

1.1 HyperLPR3

1.1.1 任务 1-HyperLPR3 介绍

HyperLPR3 是一个高性能开源中文车牌识别框架，是基于 Python 的深度学习实现，用于中文车牌的识别。与开源的 EasyPR 相比，HyperLPR3 在检测速度、鲁棒性和多场景的适应性方面都有更好的表现。

技术特点：

支持多种车牌类型：包括单行蓝牌、单行黄牌、新能源车牌、教练车牌、白色警用车牌、使馆/港澳车牌、双层黄牌、武警车牌等。

支持 Windows、Mac、Linux、Android、IOS 等平台。

除了 Python 外，还支持 PHP、C/C++ 等语言。

在 720p 分辨率下，使用单核 Intel 2.2G CPU 的识别时间平均低于 100ms。

在卡口场景下的识别准确率在 95%-97% 左右。

模型资源：

HyperLPR3 使用了多种深度学习模型，如 Faster R-CNN 和 CRNN 等，以及一系列预训练好的模型资源，如 cascade.xml（检测模型）、char_chi_sim.h5（字符识别模型）等。

功能和应用：

能够准确检测和识别车牌信息。

在各种复杂环境条件下（如光照变化、角度偏移）都能保持较高的识别准确率。

应用场景主要包括智能交通管理、停车场管理系统、安防监控、物流追踪、车载导航系统等。

优势：

通过深度学习和优化技术，实现了快速而准确的车牌识别。

提供了丰富的文档和示例代码，降低了开发门槛。

能够在多种操作系统和编程语言下运行。

遵循 MIT 许可证，鼓励社区参与和协作改进。

综上所述，HyperLPR3 是一个功能强大、性能卓越、易于使用的车牌识别工具，为智能交通等领域的发展提供了有力的技术支撑。

1.1.2 任务 2-HyperLPR3 安装

HyperLPR3 的安装非常简单，可以通过 Python 的 pip 工具直接进行安装。

提供了详尽的文档和示例代码，方便开发者进行二次开发和集成。

支持使用命令行工具对本地图像或在线 URL 进行快速测试。

HyperLPR - 基于深度学习高性能中文车牌识别，码云地址：

```
https://gitee.com/mirrors/HyperLPR
```

快速安装命令：

```
pip install hyperlpr3
```

```
pip install pillow
```

支持的车牌类别：

单行蓝牌

单行黄牌

新能源车牌

教练车牌

白色警用车牌

使馆/港澳车牌

双层黄牌

武警车牌

快速测试：

安装成功后，可以使用命令行工具对本地图像或在线 url 进行快速测试，这边我们使用一张新能源的车辆照片进行测试：

```
lpr3 sample -src 33.jpg
```

运行结果：



```
(venv) PS D:\PycharmProjects\hyperlpr3Project> lpr3 sample -src 33.jpg
-----
2024-06-03 15:50:32.975 | INFO      | hyperlpr3.command.sample:sample:70 - 共检测到车牌: 1
2024-06-03 15:50:32.985 | SUCCESS   | hyperlpr3.command.sample:sample:73 - [绿牌新能源]皖ADB4566 0.961225152
015686 [230, 520, 404, 575]
```

1.1.3 任务 3-图片的车牌识别

示例代码如下：

```
import cv2

import warnings

import numpy as np

from PIL import ImageFont

from PIL import Image

from PIL import ImageDraw

import hyperlpr3 as lpr3

# 辅助函数

def draw_plate_on_image(img, box1, text1, font):

    x1, y1, x2, y2 = box1 # 识别框的四至范围

    # random_color = (random.randint(0, 255), random.randint(0, 255),
    random.randint(0, 255))

    cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255), 2, cv2.LINE_AA) # 车牌外框

    # cv2.rectangle(img, (x1, y1 - 25), (x2, y1-3), (139, 139, 102), -1) # 识别文本
    底色

    data = Image.fromarray(img) # 读取图片

    draw = ImageDraw.Draw(data) # PIL 绘制图片

    draw.text((x1, y1 - 27), text1, (0, 0, 255), font=font) # 添加识别文本

    res = np.asarray(data) # 返回叠加识别结果的图片

    return res
```

```

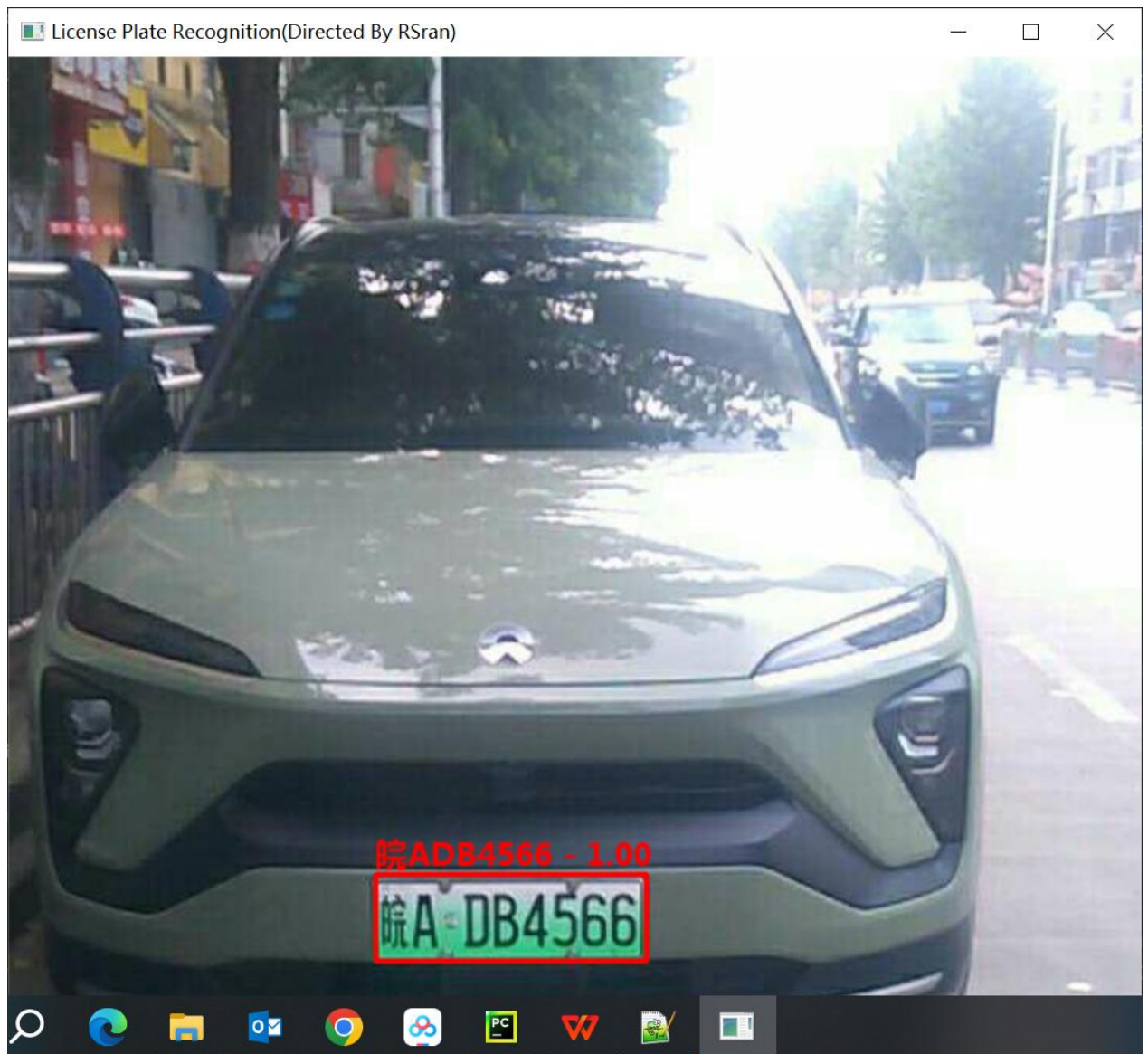
# 图片识别

def license_recognition_image(path):
    image = cv2.imread(path) # 读取图片
    results = catcher(image) # 执行识别算法
    for code, confidence, type_idx, box in results:
        # [['京 Q58A77', 0.9731929, 0, [150, 160, 451, 259]]]
        text = f"{code} - {confidence:.2f}"
        image = draw_plate_on_image(image, box, text, font=font_ch) # 绘制识
别结果
    cv2.imshow("License Plate Recognition(Directed By RSran)", image) # 显示检
测结果
    cv2.waitKey(0)

if __name__ == "__main__":
    warnings.filterwarnings("ignore", message="Mean of empty slice") # 忽略
"Mean of empty slice"的警告
    warnings.filterwarnings("ignore", message="invalid value encountered in scalar
divide")
    # 忽略"invalid value encountered in scalar divide"的警告
    catcher = lpr3.LicensePlateCatcher(detect_level=lpr3.DETECT_LEVEL_HIGH) #
实例化识别对象
    font_ch = ImageFont.truetype("platech.ttf", 20, 0) # 中文字体加载
    # 图像识别
    file_pic = r"33.jpg"
    license_recognition_image(file_pic)

```

运行结果如下：



1.1.1.4 任务 4-视频的车牌识别

示例代码如下：

```
import cv2
import warnings
import numpy as np
from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw
import hyperlpr3 as lpr3
```



```

        image = draw_plate_on_image(image, box, text,
font=font_ch) # 绘制识别结果

        cv2.imshow("License Plate Recognition(Video)", image) # 显示检
测结果

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break # 退出

    else:
        break

if __name__ == "__main__":
    warnings.filterwarnings("ignore", message="Mean of empty slice") # 忽略
"Mean of empty slice"的警告

    warnings.filterwarnings("ignore", message="invalid value encountered in scalar
divide")

    # 忽略"invalid value encountered in scalar divide"的警告

    catcher = lpr3.LicensePlateCatcher(detect_level=lpr3.DETECT_LEVEL_HIGH) #
实例化识别对象

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

    # 视频识别

    file_video = r"1.mp4"

    license_recognition_video(file_video)

```

运行结果如下：



视频车牌识别.wmv

1.1.5 任务 5-启动在线 API 服务

如果有部署到云端去调用的需求，HyperLPR3 中已经内置了启动 WebApi 服务的功能，支持一键启动，且自带 SwaggerUI 文档页面，相对方便友好：

```

# 启动服务 workers 为进程数量,请根据需求进行调节

lpr3 rest --port 8715 --host 0.0.0.0 --workers 1

```

启动后可打开 SwaggerUI 的路径：<http://localhost:8715/api/v1/docs> 查看和测试在线识别 API 服务：

HyperLPR3-Api 0.0.9 OAS3
[/openapi.json](#)
HyperLPR3 Api Serving

default

GET / Running

车牌识别

POST /api/v1/rec Vehicle License Plate Recognition

启动后即可对车牌识别 Api 进行使用。

示例代码如下：

```
import requests

# 设置你的 API 端点 URL
url = 'http://localhost:8715/api/v1/rec'

# 图片文件路径
file_path = './33.jpg'

# 准备文件以供上传
with open(file_path, 'rb') as f:
    files = {'file': (file_path, f.read(), 'image/jpeg')} # 文件名、文件内容、MIME 类型

# 发送 POST 请求
response = requests.post(url, files=files)

# 检查响应
if response.status_code == 200:
    print('图片上传成功！')
```



```
# 这里可以处理返回的数据，比如打印出来或保存到文件

print(response.json())

else:

    print(f'图片上传失败，状态码: {response.status_code}')

    # 这里可以处理错误，比如打印出错误信息

    print(response.json())
```

运行结果如下：

```
图片上传成功！
{
  'result': {
    'plate_list': [
      {
        'code': '皖 ADB4566',
        'conf': 0.9965810775756836,
        'plate_type': '绿牌新能源',
        'box': [
          233,
          519,
          405,
          574
        ]
      }
    ]
  },
  'code': 5000,
  'msg': '请求成功'
}
```