

Gesture Recognition

1. Purpose of the experiment

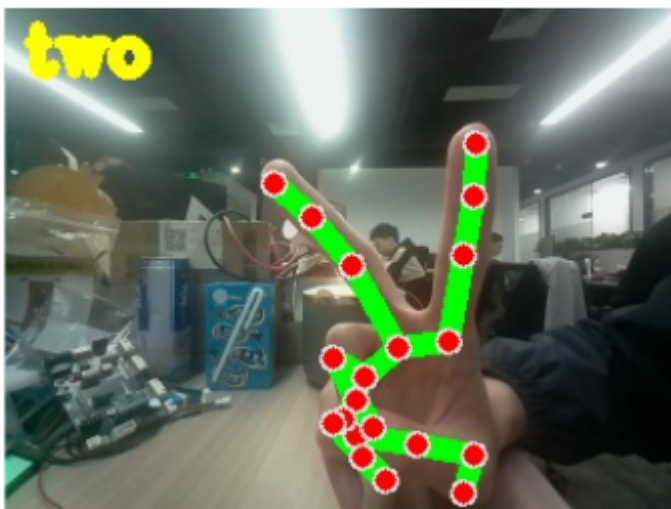
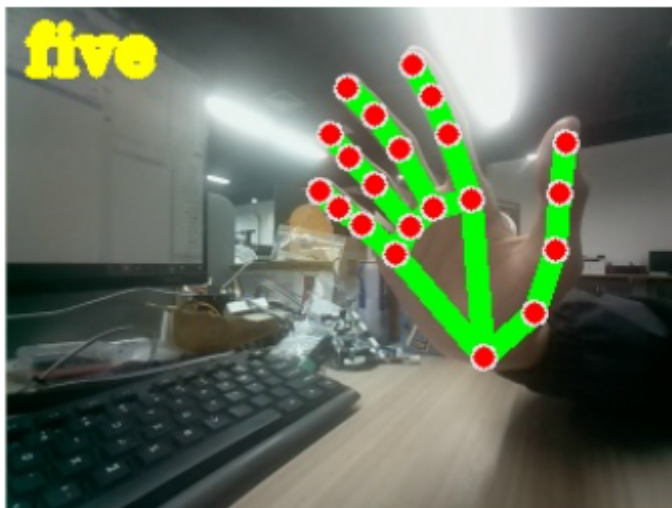
Realize the recognition of gestures by the robot dog. The gestures that can be recognized are: good, one, two, three, five, six, ok, stone

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 **DOGZILLA_Lite_class/5.AI Visual Recognition Course/15. Gesture recognition** and run **hands_detect.ipynb**.

3. Experimental Phenomenon

After running the source code, you can see that the robot dog will detect your fingers and display them.



4. Main source code analysis

```
# For webcam input:  
cap=cv2.VideoCapture(0)
```

```

cap.set(3,320)
cap.set(4,240)

mpHands = mp.solutions.hands
hands = mpHands.Hands()
mpDraw = mp.solutions.drawing_utils
handLmsStyle = mpDraw.DrawingSpec(color=(0, 0, 255), thickness=int(5))
handConStyle = mpDraw.DrawingSpec(color=(0, 255, 0), thickness=int(10))

figure = np.zeros(5)
landmark = np.empty((21, 2))

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)

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

    #把结果显示到电脑上 Display the results on the computer
    r,g,b = cv2.split(frame)
    frame1 = cv2.merge((b,g,r))
    image_widget.value = bgr8_to_jpeg(frame1)
    #cv2.imshow("image1",frame1)

    if cv2.waitKey(5) & 0xFF == 27:
        break

```

The robot dog calls the detected finger model and displays the recognized gestures on the robot dog's screen and the computer's screen.

The following gestures can be recognized: good, one, two, three, five, six, ok, stone