## 使用在线网站

[Free Online OCR - Image to text and PDF to Doc converter](https://www.onlineocr.net/)

存在信息安全问题

且效果一般

## pytesseract

适用于表结构语句

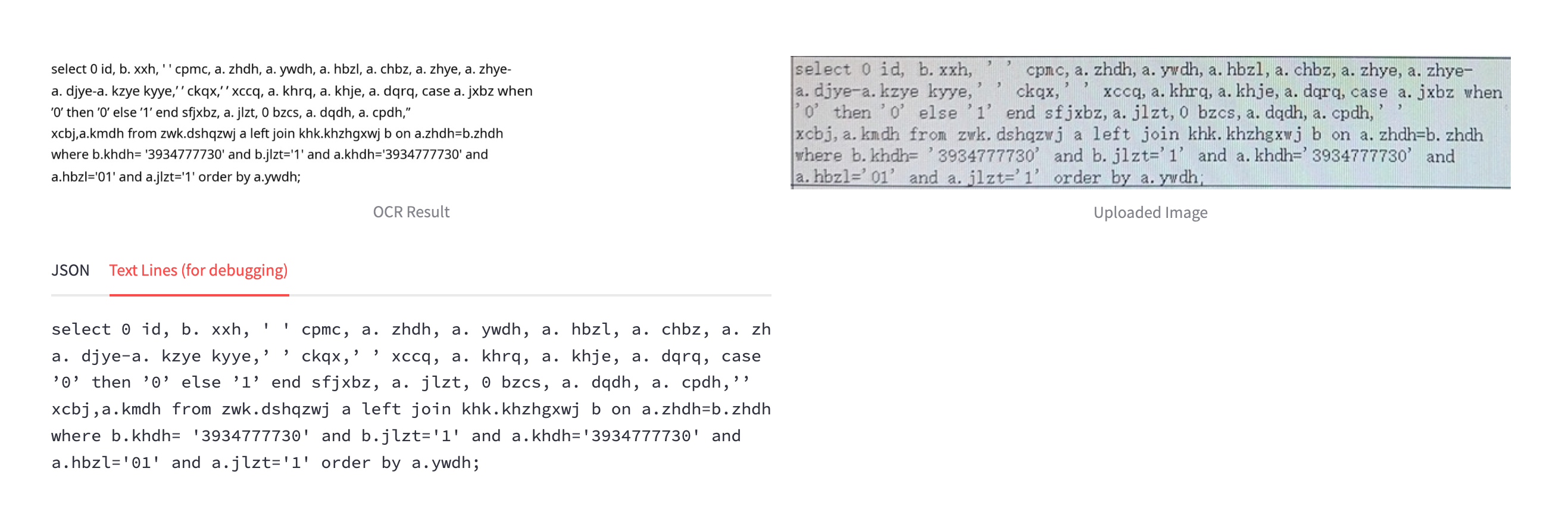
识别之后，做一些易错项目转换

import os  
import re  
from PIL import Image  
import pytesseract  
  
# 定义要保留的字符集，包括换行符  
allowed\_chars = re.compile(r'[a-zA-Z0-9(),;.@\n \_=®{}]')  
  
# 定义图片文件夹路径  
image\_folder = 'xingye'  
result\_folder = 'xingye/result'  
trans\_folder = 'xingye/trans'  
  
# 创建文件夹  
os.makedirs(result\_folder, exist\_ok=True)  
os.makedirs(trans\_folder, exist\_ok=True)  
  
  
# 遍历文件夹中的所有图片文件  
for filename in os.listdir(image\_folder):  
 if filename.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp', '.png')):  
 # 构建图片文件的完整路径  
 image\_path = os.path.join(image\_folder, filename)  
  
 # 使用 pytesseract 读取图片中的文本内容  
 image = Image.open(image\_path)  
 content = pytesseract.image\_to\_string(image)  
  
  
  
 # 过滤掉不需要的字符  
 filtered\_content = ''.join(filter(allowed\_chars.match, content))  
  
 # 将 @ 符号转换为 0  
 filtered\_content = filtered\_content.replace('@', '0')  
  
 filtered\_content = filtered\_content.replace('®', '0')  
  
  
 # 去掉下划线前后的空格  
 filtered\_content = re.sub(r'\s\*\_\s\*', '\_', filtered\_content)  
 filtered\_content = filtered\_content.replace('{', '(')  
 filtered\_content = filtered\_content.replace('}', ')')  
 # 将 InnoD8 转换为 InnoDB  
 filtered\_content = re.sub(r'InnoD8', 'InnoDB', filtered\_content)  
  
 # 构建输出文本文件的路径  
 output\_filename = os.path.splitext(filename)[0] + '.txt'  
 output\_path = os.path.join(result\_folder, output\_filename)  
  
 # 过程结果写入输出文件  
 trans\_path = os.path.join(trans\_folder, output\_filename)  
 with open(trans\_path, 'w', encoding='utf-8') as file:  
 file.write(content)  
  
  
 # 将结果写入输出文件  
 with open(output\_path, 'w', encoding='utf-8') as file:  
 file.write(filtered\_content)  
  
 print(f"处理完成，结果已写入 {output\_path} 文件。")

## surga--多模态模型

<https://github.com/VikParuchuri/surya>

适用于sql语句，因为sql语句中的 '',\* 等符号不可忽略，使用该模型较好



## PaddleOCR--模型

<https://github.com/PaddlePaddle/PaddleOCR>

## 扫描全能王在线

<https://www.camscanner.com/file/recent>

识别建表语句优秀，但是涉及数据安全问题

-- 有无开源版本？

## 文本处理

扫描全能王的处理模式

import cv2  
import numpy as np  
  
# 读取图片  
image = cv2.imread('image\_path.jpg', cv2.IMREAD\_COLOR)  
  
# 转换为灰度图像  
gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
  
# 应用高斯模糊  
blurred = cv2.GaussianBlur(gray, (5, 5), 0)  
  
# 使用自适应阈值  
thresh = cv2.adaptiveThreshold(blurred, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY, 11, 2)  
  
# 保存结果  
cv2.imwrite('enhanced\_image.jpg', thresh)

from PIL import Image, ImageEnhance, ImageFilter  
  
# 打开图片  
image = Image.open('image\_path.jpg')  
  
# 增强对比度  
enhancer = ImageEnhance.Contrast(image)  
image = enhancer.enhance(2)  
  
# 增强锐度  
enhancer = ImageEnhance.Sharpness(image)  
image = enhancer.enhance(2)  
  
# 保存结果  
image.save('enhanced\_image.jpg')