7. License Plate Recognition

7.1. Overview

For license plate recognition, we use HyperLPR3, a high-performance open source Chinese license plate recognition framework. It supports rapid deployment of WebApi services, multiple license plates, and double-layer license plates. It supports large-angle license plate recognition and lightweight recognition models.

Features:

Fast recognition speed, with an average recognition time of less than 100ms for a single-core 2.2G CPU

End-to-end license plate recognition without character segmentation

High recognition rate, with a recognition accuracy of about 95%-97% in checkpoint scenes

Supports cross-platform compilation and rapid deployment

7.2. Code implementation

Image-based license plate recognition

After running the program, we can see the recognition results below the program. At the same time, the oled will also display the recognition results

Source code path:

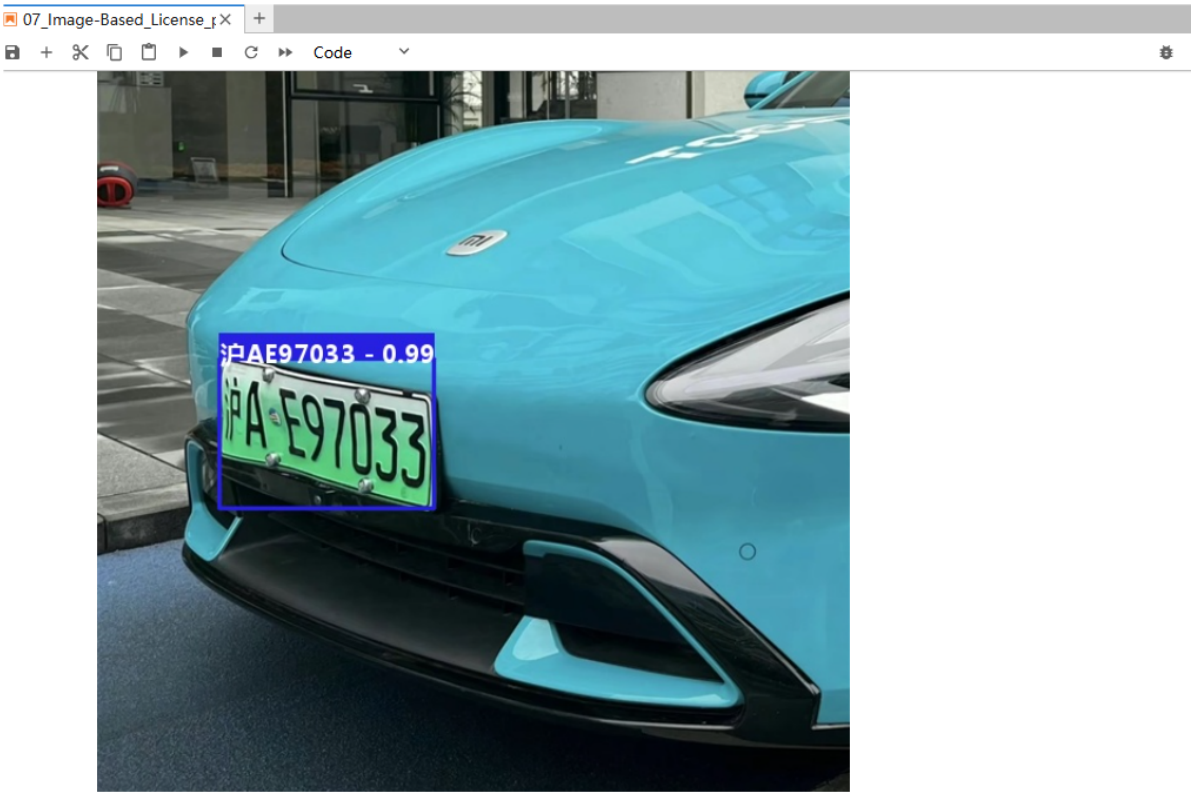
/home/pi/project_demo/07.AI_Visual_Recognition/07.License_plate_recognition/07.Image-Based_License_plate_recognition/07_Image-Based_License_plate_recognition.ipynb

```
# 导入oled屏幕库 Import oled screen library
import sys
sys.path.append('/home/pi/software/oled_yahboom/')
from yahboom_oled import *
# 创建oled对象 Create an oled object
oled = Yahboom_OLED(debug=False)
# 导入组件 Importing Components
import ipywidgets.widgets as widgets
image_widget = widgets.Image(format='jpeg', width=640, height=640)  #设置摄像头显示组件 Set up the camera display component
display(image_widget)
```

```
# 导入cv相关库 Import cv related libraries
import cv2
import numpy as np
from PIL import ImageFont
from PIL import Image
```

```
from PIL import ImageDraw
# 导入依赖包 Import dependency packages
import hyperlpr3 as lpr3

# 将BGR图像转换为JPEG格式的字节流 Convert a BGR image to a JPEG byte stream
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])

# 在图像上绘制车牌框及文字 Draw the license plate frame and text on the image
def draw_plate_on_image(img, box, text, font):
    x1, y1, x2, y2 = box
    cv2.rectangle(img, (x1, y1), (x2, y2), (225, 32, 39), 2, cv2.LINE_AA)
    cv2.rectangle(img, (x1, y1 - 20), (x2, y1), (225, 32, 39), -1)
    data = Image.fromarray(img)
    draw = ImageDraw.Draw(data)
    draw.text((x1 + 1, y1 - 18), text, (255, 255, 255), font=font)
    res = np.asarray(data)
    return res


# 中文字体加载 Chinese font loading
font_ch = ImageFont.truetype("platech.ttf", 20, 0)

# 实例化识别对象 Instantiate the recognition object
catcher =
lpr3.LicensePlateCatcher(detect_level=lpr3.DETECT_LEVEL_LOW)#DETECT_LEVEL_HIGH64
0*640
# 读取图片 Reading pictures
image = cv2.imread("沪AE97033.png")
try:
    oled.init_oled_process() #初始化oled进程 Initialize oled process
    # 执行识别算法 Execute the recognition algorithm
    results = catcher(image)
    for code, confidence, type_idx, box in results:
        # 解析数据并绘制 Parsing and plotting data
        text = f"{code} - {confidence:.2f}"
        image = draw_plate_on_image(image, box, text, font=font_ch)
    oled.clear()
    carcher_str=f'carcher : {code}'
    confidence_str=f'confidence:{confidence:.2f}'
    oled.add_cnline(carcher_str, 1)
    oled.add_line(confidence_str, 3)
    oled.refresh()
    image_widget.value = bgr8_to_jpeg(image)
    print(text)
    # 显示检测结果 Display test results
    # cv2.imshow("LicensePlate", image)
    # cv2.waitKey(0)
except KeyboardInterrupt:
    cv2.destroyAllWindows()
```

```
# 结束 Finish
# 恢复屏幕基础数据显示 Restore basic data display on screen
os.system("python3 /home/pi/software/oled_yahboom/yahboom_oled.py &")
```




```
[2]: # 导入cv相关库 Import cv related libraries
import cv2
import numpy as np
from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw
```

Real-time license plate recognition based on camera

After running the program, we can see the recognition results below the program. At the same time, the oled will also display the recognition results

Source code path:

/home/pi/project_demo/07.AI_Visual_Recognition/07.License_plate_recognition/07.Camera-Based_License_plate_recognition/07_Camera-Based_License_plate_recognition.ipynb

```
# 导入oled屏幕库 Import oled screen library
import sys
sys.path.append('/home/pi/software/oled_yahboom/')
from yahboom_oled import *
# 创建oled对象 Create an oled object
oled = Yahboom_OLED(debug=False)
# 导入组件 Importing Components
import ipywidgets.widgets as widgets
image_widget = widgets.Image(format='jpeg', width=640, height=640)  #设置摄像头显示组件  Set up the camera display component

# 导入cv相关库 Import cv related libraries
import cv2
import numpy as np
from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw
# 导入依赖包 Import dependency packages
import hyperlpr3 as lpr3
```

```
# 将BGR图像转换为JPEG格式的字节流 Convert a BGR image to a JPEG byte stream
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])

# 在图像上绘制车牌框及文字 Draw the license plate frame and text on the image
def draw_plate_on_image(img, box, text, font):
    x1, y1, x2, y2 = box
    cv2.rectangle(img, (x1, y1), (x2, y2), (225, 32, 39), 2, cv2.LINE_AA)
    cv2.rectangle(img, (x1, y1 - 20), (x2, y1), (225, 32, 39), -1)
    data = Image.fromarray(img)
    draw = ImageDraw.Draw(data)
    draw.text((x1 + 1, y1 - 18), text, (255, 255, 255), font=font)
    res = np.asarray(data)
    return res
```

```
try:
    code=0
    confidence=0
    type_idx=0
    box=0
    image=0
    display(image_widget)
    # 中文字体加载 Chinese font loading
    font_ch = ImageFont.truetype("platech.ttf", 20, 0)
    # 实例化识别对象 Instantiate the recognition object
    catcher =
lpr3.LicensePlateCatcher(detect_level=lpr3.DETECT_LEVEL_LOW)#DETECT_LEVEL_HIGH64
0*640
    oled.init_oled_process() #初始化oled进程 Initialize oled process

    camera = cv2.VideoCapture(0)      # 定义摄像头对象，参数0表示第一个摄像头，默认
640x480 Define the camera object. Parameter 0 indicates the first camera. The
default resolution is 640x480.
    camera.set(3, 320)
    camera.set(4, 240)
    pTime, cTime = 0, 0
    while True:
        ret, frame = camera.read()
        # 执行识别算法
        results = catcher(frame)
        # 计算帧率
        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime
        text = "FPS : " + str(int(fps))
        cv2.putText(frame, f"FPS: {fps:.1f}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
        # 初始化图像变量 Initialize image variables
        image = frame.copy()  # 使用原始帧作为默认图像 Use original frame as default
image
        for code, confidence, type_idx, box in results:
                text = f"{code} - {confidence:.2f}"
                image = draw_plate_on_image(frame, box, text, font=font_ch)
        oled.clear()
```

```
        if results and len(results) > 0:
            code, confidence, _, _ = results[0]
            carcher_str = f'carcher : {code}'
            confidence_str = f'confidence: {confidence:.2f}'
            oled.add_cnline(carcher_str, 1)
            oled.add_line(confidence_str, 3)
        else:
            oled.add_cnline('carcher : 粤B',1)
            oled.add_line('confidence: 0',3)
        oled.refresh()

        image_widget.value = bgr8_to_jpeg(image)
        # cv2.imshow('frame', frame)
        cher_list = results[0] if results and results[0] is not None else None
        if cher_list is not None:
            print(cher_list)
        # if cv2.waitKey(1) & 0xFF == ord('q'):
        #     break

except KeyboardInterrupt:
    # picam2.stop()
    # picam2.close()
    camera.release()
    # 恢复屏幕基础数据显示 Restore basic data display on screen
    os.system("python3 /home/pi/software/oled_yahboom/yahboom_oled.py &")
```

```
#最后需要释放掉摄像头的占用 Finally, you need to release the camera's occupancy
camera.release()
oled.init_oled_process() #初始化oled进程 Initialize oled process
# 恢复屏幕基础数据显示 Restore basic data display on screen
os.system("python3 /home/pi/software/oled_yahboom/yahboom_oled.py &")
# picam2.stop()
# picam2.close()
#最后需要释放掉摄像头的占用 Finally, you need to release the camera's occupancy
```

07_Camera-Based_License

Code

```
        oled.refresh()

        image_widget.value = bgr8_to_jpeg(image)
        # cv2.imshow('frame', frame)
        cher_list = results[0] if results and results[0] is not None else None
        if cher_list is not None:
            print(cher_list)
        # if cv2.waitKey(1) & 0xFF == ord('q'):
        #     break

except KeyboardInterrupt:
    # picam2.stop()
    # picam2.close()
    camera.release()
    # 恢复屏幕基础数据显示 Restore basic data display on screen
    os.system("python3 /home/pi/software/oled_yahboom/yahboom_oled.py &")
```