一、分析任务要求

**1. 手动为图片的每个字标框，框大小不一、左右不一**

方案为：用opencv实现鼠标画框，鼠标左键按下开始记录框的左上角坐标，鼠标左键松开开始记录框的右下角坐标。当键盘按下某个特定按键时，画框过程结束。保存该过程所有的框。

**2.Python+opencv写个程序，输入标过框的一整幅图片，输出一些数据到excel中**方案为：用lxml库实现保存数据到excel。

①每个框的正中心位置+字的半径

**②每个框内的字的重心（需要想方法算一算）**  
③每个框中心&字重心的连线向量方向角度，长度  
④每个框内黑色占比  
⑤整幅图画，笔墨最细多少，最粗多少，最细的位置，最粗的位置

**⑥每个字最后一笔收笔方向，就用框内的最下/最右下一处与水平线的角度判定即可**

**3.输出2幅新的图**

1. 输出每列字大小、左偏右偏的抽象呈现

输出原图背景颜色+每列的固定直线（黄色）+每个字抽象的大小不一的圆圈（红色）+每列圆圈的中心的折线连线（蓝色）+笔墨最细&最粗处的圆用红色实心圆覆盖

大概意思如右边这张图，直线和圈的颜色变一变，再把每列圆心连成折线。

→

1. 每个框内的字进行多次膨胀腐蚀，最后得到很模糊的黑团

（处理到认不清字的模糊程度）

输出原图背景颜色+黑团

二、算法流程

1、读RGB彩色图

2、画框并记录坐标：

1）用opencv实现鼠标画框，鼠标左键按下开始记录框的左上角坐标，鼠标左键松开开始记录框的右下角坐标。函数为cv2.setMouseCallback

2）当键盘按下某个特定按键时，画框过程结束。

3）依次保存该过程的所有坐标，存入

3、解析坐标为框：

依次遍历框

以框所在的区域为感兴趣区域

对该区域灰度化

对该区域二值化

对该区域形态学变化，