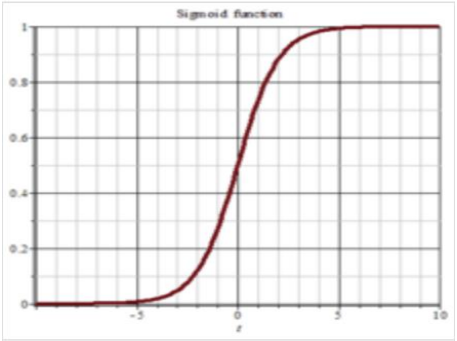
实验题目：
2.1 对比度调整
设计一个 Sigmoid 函数，实现对图像的对比度调整；
使用 opencv 窗口系统的 slider 控件，交互改变 Sigmoid 函数的参数，实现不同程度的对比度调整；
2.2 背景相减
对图像 I 和对应的背景图 B，基于背景相减检测 I 中的前景区域，并输出前景的 mask.
分析你的方法可能产生误检的情况，并上网查阅背景相减的改进方法，设法改进结果。

一. 敲脑袋想出来的调整对比度的方法：
线性方法：

$$g(i,j) = a * f(i,j) + b$$

我可以通过调整常数 a 和 b 的值通过使用 ptr 指针遍历访问每个像素点来对图像的对比度进行调整,但实验要求通过 sigmoid 函数压缩取值范围只需要调整 slidebar 的一个值来对图像进行整体的对比度调整，当然这个线性函数是调整的思想基础。
二. sigmoid 函数的设计

$$S(x) = \frac{1}{1+e^{-x}}$$



Sigmoid 曲线

选择对于 jpg 图像进行对比度调整
我们每次滑动 slidebar 只需要给出的值
将对比度 slidebar 命名为 Contrast，对应的初始值 ContrastValue = 20
亮度 slidebar 命名为 Bright ,对应的初始值 BrightValue = 100
在滑动块创建函数中将两者调节范围的最大值都设置为 200
最后编写回调函数 cvTrackbarCallback
对于像素级别的调整，我使用 ptr 函数来进行访问
首先设计了指数 t 的表达式

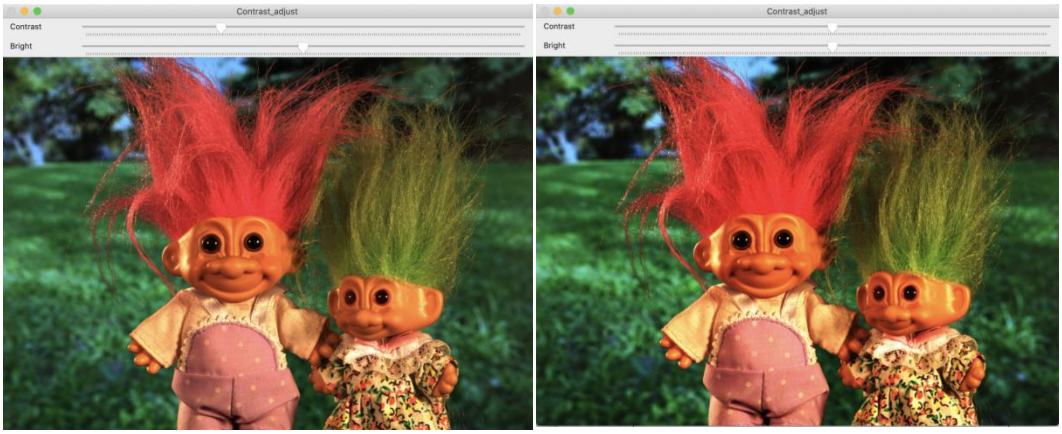
```
float  t = (((src.ptr<Vec3b>(i)[j][c]-127.5) / 255.00)*ContrastValue*0.1);
```

通过对比原图和初始化生成图片进行最终的常数微调

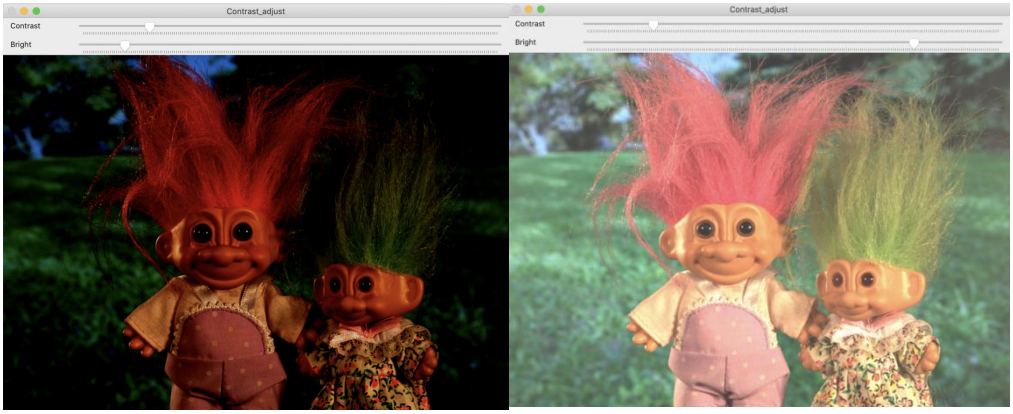
```
dst.ptr<Vec3b>(i)[j][c]= saturate_cast<uchar>(src.ptr<Vec3b>(i)[j][c]*((1.00
/ (1.00 + exp(-t))) + 0.3) + BrightValue - 100);
```

最后得到的结果如下图所示

（1）调整对比度至不同的程度的结果：



（2）调整亮度（实验未要求）



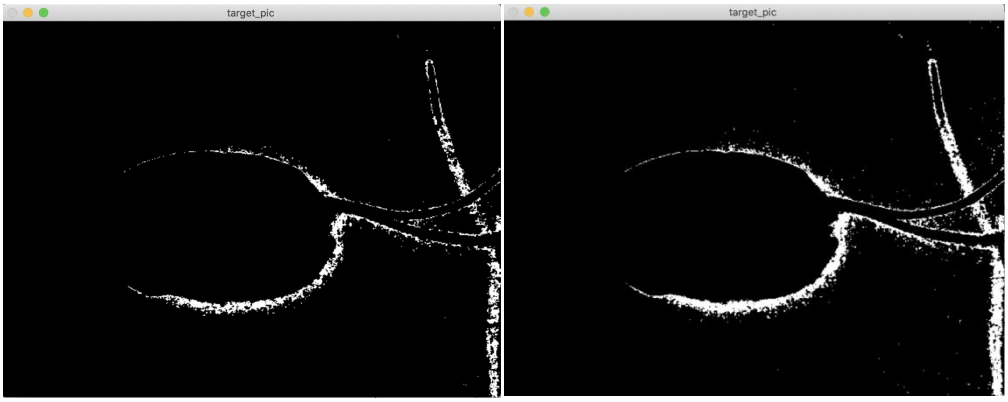
虽然感觉调对比度好像和调亮度没有什么很大区别。。

实验 2.2 背景相减实验

首先基础的思路肯定还是遍历图像，通过背景图和原图之间的像素代数运算来进行 mask（掩膜）提取

尝试了进行简单的像素三通道和相减，设置了阈值进行 mask 提取

对于差值超过阈值的像素赋值为白色，差值小于阈值的像素赋值为黑色



(阈值=30)　　(阈值=60)

阈值翻倍后未发现有优化，反而出现了更多的图像噪声

但是连续设置变化的阈值（即在 30-60 间随机取值）也未见有任何的改善

所以，这个简单相减的想法过于简陋，需要更为复杂的算法

查阅网上的资料 ，进行了如下图的改进

$$I(x,y)=(r,g,b)$$
$$B(x,y)=(r,g,b)$$

$$Diff(x,y)=\| I(x,y)-B(x,y) \|^2$$

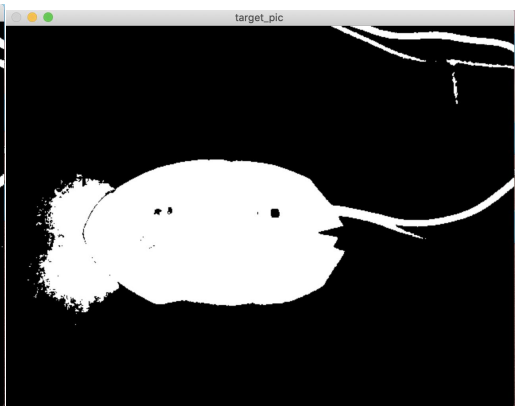
$$Is\ foreground\ if\ Diff(x,y)>T\ \text{(a given threshold)}$$

对于每个像素求前后景每个通道差的平方和，并最后对和进行平方

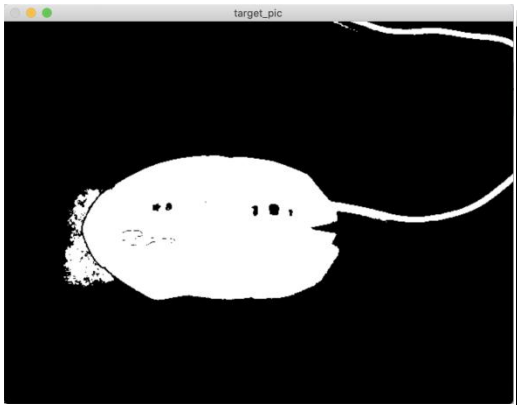
对于该值再进行阈值判断

图 01

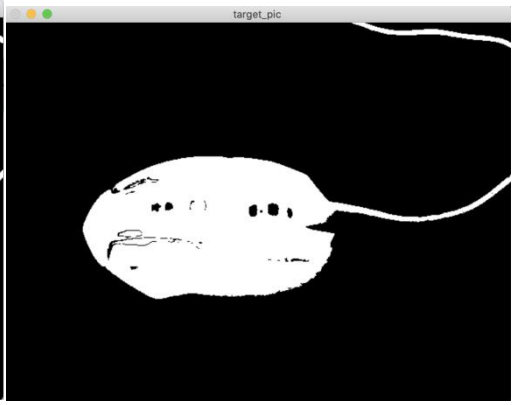
阈值=30



阈值=60



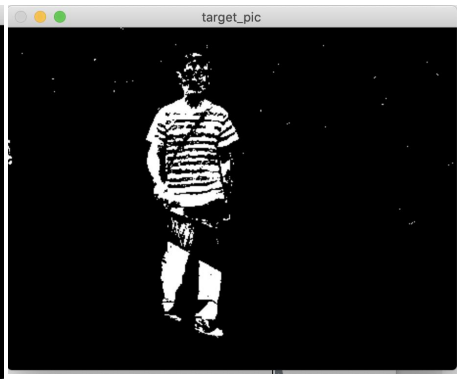
阈值=90



阈值=120

对于这张图片，基本能够得到比较好效果的阈值在 90-120 之间
过低的阈值导致鼠标后部的光没有被滤去
图 02

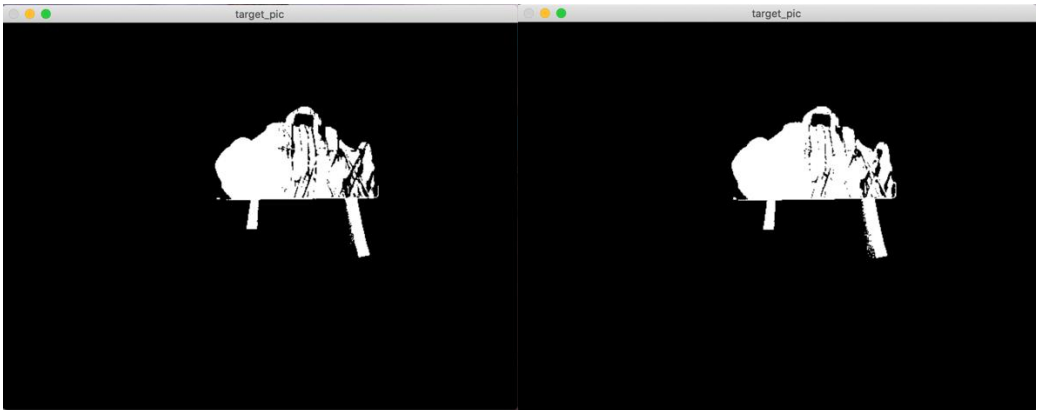
阈值=160



阈值=120

阈值=80
可以看出对于这张图而言，随着阈值的提高，关于人的一部分特征也变模糊
但是阈值降低又会在图像的其他地方产生急剧增多的噪声
所以平衡值大概在 80-120 之间
图 07

阈值=80　　阈值=100
最佳阈值在 100 附近



阈值=100　　阈值=80

阈值=80　　阈值=100

综上所述，使用该算法基本的可以稳定提取出前景的阈值大部分稳定在 80-120 范围内，过低的阈值一般会造成提取出的 mask 出现更多图像噪声，而过高的阈值会使得 mask 失去原先的一部分图像特征

之后我在网上继续搜索背景相减，找到的大部分方法是用于多帧的视频处理

了解了 ostu 算法来对图像进行二值化阈值处理

结果分析与体会：

这次实验拖了比较长的时间，一直没有找到合适的用于 mask 提取的函数，花了比较多的时间，但是收获匪浅。