您好老师，整理了到目前为止进行的内容。  
  
前处理中能使用的方法都试过了。  
使用过的算法如下。:grayscale,edge detection,threshold,adaptive threshold,shadow removal,blur,dilation,erotion,perspective  
  
当使用adaptive threshold时，视觉效果并不差，但当实际运行ocr时，结果很差。  
总之，对应用perspective的图像进行ocr时效果最好。  
  
使用easyocr和paddle两种比较了ocrtool。  
  
easyocr虽然进行了custom training,但效果不佳而放弃,paddle的基本性能良好,因此采用了paddle。  
目前，除了模糊的部分外，所有部分都可以进行影像处理，O和0的区分存在模糊的问题。  
这一点可以从处理的数据中找到O，然后进行转换为0的过程。  
  
像csv文件的转换，对于进行perspective的图像，利用bounding box的坐标，找出min\_x、min\_y、max\_x、max\_y四个点。  
通过按照csv文件格式填充的方法实现了算法。  
  
上面标记的功能现在使用Python pyqt5以gui形式实现，perspective则自动装蜡盒的  
提取contour的最外角坐标进行perspective的方法，如果箱子看不清，以user输入的坐标为基准  
实现了两种进行perspective的方法。  
  
同时检测文本超过120时，存在识别率急剧下降的现象。 如何将相机分成两个进行看起来是最有效的。  
使用网络摄像头时，识别率为90%。 但是要给相机提供充足的光源。  
对于基本提供的img，除了严重模糊的部分外，全部成功识别了文本。  
front和behind合起来的算法还没有实现。  
现在考虑中的算法即使front和behind中不指定row和column也要实现  
输入整个盒子的row和column，提取x坐标图、y坐标图，然后是图表的  
front 从上到下， behind 从下到上填充 csv 文件， 如果有重复的部分  
打算以省略的方式在明天之前实现。  
  
现在可以做的部分是,通过影像的深度学习去除阴影和升级部分。  
如果时间允许,我想通过深度学习实现Upscaling算法,并正在寻找相关论文。  
只是这种情况下硬件配置不够充分，没有条件训练，所以只能用韩国的电脑进行。  
如果还有需要体现的部分，请告诉我。

Figure1. before preprocessing

figure2. After preprocessing

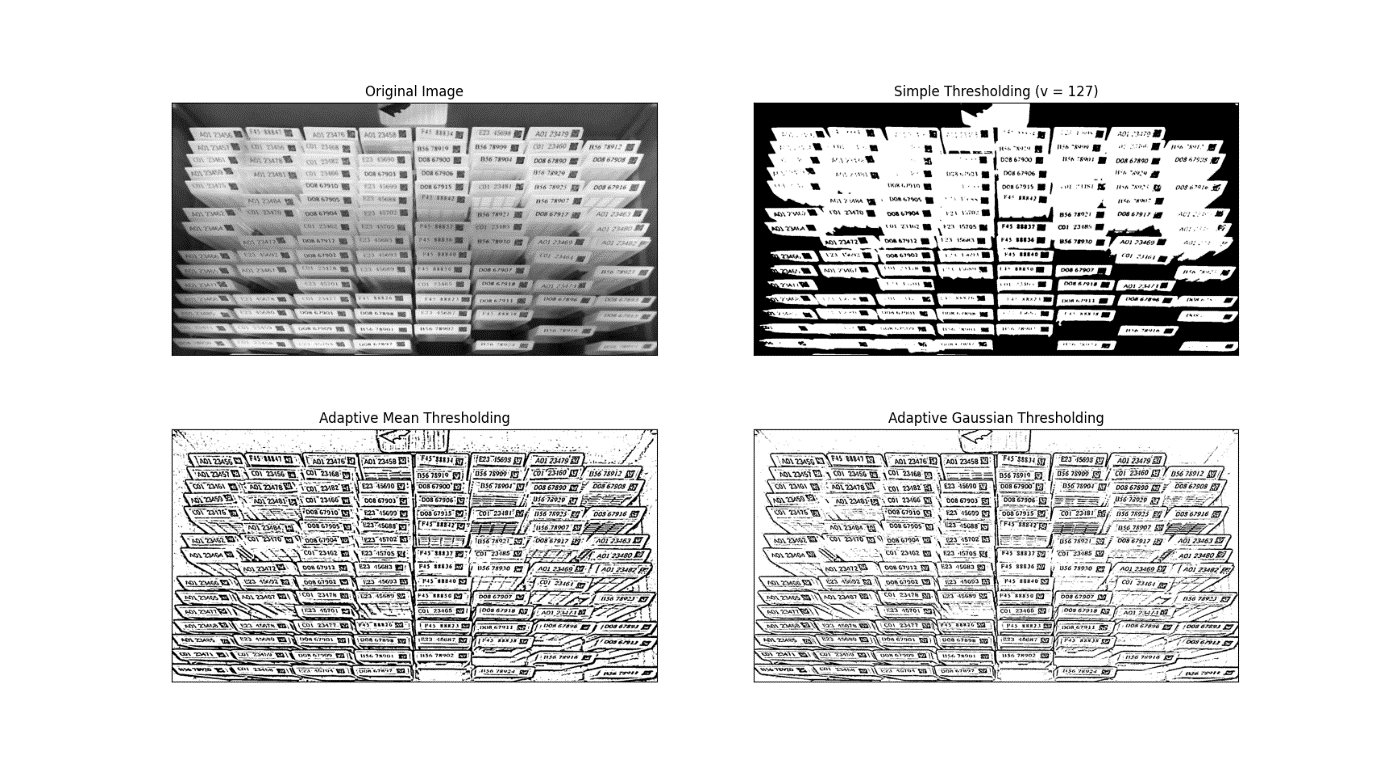


Figure3. threshold algorithms



Figure4. image captured by smartphone camera.

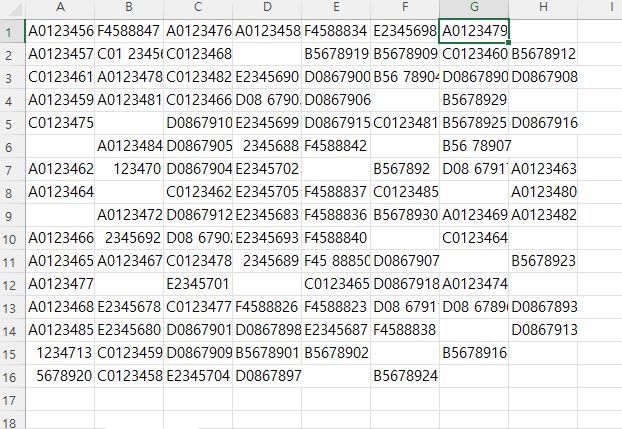


Figure5 result for Figure2.

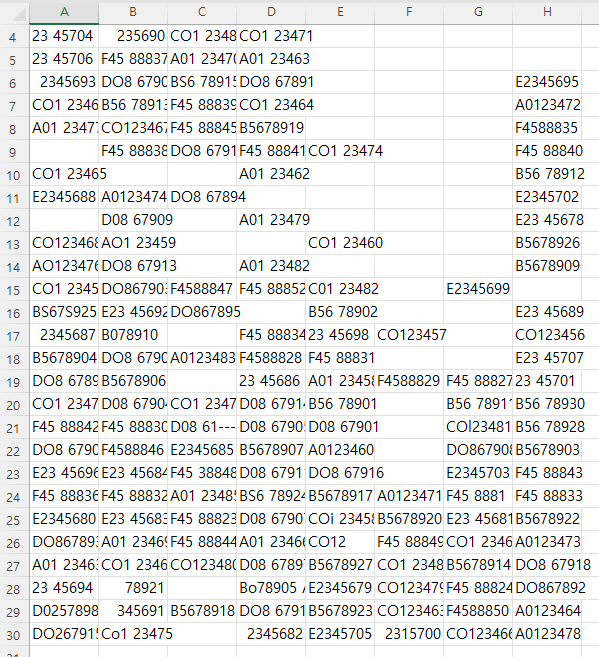


Figure6 results for Figure4.

I텍스트, 스크린샷, 컴퓨터, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Figure7 GUI tool