

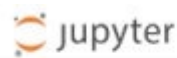
# Gesture tracking

## 1. Purpose of the experiment

This tutorial will teach you how to detect finger positions in real time through visual recognition and control the joint motors of the robot dog to achieve dynamic tracking. The system includes coordinate transformation, kinematic calculation and real-time posture adjustment, so that the robot dog can move with the finger.

## 2. Main source code path

First, end the big program, then open the browser and enter "ip (ip is the ip of the robot dog): 8888", enter the password "yahboom" and enter



Password:

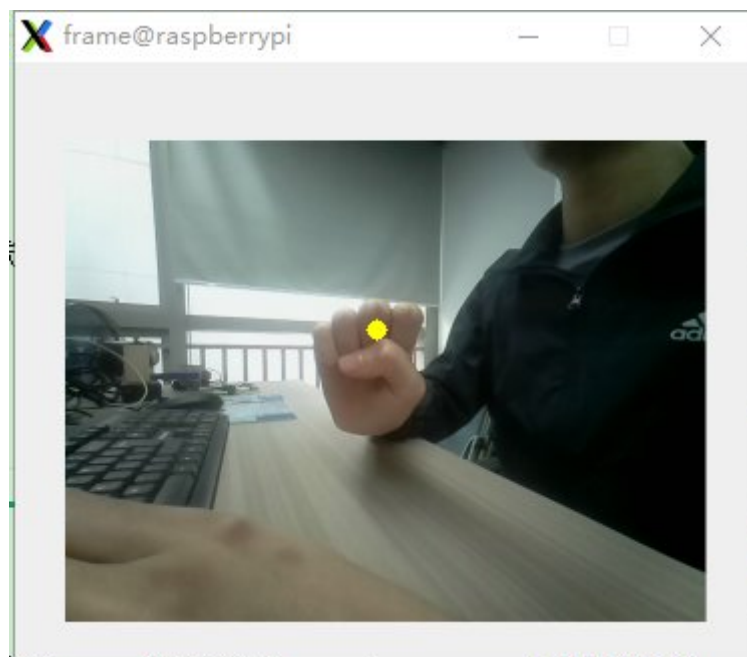
Log in

**the path to** ~/DOGZILLA\_Lite\_class/6.AI Visual Interaction Course/09.Gesture following. Open the program **Gesture\_follows.ipynb** and run it , or enter it in the terminal

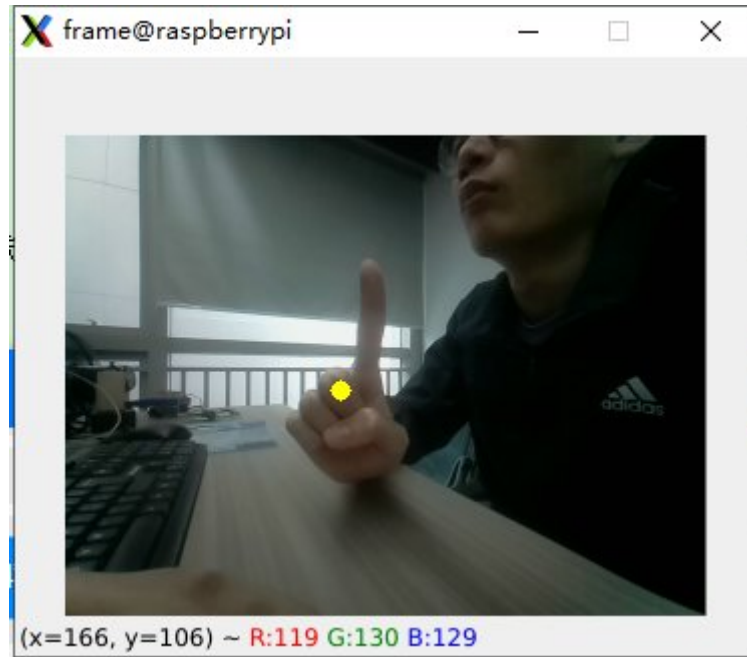
```
cd ~/DOGZILLA_Lite_class/6.AI Visual Interaction Course/09.Gesture following  
python3 Gesture_follows.py
```

## 3. Experimental Phenomenon

After running the source code, you can use the finger position to let the robot dog track it.



If you clench your fist, no tracking will occur , as shown in the picture



## 4. Main source code analysis

```
try:
    while True:
        global bot
        ret, frame = cap.read()
        img_height, img_width, _ = frame.shape
        hand_detector.findHands(frame, draw=False)
        if len(hand_detector.lmList) != 0:
            # 转向控制部分
            # Turning control section
            # MediaPipe中，手部最中心的指关节的编号为9
            # In MediaPipe, the index of the central finger joint is 9
            x,y = hand_detector.findPoint(9)
            cv2.circle(frame,(int(x),int(y)),2,(0,255,255),6)

            value_x = x - 160
            value_y = y - 120

            if value_x > 55:
                value_x = 55
            elif value_x < -55:
                value_x = -55
            if value_y > 75:
                value_y = 75
            elif value_y < -75:
                value_y = -75

            finger_number = hand_detector.get_gesture()
            finger_str=f"Number:{finger_number}"

            #print(finger_number)

            if(finger_number != "Zero"):
                g_dog.attitude(['y','p'],[-value_x/10, value_y/10])
        else:
```

```

x = 0
y = 0

try:
    cv2.imshow('frame', frame)
    cv2.waitKey(1)

    #图片显示在lcd屏上 The picture is displayed on the LCD screen
    b,g,r = cv2.split(frame)
    frame = cv2.merge((r,g,b))
    frame = cv2.flip(frame, 1)
    imgok = Image.fromarray(frame)
    mydisplay.ShowImage(imgok)
except:
    continue

    if button.press_b():
        break
finally:
    g_dog.reset()
    cap.release()

```

The above source code controls the robot dog according to the position of the finger. Press the button on the lower left of the screen to terminate this experiment (this experiment is only valid when running on the terminal)