

QR code identification

QR code identification

- 1. Introduction
- 2. Code analysis
- 1.3 Steps

1. Introduction

This course mainly uses the camera of Raspberry Pi to obtain the image of the camera, calls the pyzbar library to analyze the image, and analyzes the position and band information of the two-dimensional code in the image. According to the recognized information, if the action information is met, a corresponding action is made. The two-dimensional code image of motion recognition is in the data.

2. Code analysis

On the basis of two-dimensional code recognition, add control_Action: manage the actions of the robot dog.

```
# 解析图像中的二维码信息 Analyze the qrcode information in the image
def decodeDisplay(image, display):
    barcodes = pyzbar.decode(image)
    for barcode in barcodes:
        # 提取二维码的边界框的位置, 画出图像中条形码的边界框
        # Extract the position of the bounding box of the qrcode,
        # and draw the bounding box of the barcode in the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(display, (x, y), (x + w, y + h), (225, 225, 225), 2)

        # 提取二维码数据为字节对象, 转换成字符串
        # The qrcode data is extracted as byte objects and converted into strings
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type

        # 绘出图像上条形码的数据和条形码类型
        # Plot the barcode data and barcode type on the image
        text = "{} ({}).format(barcodeData, barcodeType)
        cv2.putText(display, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (225, 0, 0), 2)

    print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
    control_action(barcodeData)
    return display
```

```

# 根据识别到的命令做动作 Act on recognized commands
def control_action(info):
    if info == "LIE DOWN":
        g_dog.action(1) #躺下

    elif info == "STAND UP":
        g_dog.action(2) #站起

    elif info == "CRAWL":
        g_dog.action(3) #匍匐前进

    elif info == "TURN AROUND":
        g_dog.action(4) #转圈

    elif info == "MARK TIME":
        g_dog.action(5) #原地踏步

    elif info == "SQUAT":
        g_dog.action(6) #蹲起

```

Start a daemon thread, run the camera identification task, and display the camera image.

```

# 启动摄像头显示任务 Start the camera display task
thread1 = threading.Thread(target=Detect_Qrcode_Task)
thread1.setDaemon(True)
thread1.start()
|
box_display = widgets.HBox([image_widget, button_Close_Camera])
display(box_display)

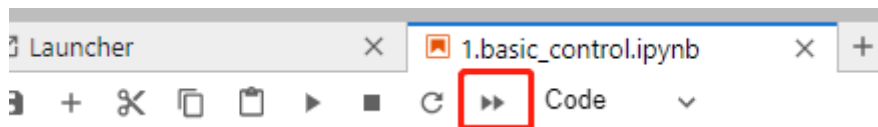
```

1.3 Steps

Open the jupyterLab client and find following code path

DOGZILLA/Samples/3_AI_Visual/8.QRCode_action.ipynb

Click the following icon to run all cells, and then pull to the bottom to see the generated controls.



The left side shows the camera screen. If the two-dimensional code is placed in front of the camera, the robot dog will recognize the two-dimensional code and display the recognized data on the two-dimensional code. If the data content of the two-dimensional code corresponds to the following table, it will make corresponding actions.