brush

1. Purpose of the experiment

Use the robot dog's camera to draw on the computer screen with your fingers.

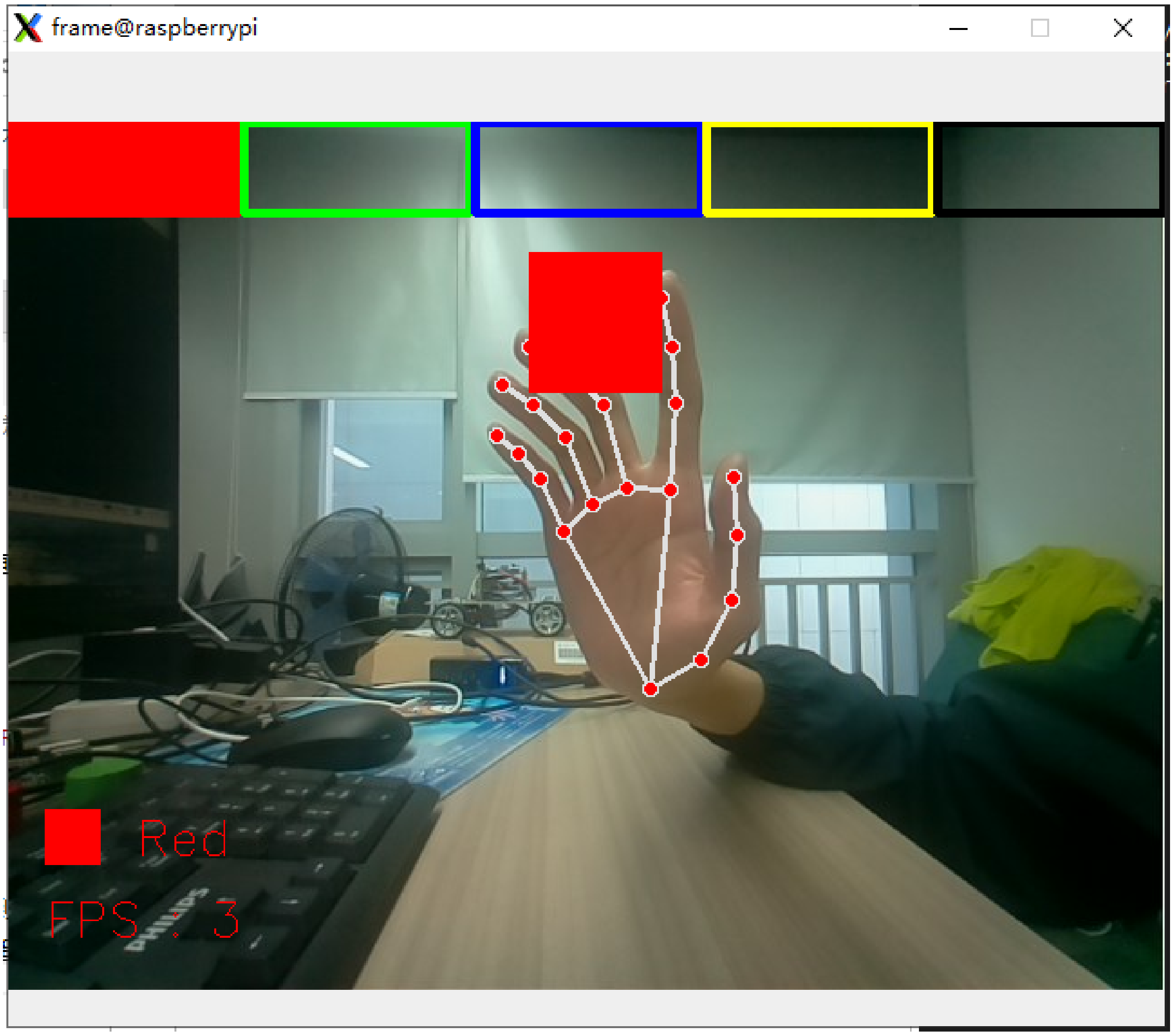
2. Experimental path source code

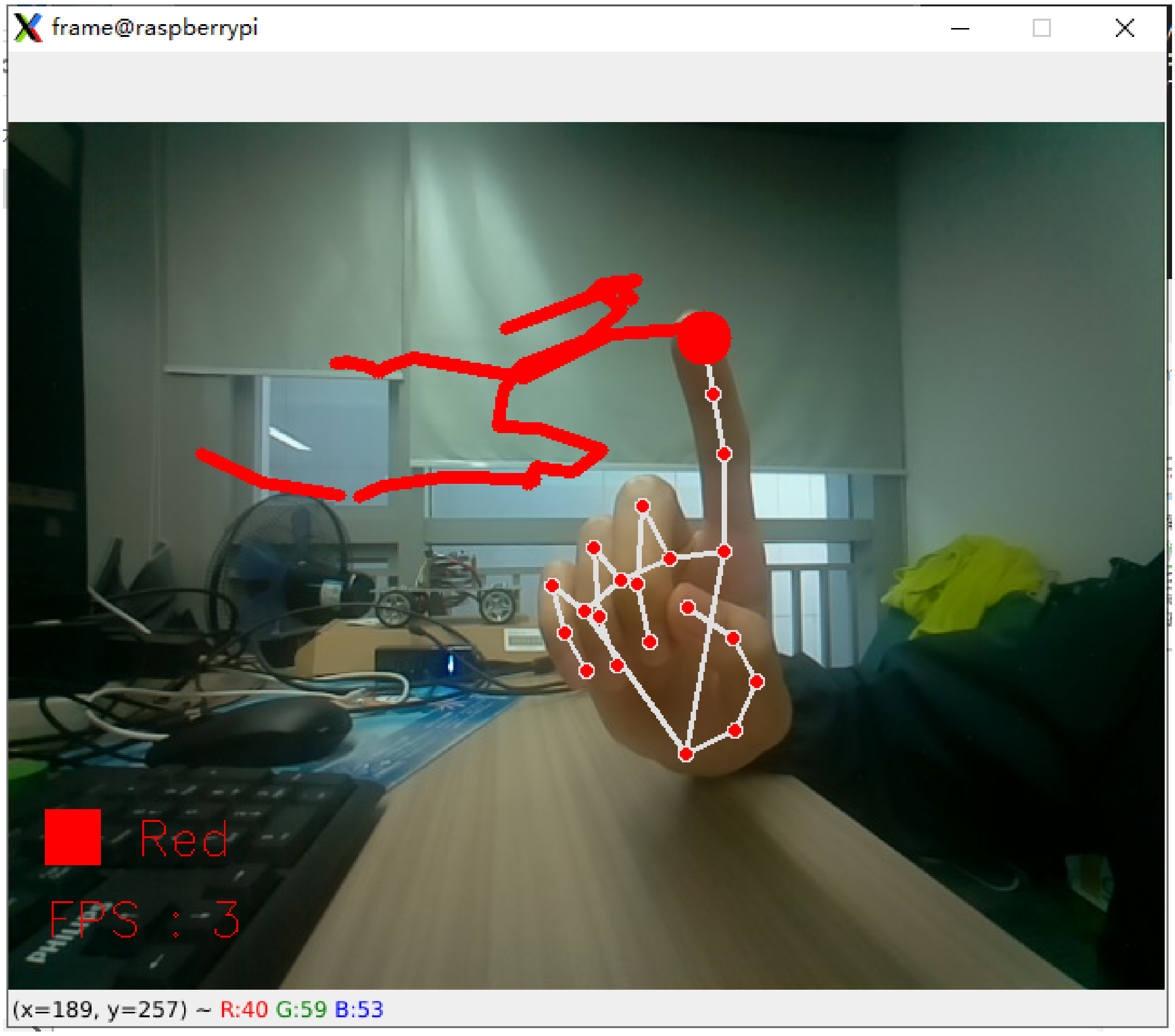
Enter the command in the terminal to directly start the python script

```
cd /home/pi/DOGZILLA_Lite_class/5.AI Visual Recognition Course/13. Brush
python3 VirtualPaint_USB.py
```

3. Experimental Phenomenon

After running the source code, you can see that the robot dog displays the camera image on the computer screen. Because the robot dog screen is too small, it cannot be displayed on the robot dog screen synchronously.





4. Main program source code analysis

```
if __name__ == '__main__':
    capture = cv.VideoCapture(0)
    capture.set(cv.CAP_PROP_AUTO_EXPOSURE, 0) #关闭自动曝光 Turn off automatic
exposure
    capture.set(6, cv.VideoWriter.fourcc('M', 'J', 'P', 'G'))
    capture.set(cv.CAP_PROP_FRAME_WIDTH, 640)
    capture.set(cv.CAP_PROP_FRAME_HEIGHT, 480)
    print("capture get FPS : ", capture.get(cv.CAP_PROP_FPS))
    hand_detector = handDetector(detectorCon=0.85)
    while capture.isOpened():
        ret, frame = capture.read()
        # frame = cv.flip(frame, 1)
        h, w, c = frame.shape
        frame,lmList = hand_detector.findHands(frame, draw=False)
        if len(lmList) != 0:
            # print(lmList)
            # tip of index and middle fingers
            x1, y1 = lmList[8][1:]
            x2, y2 = lmList[12][1:]
            fingers = hand_detector.fingersUp()
            if fingers[1] and fingers[2]:
                # print("Seclection mode")
                if y1 < top_height:
                    if 0 < x1 < int(w / 5) - 1:
                        boxx = 0
                        Color = "Red"
                    if int(w / 5) < x1 < int(w * 2 / 5) - 1:
                        boxx = int(w / 5)
                        Color = "Green"
                    elif int(w * 2 / 5) < x1 < int(w * 3 / 5) - 1:
                        boxx = int(w * 2 / 5)
                        Color = "Blue"
                    elif int(w * 3 / 5) < x1 < int(w * 4 / 5) - 1:
                        boxx = int(w * 3 / 5)
                        Color = "Yellow"
                    elif int(w * 4 / 5) < x1 < w - 1:
                        boxx = int(w * 4 / 5)
                        Color = "Black"
                cv.rectangle(frame, (x1, y1 - 25), (x2, y2 + 25),
ColorList[Color], cv.FILLED)
                cv.rectangle(frame, (boxx, 0), (boxx + int(w / 5), top_height),
ColorList[Color], cv.FILLED)
                cv.rectangle(frame, (0, 0), (int(w / 5) - 1, top_height),
ColorList['Red'], 3)
                cv.rectangle(frame, (int(w / 5) + 2, 0), (int(w * 2 / 5) - 1,
top_height), ColorList['Green'], 3)
                cv.rectangle(frame, (int(w * 2 / 5) + 2, 0), (int(w * 3 / 5) -
1, top_height), ColorList['Blue'], 3)
                cv.rectangle(frame, (int(w * 3 / 5) + 2, 0), (int(w * 4 / 5) -
1, top_height), ColorList['Yellow'], 3)
                cv.rectangle(frame, (int(w * 4 / 5) + 2, 0), (w - 1,
top_height), ColorList['Black'], 3)
            if fingers[1] and fingers[2] == False and math.hypot(x2 - x1, y2 -
y1) > 50:
                # print("Drawing mode")
                if xp == yp == 0: xp, yp = x1, y1
                if Color == 'Black':
                    cv.line(frame, (xp, yp), (x1, y1), ColorList[Color],
eraserThickness)
                    cv.line(imgCanvas, (xp, yp), (x1, y1), ColorList[Color],
eraserThickness)
                else:
                    cv.line(frame, (xp, yp), (x1, y1), ColorList[Color],
brushThickness)
                    cv.line(imgCanvas, (xp, yp), (x1, y1), ColorList[Color],
brushThickness)
                cv.circle(frame, (x1, y1), 15, ColorList[Color], cv.FILLED)
                xp, yp = x1, y1
            else: xp = yp = 0
        imgGray = cv.cvtColor(imgCanvas, cv.COLOR_BGR2GRAY)
        _, imgInv = cv.threshold(imgGray, 50, 255, cv.THRESH_BINARY_INV)
        imgInv = cv.cvtColor(imgInv, cv.COLOR_GRAY2BGR)
        frame = cv.bitwise_and(frame, imgInv)
        frame = cv.bitwise_or(frame, imgCanvas)
        if cv.waitKey(1) & 0xFF == ord('q'): break
        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime
        text = "FPS : " + str(int(fps))
        cv.rectangle(frame, (20, h - 100), (50, h - 70), ColorList[Color],
cv.FILLED)
```

```
        cv.putText(frame, Color, (70, h - 75), cv.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 1)
        cv.putText(frame, text, (20, h-30), cv.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 1)
        cv.imshow('frame', frame)

    capture.release()
    cv.destroyAllWindows()
```

From the source code analysis, we can know that the robot dog will call the camera to display the brush on the computer screen. Press the 'Q' key on the keyboard to exit the program.