3. QR code creation and recognition

1 QR Code

1.1 QR Code Introduction

QR code is a kind of two-dimensional barcode, QR comes from the abbreviation of English "Quick Response", that is, the meaning of quick response, originated from the inventor hopes that the QR code can let its content be decoded quickly.QR code not only has large information capacity, high reliability and low cost, but also can represent a variety of textual information such as Chinese characters and images, its QR code not only has large information capacity, high reliability, low cost, but also can represent many kinds of text information such as Chinese characters and images, its confidentiality and anti-counterfeiting is strong and very convenient to use. What's more, QR code technology is open source.

1.2 Structure of QR Code

Picture	Parsing
	(Positioning markings) Indicate the direction of the QR code.
	(Alignment markings) If the QR code is large, these additional elements help with positioning.
	(Timing pattern) From these lines, the scanner can identify how large the matrix is.
	(Version information) This specifies the version number of the QR code being used, there are currently 40 different versions of QR codes. The version numbers used in the sales industry are usually 1-7.
	(Format information) The format pattern contains information about fault tolerance and data mask patterns, and makes it easier to scan code.
	(Data and error correction keys) These schemas hold the actual data.
	(Quiet zone) This area is very important for the scanner, its role is to separate itself from the surrounding.

1.3. Characteristics of QR codes

Data values in QR codes contain repeated information (redundant values). Therefore, even up to 30% of the QR code structure is destroyed without affecting the readability of the QR code.QR codes have a storage space of up to 7,089 bits or 4,296 characters, including punctuation marks and special characters, which can be written into the QR code. In addition to numbers and characters, words and phrases (such as web addresses) can be encoded. As more data is added to the QR code, the code size increases and the code structure becomes more complex.

1.4, QR code creation and recognition

1) Source code path

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_visual/simple_qrcode
```

2) Installation package

```
python3 -m pip install qrcode pyzbar
sudo apt-get install libzbar-dev
```

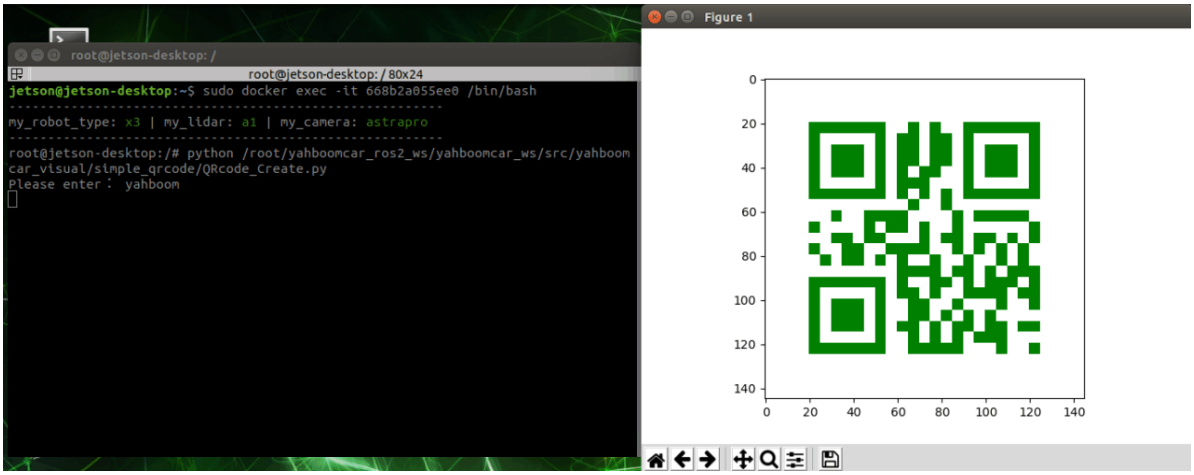
The factory docker image is already installed.

3) Create QRcode_Create.py

Go to docker, open a terminal and type.

```
python
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_visual/simple_qrcode/QRcod
e_Create.py
```

After the program is run, you will be prompted to enter the generated content, and the Enter key will confirm the content. Here we create the string "yahboom" as an example.



The QR code appears on the right side, take out your cell phone and try to scan it, the result will be yahboom characters.

Source code analysis.

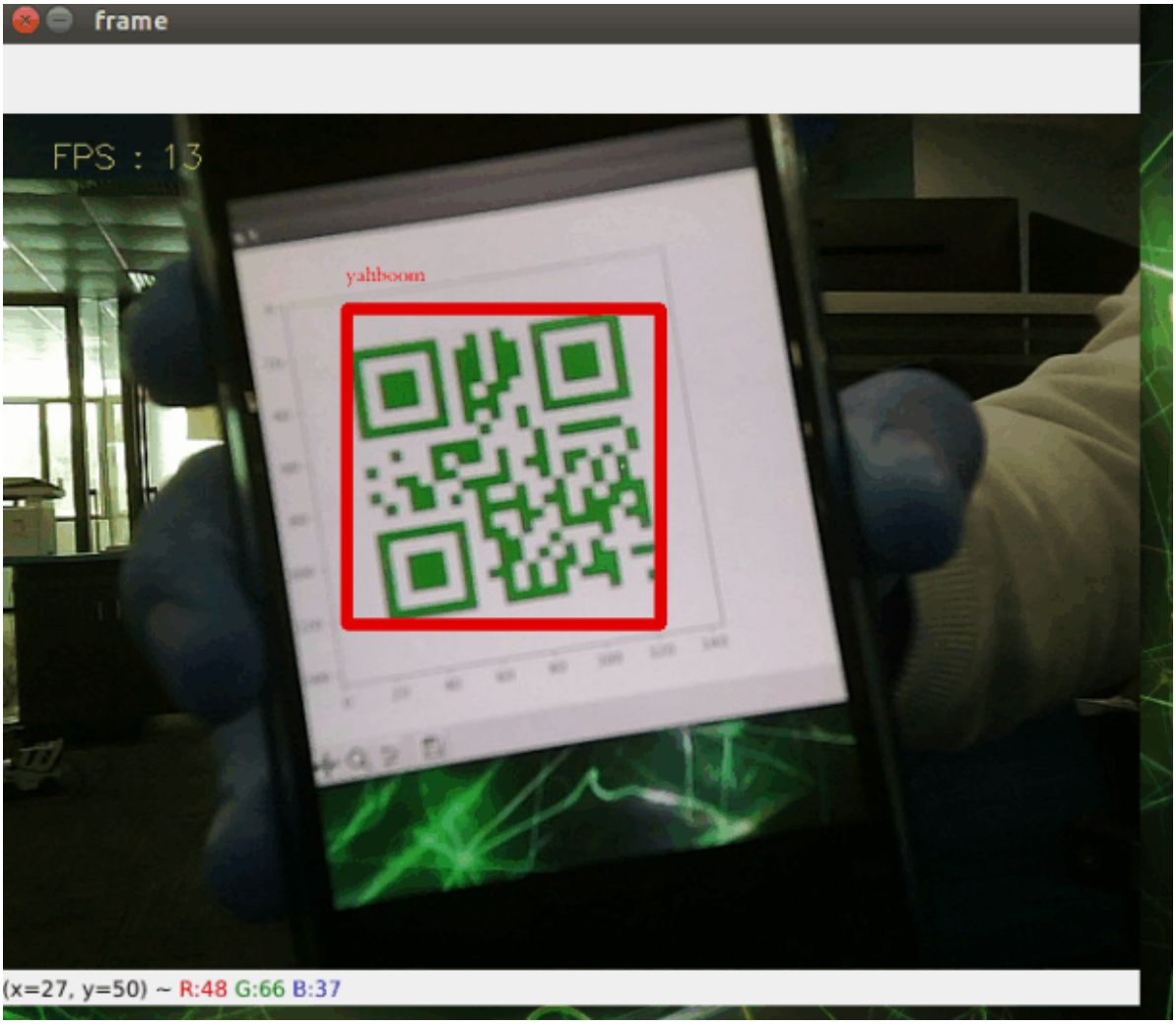
```
#创建qrcode对象 # Create a qrcode object
qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_H,
        box_size=5,
        border=4,)
#各参数含义
'''version：值为1~40的整数，控制二维码的大小（最小值是1，是个12×12的矩阵）。
              如果想让程序自动确定，将值设置为 None 并使用 fit 参数即可。
error_correction：控制二维码的错误纠正功能。可取值下列4个常量。
        ERROR_CORRECT_L：大约7%或更少的错误能被纠正。
        ERROR_CORRECT_M（默认）：大约15%或更少的错误能被纠正。
        ROR_CORRECT_H：大约30%或更少的错误能被纠正。
box_size：控制二维码中每个小格子包含的像素数。
```

```
border: 控制边框（二维码与图片边界的距离）包含的格子数（默认为4，是相关标准规定的最小值）'
'version: value is an integer from 1 to 40, which controls the size of the QR
code (the minimum value is 1, which is a 12×12 matrix).
             If you want the program to determine it automatically, set the value
to None and use the fit parameter.
error_correction: control the error correction function of the QR code. It can
take the following 4 constants.
        ERROR_CORRECT_L: about 7% or less of errors can be corrected.
        ERROR_CORRECT_M (default): about 15% or less of errors can be corrected.
        ROR_CORRECT_H: about 30% or less of errors can be corrected.
box_size: controls the number of pixels contained in each cell of the 2D code.
border: control the number of cells contained in the border (the distance between
the QR code and the image boundary) (default is 4, which is the minimum value
specified by the relevant standard).''''
#qrcode二维码添加logo #qrcode QR code to add logo
my_file = Path(logo_path)
if my_file.is_file(): img = add_logo(img, logo_path)
#添加数据 #Add data
qr.add_data(data)
# 填充数据
# fill data
qr.make(fit=True)
# 生成图片
# generate images
img = qr.make_image(fill_color="green", back_color="white")
```

4) Recognize QRcode_Parsing.py

Go to docker, open a terminal and type.

```
python
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_visual/simple_qrcode/QRcod
e_Parsing.py
```



After the program runs, we place the QR code in front of the camera, the program will recognize the content of the QR code, mark it on the picture and print out the recognized content in the terminal.

Source code analysis.

```
def decodeDisplay(image, font_path):
    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    # 需要先把输出的中文字符转换成Unicode编码形式
    # Need to convert the output Chinese characters to Unicode first
    barcodes = pyzbar.decode(gray)
    for barcode in barcodes:
    # 提取二维码的边界框的位置
    # Extract the location of the bounding box of the QR code
    (x, y, w, h) = barcode.rect
    # 画出图像中条形码的边界框
    # Draw the bounding box of the barcode in the image
    cv.rectangle(image, (x, y), (x + w, y + h), (225, 0, 0), 5)
    encoding = 'UTF-8'
    # 画出来，就需要先将它转换成字符串
     # To draw it, you need to convert it to a string first
    barcodeData = barcode.data.decode(encoding)
    barcodeType = barcode.type
    # 绘出图像上数据和类型
    # Plot the data and types on the image
    pilimg = Image.fromarray(image)
    # 创建画笔 # Create brushes
    draw = ImageDraw.Draw(pilimg)
```

```
    # 参数1：字体文件路径，参数2：字体大小
      # Parameter 1: font file path, parameter 2: font size
    fontStyle = ImageFont.truetype(font_path, size=12, encoding=encoding)
    # 参数1：打印坐标，参数2：文本，参数3：字体颜色，参数4：字体
     # Parameter 1: print coordinates, parameter 2: text, parameter 3: font
color, parameter 4: font
    draw.text((x, y - 25), str(barcode.data, encoding), fill=(255, 0,
0),font=fontStyle)
    # PIL图片转cv2 图片
     # PIL images to cv2 images
    image = cv.cvtColor(np.array(pilimg), cv.COLOR_RGB2BGR)
    # 向终端打印条形码数据和条形码类型
      # Print bar code data and bar code type to terminal
    print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
    return image
```