3.1.5 QR code recognition

Introduction to QR Code

QR code uses a certain geometric figure to record data symbol information in a black and white pattern distributed on a plane (two-dimensional direction) according to a certain rule.

Code path:

/home/pi/Yahboom_Project/3.AI_Visual_course/05.QR_code_recognition.ipynb

```
#bgr8 to jpeg format
import enum
import cv2

def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
    # Import library and show camera display component
# import the necessary packages
#import simple_barcode_detection
import cv2
import numpy as np
import pyzbar.pyzbar as pyzbar
from PIL import Image
import ipywidgets.widgets as widgets

image_widget = widgets.Image(format='jpeg', width=320, height=240)
display(image_widget)                    # Display camera component
```

```
# Define the parse QR code interface
def decodeDisplay(image):
    barcodes = pyzbar.decode(image)
    for barcode in barcodes:
        # Extract the position of the bounding box of the QR code
  # Draw the bounding box of the barcode in the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(image, (x, y), (x + w, y + h), (225, 225, 225), 2)
        # Extract the QR code data as a byte object, so if we want to output the
image, you need to convert it to a string
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type

        # Draws the data and barcode type of the barcode on the image
        text = "{} ({})".format(barcodeData, barcodeType)
        cv2.putText(image, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,0.5, (225,
225, 225), 2)

        # Print the data and barcode type of the barcode on the terminal
```

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

def detect():
        camera = cv2.VideoCapture(0)
        camera.set(3, 320)
        camera.set(4, 240)
        camera.set(5, 120)    # Set frame rate
        # fourcc = cv2.VideoWriter_fourcc(*"MPEG")
        camera.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter.fourcc('M', 'J', 'P', 'G'))
        camera.set(cv2.CAP_PROP_BRIGHTNESS, 40) #Set brightness -64 - 64    0.0
        camera.set(cv2.CAP_PROP_CONTRAST, 50) #Set contrast -64 - 64    2.0
        camera.set(cv2.CAP_PROP_EXPOSURE, 156) #Set exposure 1.0 - 5000    156.0
        ret, frame = camera.read()
        image_widget.value = bgr8_to_jpeg(frame)
        while True:
            # Read frame currently
            ret, frame = camera.read()
            # To Grayscale image
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            im = decodeDisplay(gray)

            cv2.waitKey(5)
            image_widget.value = bgr8_to_jpeg(im)
            # If you press q, you will be out of the loop
            if cv2.waitKey(10) & 0xFF == ord('q'):
                break
        camera.release()
        cv2.destroyAllWindows()
```

```
while 1:
        detect()
```

After run above program, put a QR code front the camera, it will be recognized.

```
[*]: while 1:
         detect()

     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
     [INFO] Found QRCODE barcode: Yahboom
```

We can click the pause button to stop the process. As shown below.