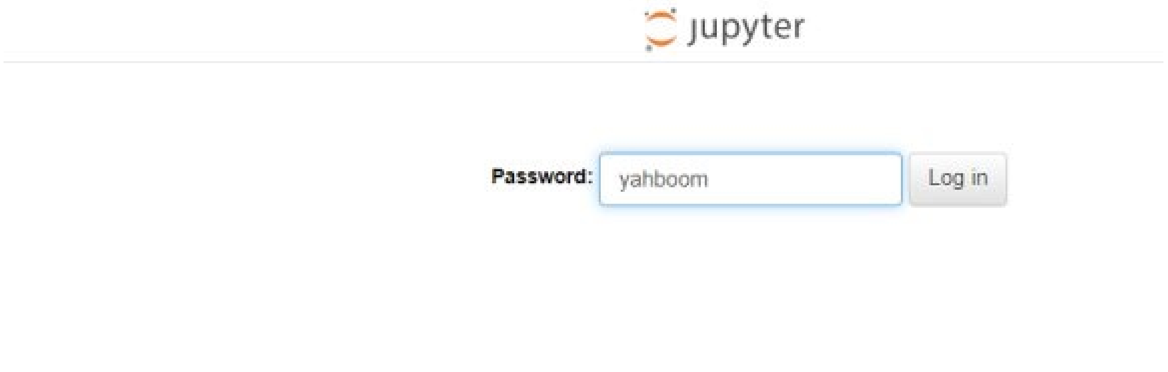
Human pose estimation

1. Purpose of the experiment

Driving the robot dog to detect human posture

2. Experimental path source code

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

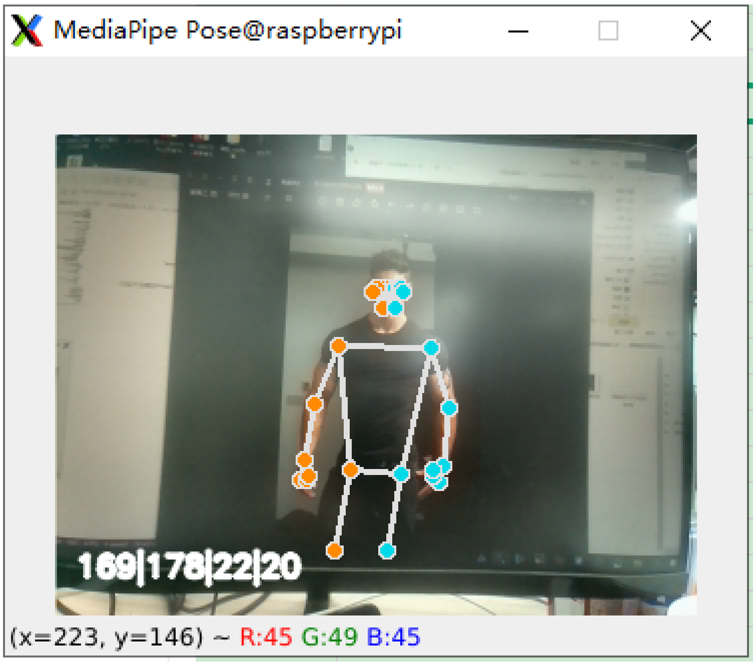


and log in. Enter the path of **cd ~/DOGZILLA_Lite_class/5.AI Visual Recognition Course/07. Human pose estimation** and run **pose.ipynb** . You can also enter the command in the terminal to directly start the python script

```
python3 pose.py
```

3. Experimental Phenomenon

After running the source code, you can see that the robot dog can imitate human postures.



4. Main program source code

```python
def main():
    global h, w, start_time, status, height, quitmark
    flag = False
    if not cap.isOpened():
        print("Camera not open")
        exit()

    tmp = f"a{sport['count']}\n"
    tmp = f"b{sport['calories']}\n"


    while not flag:
        ret, frame = cap.read()
        if not ret:
            print("Read Error")
            break
        frame = cv2.flip(frame, 1)
        rgbframe = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        poseoutput = poses.process(rgbframe)
        h, w, _ = frame.shape
        preview = frame.copy()

        if poseoutput.pose_landmarks:
            mpdraw.draw_landmarks(
                preview, poseoutput.pose_landmarks, mppose.POSE_CONNECTIONS
            )
            knee_angles = get_knee_angle(poseoutput.pose_landmarks.landmark)
            body_ratio = get_body_ratio(poseoutput.pose_landmarks.landmark)
            avg_angle = (knee_angles[0] + knee_angles[1]) // 2

            # determine the status
            if status:
                if avg_angle > 160:
                    status = False
                    pass_time = time.time() - start_time
                    start_time = 0
                    if 3000 > pass_time > 3:
                        sport["count"] = sport["count"] + 1
                        sport["calories"] = sport["calories"] + int(0.66 *
pass_time)
                        logger(sport["count"], sport["calories"])
                        tmp = f"a{sport['count']}\n"
                        tmp = f"b{sport['calories']}\n"

            else:
                if avg_angle < 120 and body_ratio < 1.2:
                    start_time = time.time()
                    status = True
            height = int(115 - (180 - avg_angle) / 90 * 40)
            print(avg_angle, height)

            if status:
                cv2.putText(
                    preview,
                    f"{height} : {avg_angle:.1f} {body_ratio:.3f}",
                    (10, 40),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1,
                    (0, 255, 0),
                    1,
                    cv2.LINE_AA,
                )
            else:
                cv2.putText(
                    preview,
                    f"{height} : {avg_angle:.1f} {body_ratio:.3f}",
                    (10, 40),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1,
                    (0, 0, 255),
                    1,
                    cv2.LINE_AA,
                )
        else:
            start_time = 0
            # car.reset()
        b, g, r = cv2.split(preview)
        image = cv2.merge((r, g, b))
        # image = cv2.flip(image, 1)

        imgok = Image.fromarray(image)
        mydisplay.ShowImage(imgok)

        r, g, b = cv2.split(image)
        image1 = cv2.merge((b, g, r))
        image_widget.value = bgr8_to_jpeg(image1)
        #cv2.imshow("image1",image1)

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

The source code turns on the camera, and then detects the human body's posture accordingly. The results are displayed on the robot dog screen and the computer screen.