## 1环境要求：

* **基本要求：**

**python3.8.19 ，3.8应该都适用，服务器和windows都是3.8**。

cuda是10.2，不涉及cuda版本问题的话11的也可以。

其中服务器上python的依赖包使用下面命令安装 pip install -r requirement.txt ，其中requirement.txt在软件包的顶层目录中。

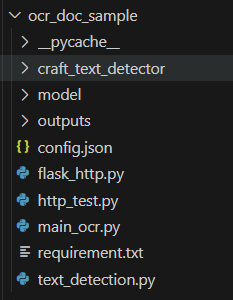
客户端运行在windows中，windows上也需要配置python环境，使用如下命令安装依赖包： pip install beautifulsoup4==4.12.3 natsort pillow==10.4.0 requests==2.32.3

客户短和服务器尚如果缺xx模块，就使用pip install xx，安装慢就后面加上 -i https://pypi.tuna.tsinghua.edu.cn/simple

## 2程序及目录描述

* 本软件包目录：

将该软件包同时拷贝到服务器上，把客户端程序http\_test.py和配置文件config.json拷贝到windows客户机上。



* 主目录Work目录

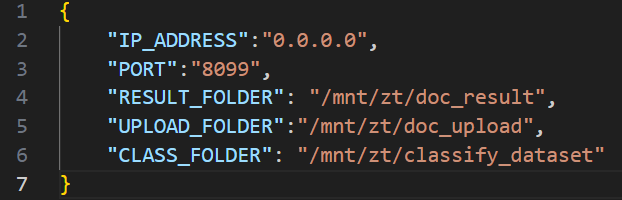
*主目录（work目录）：ocr\_doc\_sample*，包含所有东西

* config.json配置文件  
  IP\_ADDRESS是服务器的IP地址，默认为0.0.0.0即可

PORT是开放端口，按需修改，默认8099  
RESULT\_FOLDER：分割识别后的输出文件夹，需要修改为服务器上某个文件夹路径（绝对路径）

UPLOAD\_FOLDER：从windows上传到服务器的原档案图片存储路径

CLASS\_FOLDER：分类后的文件存储文件夹。在windows客户端程序上进行分类，会将图片及信息移动到该文件夹下的print/number/handwriting文件夹。

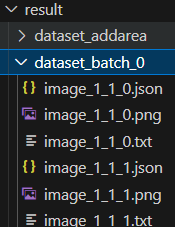
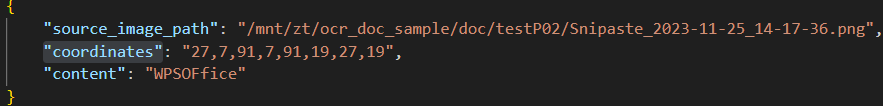


* 原始图像目录UPLOAD\_FOLDER

原始档案图片文件夹，windows上**按照每个人一个文件夹，**文件夹内是图片，一个总的文件夹包含多个人的文件夹，然后上传总文件夹。（例如doc\_windows/张三，doc\_windows/李四……，上传doc\_windows文件夹）

* 文本识别结果RESULT\_FOLDER

*result*：**分割识别后的输出**，内容包含json（内容+坐标，坐标顺序是左上，右上，右下，左下）和txt文本。  
*RESULT\_FOLDER/dataset\_addarea*：后续加框后存储的文件夹，实时识别，也是json和txt。

* 服务器监听程序

*flask\_http.py*：http服务程序，直接启动，等待客户端连接。

* 客户端程序（运行在windows下）

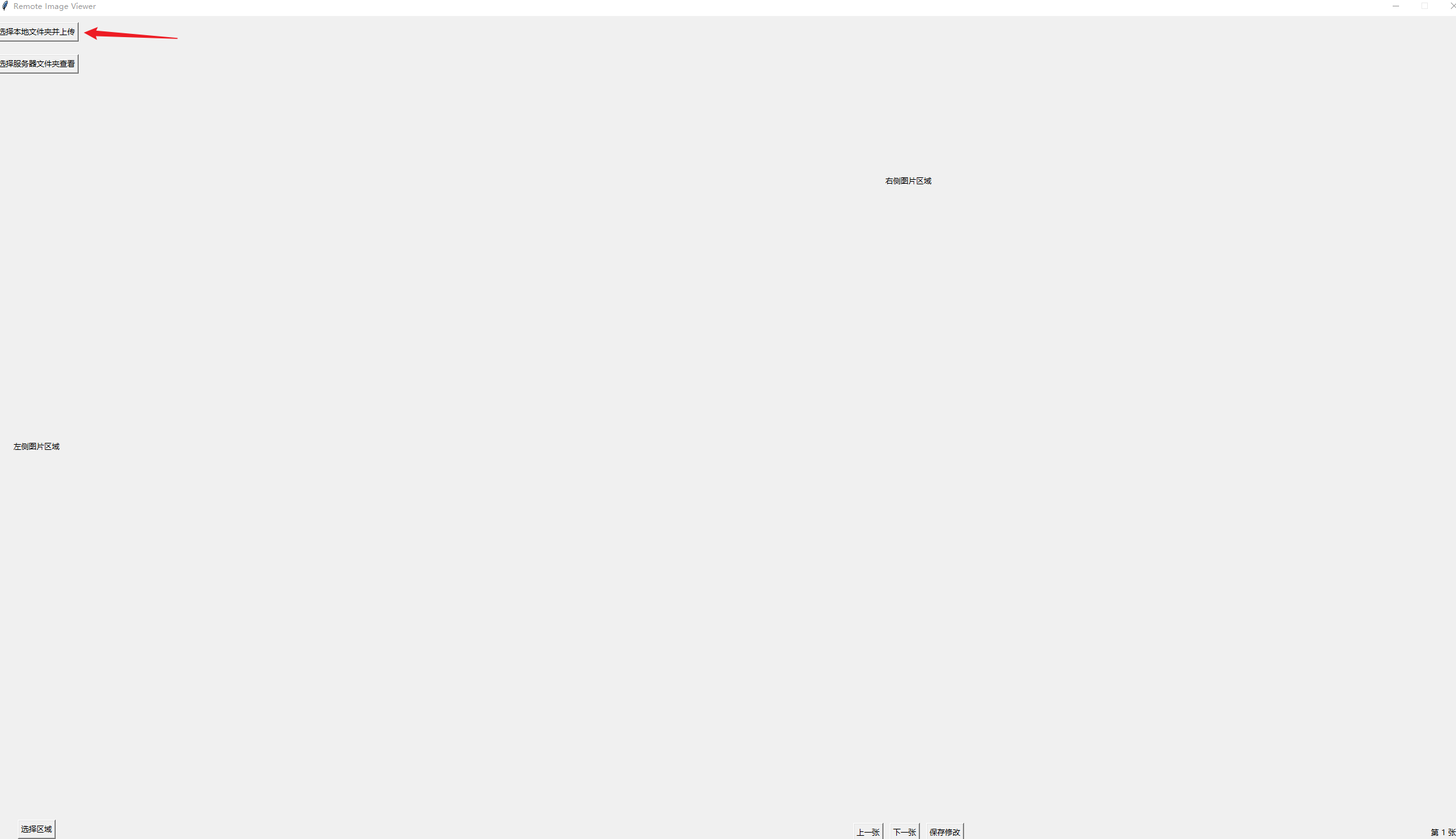
*http\_test.py*：是windows下的图形化应用。直接运行

## 3客户端使用方法

**http\_test.py**

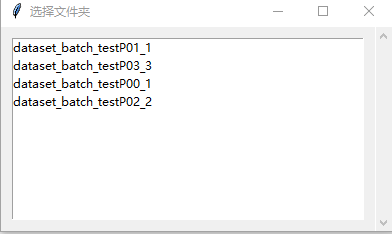
windows下的小程序。

1. 启动，点击选择本地文件夹并上传。将windows上的档案图片上传到服务器。（此时上传文件会回自动分割识别，可能需要等待较长时间后才能打开第一个文件夹，后续文件夹会在后台自动进行，可以先处理前面的文件夹）



1. 选择文件夹，

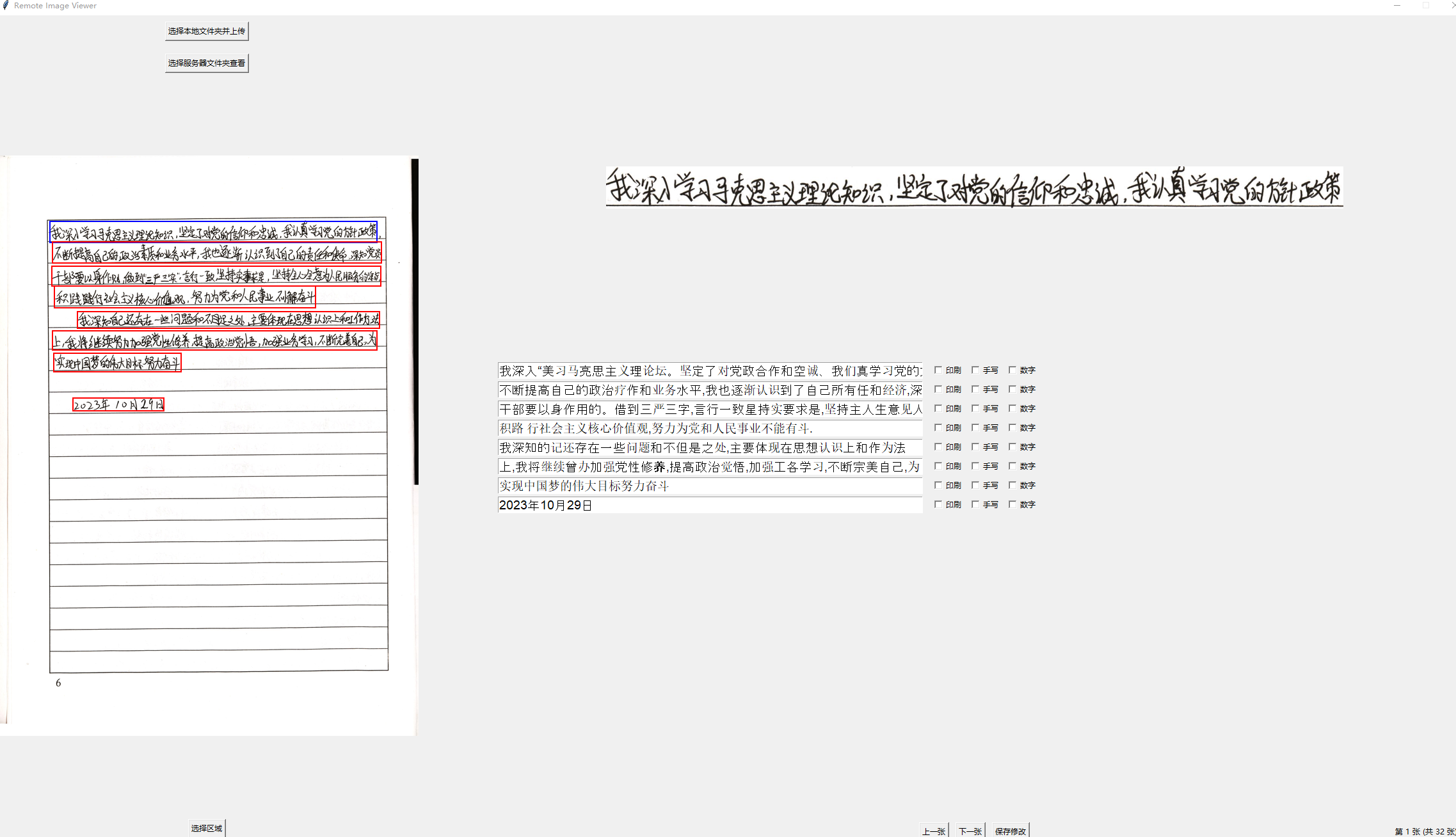
带纠错文件有两种，处理方式相同，但是可以通过文件夹名称区分。第一种dataset\_batch\_x是服务器程序分割后的图像。  
 第二种是手动 识别加框后的图象dataset\_addarea是“选择区域”手动加框后的文件夹（自动创建）。

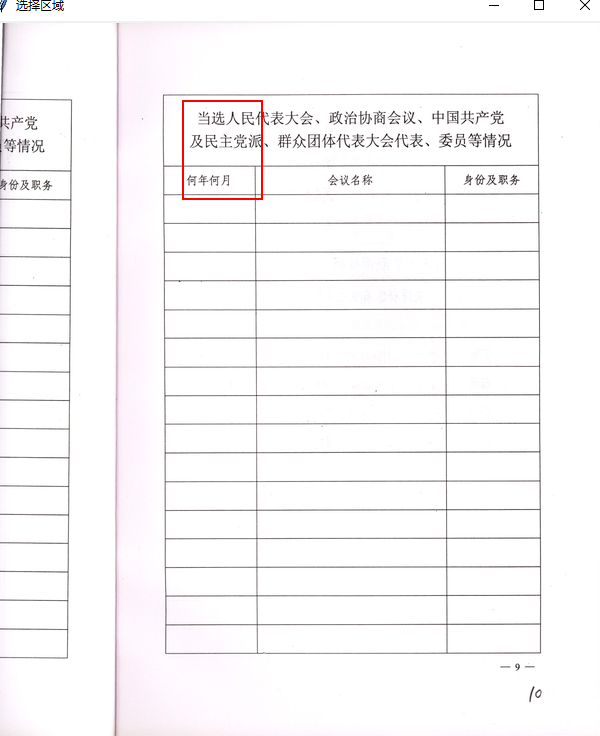


1. 左边显示原图以及当前切割图片的位置（红框），右边是当前选择的文本框及其识别出来的文字，用户可以直接编辑文本进行纠错，并保存纠错后的结果。

左下角“选择区域”按钮是**加框**，点击后用鼠标框柱需要的区域（**类似截图，选择完成后需要等几秒**服务器识别，保存到服务器的result/dataset\_area文件夹）。

点击左边原图上的框，右边会显示出大图，**大图右键可以删除图片**。





图x 用户手动加框示意图

4.文本分类保存。右键图片，删除就是将当前图片从服务器上删除。点击对应的分类按钮会将图片移动到CLASS\_FOLDER下。（有个bug，当只剩最后一张图片的时候，分类要双击才会分类）



5.文本框内修改内容，修改完点击保存修改，切换图片也**自动保存**，修改完点击分类按钮也**自动保存**。