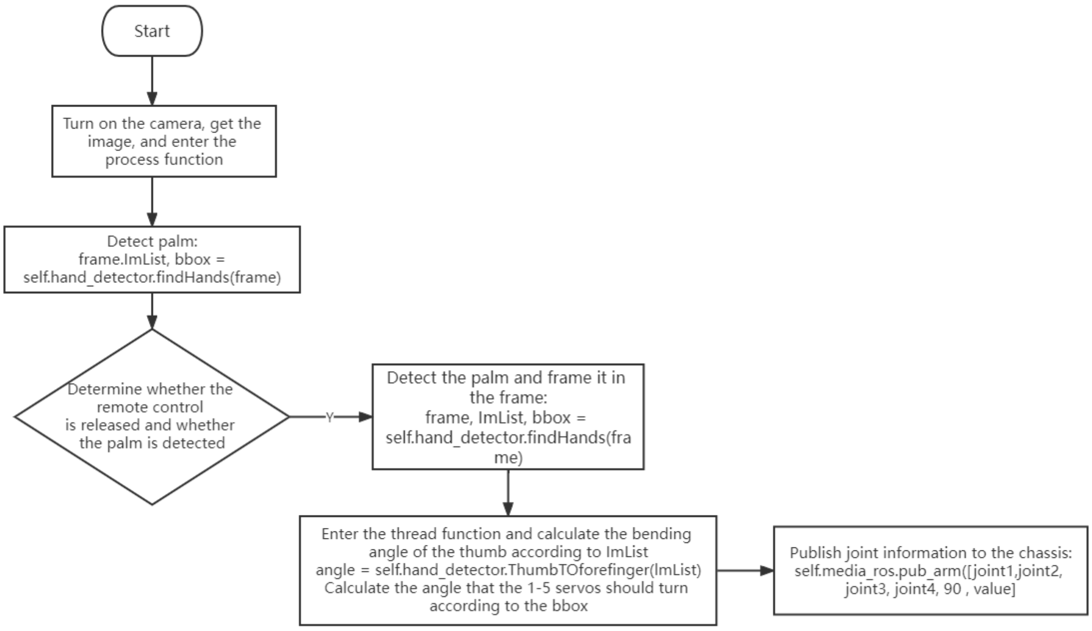
  ```
  frame, lmList, bbox = self.hand_detector.findHands(frame)
  #bbox is the minimum and maximum xy value of the frame that frames the detected hand. This value is very important. By calculating the center coordinates, the position of the palm on the screen can be determined. The source code is in media_library.py
  angle = self.hand_detector.ThumbTOforefinger(lmList)
  #Here you can calculate the bending angle of your thumb to control the opening and closing of the clamping jaws
  ```

  3. Calculate the angle of each steering gear

  ```
  indexX = (bbox[0] + bbox[2]) / 2
  indexY = (bbox[1] + bbox[3]) / 2
  if indexY > 400: indexY = 400
  elif indexY < 200: indexY = 200
  joint2 = -0.4 * indexY + 170
  joint3 = 0.05 * indexY + 25
  joint4 = -0.125 * indexY + 85
  if 300 < indexX < 340: joint1 = 90
  else: joint1 = -0.3 * indexX + 186
  ```

  It can be seen that the index Calibrated according to the range.

13.4. Flowchart


Start
Turn on the camera, get the image, and enter the process function
Detect palm: frame.ImList, bbox = self.hand_detector.findHands(frame)
Determine whether the remote control is released and whether the palm is detected
Y
Detect the palm and frame it in the frame: frame, ImList, bbox = self.hand_detector.findHands(frame)
Enter the thread function and calculate the bending angle of the thumb according to ImList angle = self.hand_detector.ThumbTOforefinger(lmList) Calculate the angle that the 1-5 servos should turn according to the bbox
Publish joint information to the chassis: self.media_ros.pub_arm([joint1,joint2, joint3, joint4, 90 , value]