Gesture control

1. Experimental purpose

Drive the car for gesture control

2. Experimental path source code

Enter the car system, end the car program, enter "ip (ip is the car's ip): 8888" in the browser, enter the password "yahboom"

Then log in
Enter the path **Rider-pi_class/6.AI Visual Interaction Course/7. Gesture controlled car/** and run **hands.ipynb**.
Or enter the terminal and type

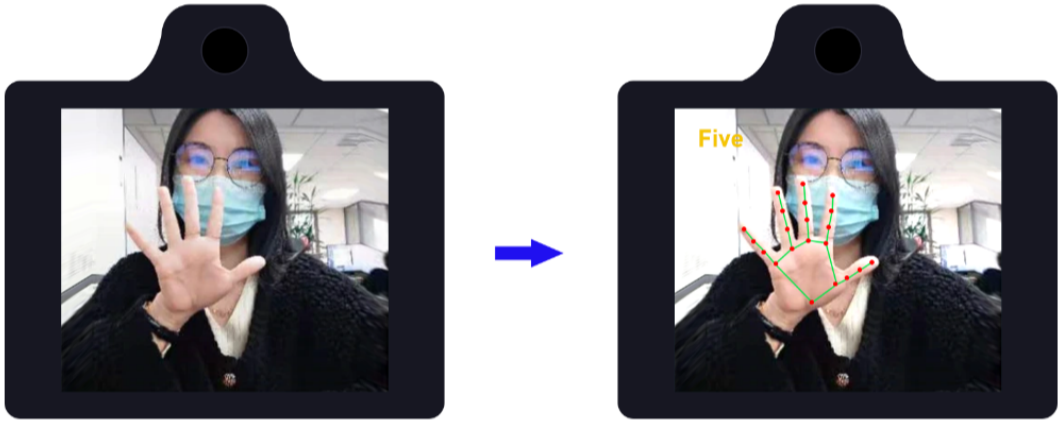
```
cd /home/pi/Rider-pi_class/6.AI Visual Interaction Course/7. Gesture controlled car
python3 hands.py
```

3. Experimental phenomenon

After running the source code, the car can recognize gestures such as 1, 2, 3, 4, 5, 6, good, fist, etc. Note that you need to wait for the robot to complete the previous action before recognizing again.
1: Swing left and right
2: Squat up and down
3: Move forward and backward
4: Rotate left and right
5: Ballet

6: Swing left and right and dance

The car cannot be in a stopped state (that is, not a standing balance state), otherwise it cannot move.



4. Analysis of main source code parameters

```
with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
  while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("Can not receive frame (stream end?). Exiting...")
        break
    frame_RGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    result = hands.process(frame_RGB)
    frame_height = frame.shape[0]
    frame_width  = frame.shape[1]
    gesture_result=[]
    if result.multi_hand_landmarks:
        for i, handLms in enumerate(result.multi_hand_landmarks):
            mpDraw.draw_landmarks(frame,
                                  handLms,
                                  mpHands.HAND_CONNECTIONS,
                                  landmark_drawing_spec=handLmsStyle,
                                  connection_drawing_spec=handConStyle)

            for j, lm in enumerate(handLms.landmark):
                xPos = int(lm.x * frame_width)
                yPos = int(lm.y * frame_height)
                landmark_ = [xPos, yPos]
                landmark[j,:] = landmark_

            for k in range (5):
                if k == 0:
                    figure_ =
finger_stretch_detect(landmark[17],landmark[4*k+2],landmark[4*k+4])
                else:
                    figure_ =
finger_stretch_detect(landmark[0],landmark[4*k+2],landmark[4*k+4])
```

```
                figure[k] = figure_

            gesture_result = detect_hands_gesture(figure)

    b,g,r = cv2.split(frame)
    frame = cv2.merge((r,g,b))
    frame = cv2.flip(frame, 1)
    if result.multi_hand_landmarks:
      cv2.putText(frame, f"{gesture_result}", (10,30), cv2.FONT_HERSHEY_COMPLEX,
1, (255 ,255, 0), 5)
    if time.time()>cartime:
      if car_type=='L' or car_type=='M':
        if gesture_result=="good":
          cartime=time.time()
          car.action(23)
          cartime+=3
        elif gesture_result=="one":
          cartime=time.time()
          car.action(7)
          cartime+=3
        elif gesture_result=="two":
          cartime=time.time()
          car.action(8)
          cartime+=3
        elif gesture_result=="three":
          cartime=time.time()
          car.action(9)
          cartime+=3
        elif gesture_result=="four":
          cartime=time.time()
          car.action(22)
          cartime+=3
        elif gesture_result=="five":
          cartime=time.time()
          car.action(1)
          cartime+=3
        elif gesture_result=="six":
          cartime=time.time()
          car.action(24)
          cartime+=3
        elif gesture_result=="OK":
          cartime=time.time()
          car.action(19)
          cartime+=3
        elif gesture_result=="stone":
          cartime=time.time()
          car.action(20)
          cartime+=3
      elif car_type=='R':
        if gesture_result=="one":
          cartime=time.time()
          car.action(1)
          cartime+=5
        elif gesture_result=="two":
```

```
            cartime=time.time()
            car.action(2)
            cartime+=5
        elif gesture_result=="three":
            cartime=time.time()
            car.action(3)
            cartime+=5
        elif gesture_result=="four":
            cartime=time.time()
            car.action(4)
            cartime+=5
        elif gesture_result=="five":
            cartime=time.time()
            car.action(5)
            cartime+=5
        elif gesture_result=="six":
            cartime=time.time()
            car.action(6)
            cartime+=5


    imgok = Image.fromarray(frame)
    display.ShowImage(imgok)

    #把结果显示到电脑上
    #The picture is displayed on the computer
    r,g,b = cv2.split(frame)
    frame1 = cv2.merge((b,g,r))
    cv2.imshow("image1",frame1)

    if cv2.waitKey(5) & 0xFF == 27:
      break
    if button.press_b():
      car.reset()
      break
cap.release()
```

From the source code analysis, we can see that the car will make corresponding movements according to the camera's gestures. It will not be able to recognize new gestures until the old gestures are completed.