

OCR 识别系统说明手册

目录

系统构成.....	2
服务器端.....	2
客户端.....	2
系统测试展示.....	3
代码地址.....	4

系统构成

由于考虑展示的可移植性，整个系统分为服务器端和客户端两个部分，识别算法部分会占用较大的硬件资源，因而考虑布局在服务器端；客户端负责通过摄像头获取要展示的图片，将图片通信至服务器端获取识别结果，展示服务器端的识别结果。两者通过 TCP socket 进行通信。

服务器端

功能 服务器端的功能为处理客户端发送的图片并进行 OCR 端到端识别，将识别结果返回客户端进行展示。

实现 TCP 服务接收客户端上传的图片，将图片送入识别部分检测算法，检测部分采用 Mask R-CNN 提取图片中的字符区域，经过简单的尺寸整理，送入识别部分的 GRCNN 算法，将每个区域中的字符串提取出来。

部署环境

系统	Linux 17.04
显卡环境	CUDA9 + Cudnn7
深度学习开发环境	Pytorch 1.0
GCC 编译器	>= 4.9
语言	Python3 >= 3.6

安装 参考：https://github.com/mjq11302010044/RRPN_pytorch/README.md

使用 架设服务器直接使用命令：(包括 GPU 的指定)

```
CUDA_VISIBLE_DEVICES=GPU_ID python3 demo/demo_server.py
```

客户端

功能 客户端向服务器端发送图片，供服务器端识别，接收服务器端产生的识别结果并进行可视化。

实现 需要一台具有摄像头的计算机，通过 opencv 调取摄像头画面，将截取的摄像头画面通过 TCP 通信传到服务器端，等待服务器端识别完毕之后接收识别结果进行展示。展示内容包括提取的字符区域，以及字符区域对应的识别结果。

部署环境

系统	Linux/Windows
显卡环境	无
摄像头	可调取
GCC 编译器	无
语言	Python3 >= 3.6

使用

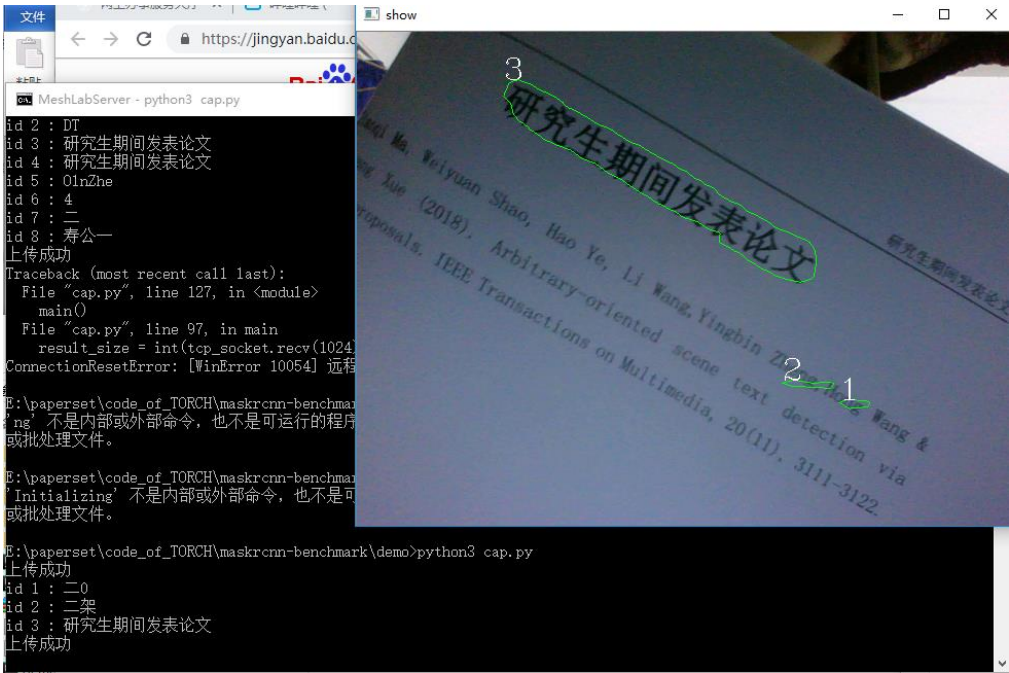
```
python3 cap.py
```

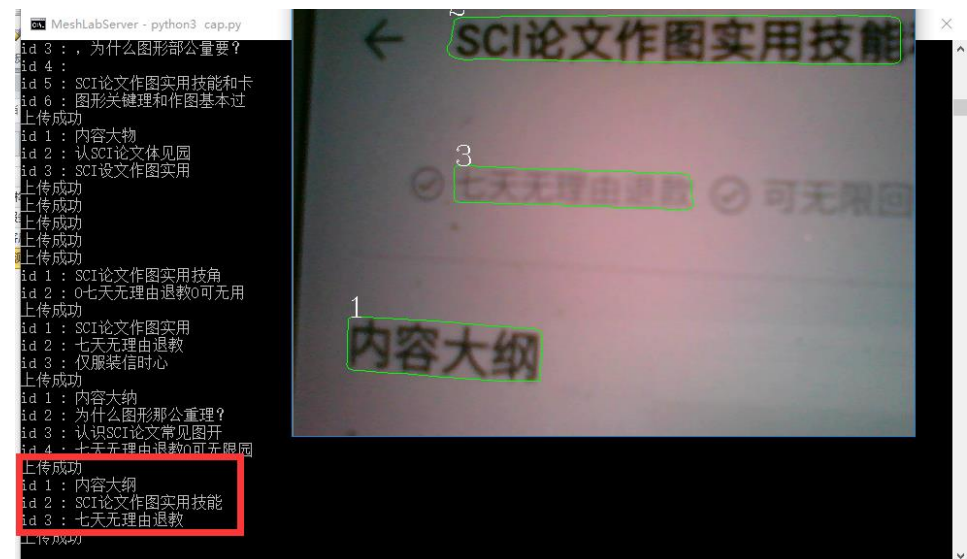
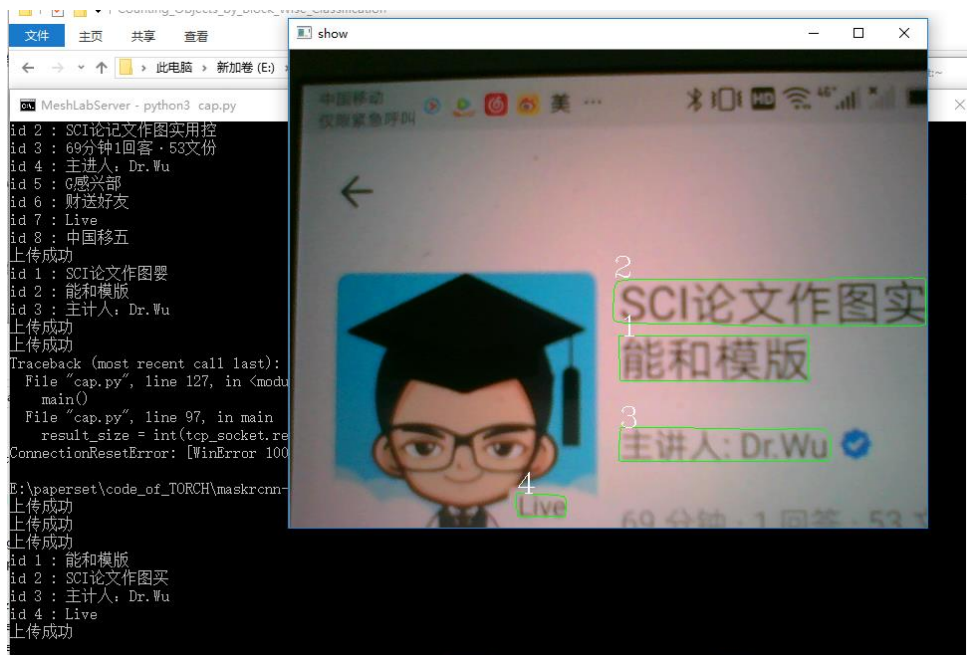
系统测试展示

网络环境 保证服务器和客户端在同一个网络内，可以采用 ping 命令进行测试。

网络设置 服务器 IP 和端口：10.141.209.43:9992 (端口和 IP 都可以更改)

测试信息输出





代码地址

在 43 服务器上 mjq 账号下: mjq@10.141.209.43:~/workspace/maskrcnn_benchmark/
SSH 密码: mjq