Otsu的思想假设某个阈值将一副图像分为前景背景，计算两幅图像之间的类间方差，前景占比乘以前景的平均灰度与全局的平均灰度差的平方+背景占比乘以背景的平均灰度与全局的平均灰度差的平方，当然这只是类间方差的定义，实际在计算时是将定义式与其他条件合并做了一个化简，得到前景占比乘以背景占比乘以前景平均灰度与背景平均灰度差的平方。源代码中是通过遍历1-255作为阈值，选取使最大类间方差最大的阈值作为最佳阈值，根据这个最佳阈值分割出前景。

自己曾在写带mask的Otsu算子上做了一点点优化，对于原始的Ostu有两层for循环，第一层是遍历阈值，第二层是统计前景与背景的占比与灰度累计。 而我在第二层时改用动态规划的思想，利用阈值i去计算阈值i+1的前景与背景的占比与灰度累计。省去了一层for循环。使用1000\*600的图像循环运行10万次，效率提高了8.4%。

实际上如果图像前景与背景像素值差距不大，大小比例悬殊，图像复杂，细节多，表现在直方图上就是没有清晰的双波峰、单波谷，Otsu的效果不会很好，但是我们可以采取一些方法去改善直方图的形状。

1. 图像平滑
2. 利用边缘梯度
3. 使用mask掩膜