4.1.3 QR code recognition+Voice broadcast

Combining the two functions of QR code recognition and voice broadcast.
The whole process is as follows:
1)The camera collects pictures
2)Send the picture to the QR code for recognition
3)Synthesize the recognition result and broadcast it

Code path:
/home/pi/Yahboom_Project/4.AI Voice course/ 02.QR_code_voice.ipynb

```
# Import the library for speech synthesis and broadcast
import time
import pygame
from aip import AipSpeech

# The following key should be replaced with your own key
""" Voice technology APPID AK SK """
SpeechAPP_ID = '17852430'
SpeechAPI_KEY ='eGeO4iQGAjHCrzBTYd1uvTtf'
SpeechSECRET_KEY = 'Cn1EVsUngZDbRLv4OxAFrDHSo8PsvFVP'

# Connect client
Speechclient = AipSpeech(SpeechAPP_ID, SpeechAPI_KEY, SpeechSECRET_KEY)

# Voice broadcast initialization
pygame.mixer.init()

def AudioPlay(text):
    result = Speechclient.synthesis(text, 'zh', 1, {'spd': 2, 'vol': 2, 'per': 1})

    if not isinstance(result, dict):
        with open('./02.mp3', 'wb') as f:
            f.write(result)
        pygame.mixer.init()
        pygame.mixer.music.load('./02.mp3')
        pygame.mixer.music.play()
        time.sleep(2)
```

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

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

```
# Import QR code recognition library and display camera display components
```

```
# 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 components
```

```
# Define parsing QR code interface
def decodeDisplay(image):
    barcodes = pyzbar.decode(image)
    for barcode in barcodes:
        # Extract the location of the QR code bounding box
        # Draw the bounding box for the bar code in the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(image, (x, y), (x + w, y + h), (225, 225, 225), 2)

        # To extract the QR code data as a byte object, we need to convert it to a
string first, and then draw it on the output image
        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)

        # Prints the barcode data and barcode type on the terminal
        print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
        AudioPlay(barcodeData)
    return image

def detect():
    camera = cv2.VideoCapture(0)
    camera.set(3, 320)
    camera.set(4, 240)
    camera.set(5, 120)    #Set the 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 current frame
        ret, frame = camera.read()
        # Convert to grayscale image
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        im = decodeDisplay(gray)
        image_widget.value = bgr8_to_jpeg(im)
        # If you press q on keyboard, it will jump out of this loop
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    camera.release()
    cv2.destroyAllWindows()
```

```
#Main process
while 1:
    detect()
```

After running the above program, put the QR code in front of the camera, we can see that the QR code will be recognized, and the voice broadcast.

```
while 1:
    detect()

[INFO] Found QRCODE barcode: left
[INFO] Found QRCODE barcode: left
[INFO] Found QRCODE barcode: left
[INFO] Found QRCODE barcode: forward
[INFO] Found QRCODE barcode: forward
[INFO] Found QRCODE barcode: forward
[INFO] Found QRCODE barcode: forward
[INFO] Found QRCODE barcode: forward
[INFO] Found QRCODE barcode: forward
```