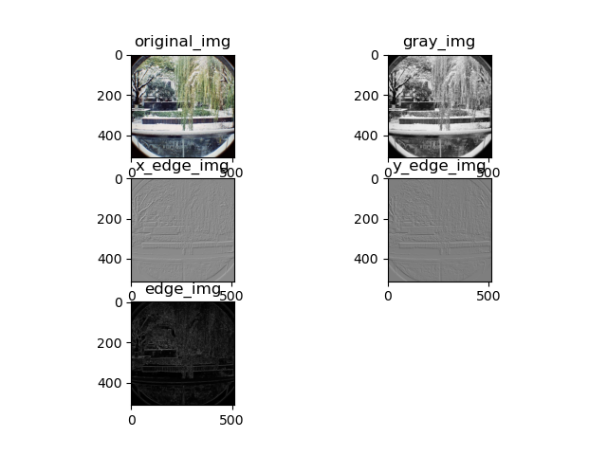
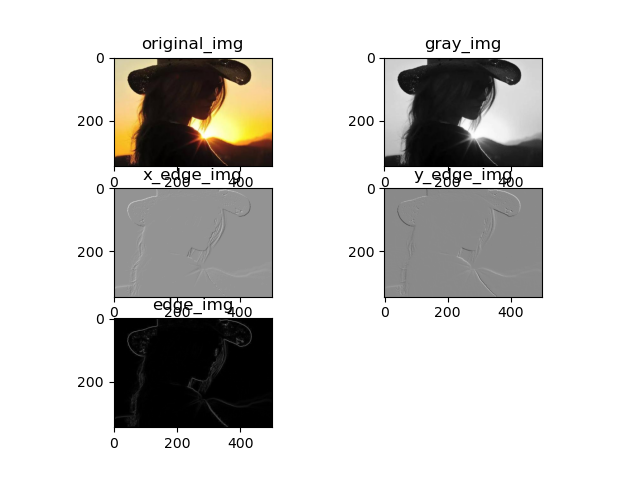
**实验2 数字图像分割与边缘检测实验**

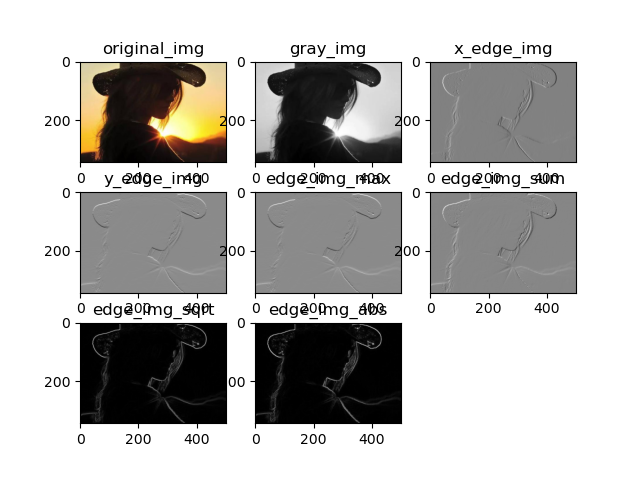
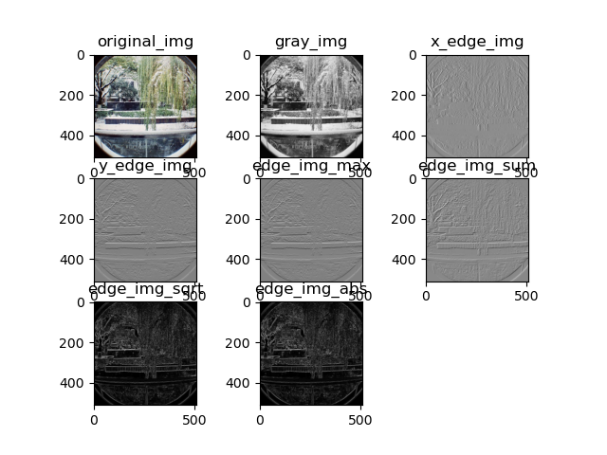
**作业一：使用一阶检测算子（导数）进行图像的边缘检测。**

我选择了Roberts算子和Prewitt算子进行了图像的边缘检测。

Roberts算子对其卷积进行计算时，采用的公式为：



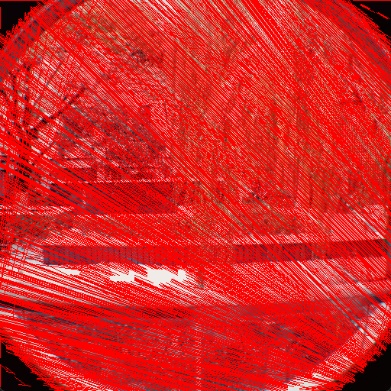
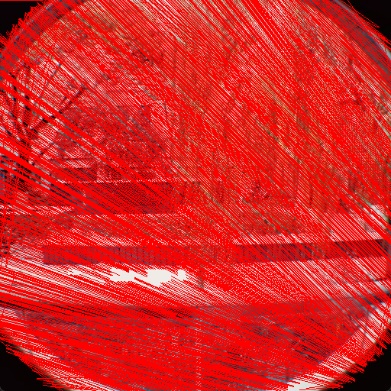
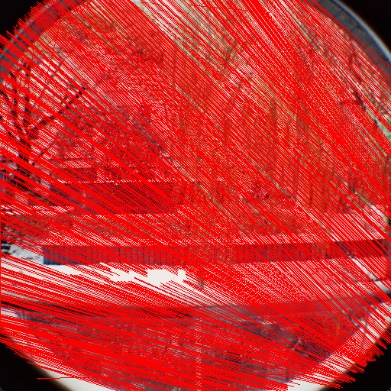
Prewitt算子对其卷积进行计算时，考虑了四种方法：

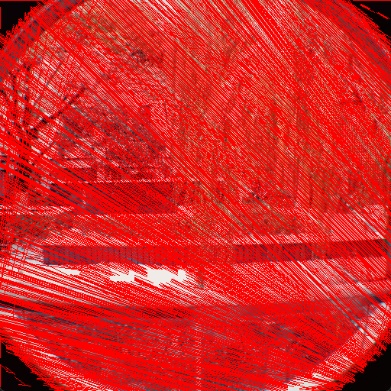
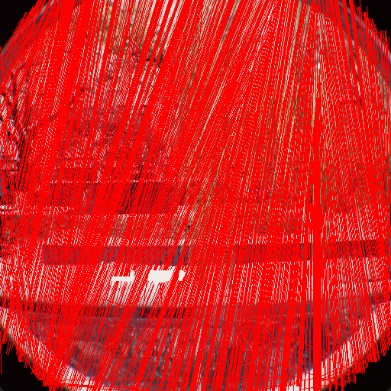
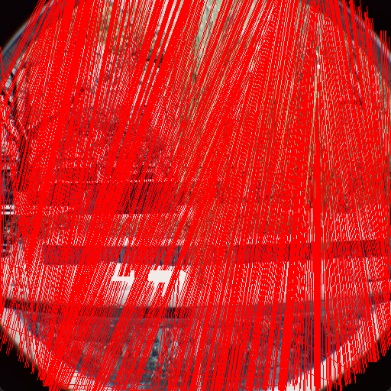
1. 用某点的x向梯度和y向梯度的最大值代表该点梯度；
2. 用x向梯度和y向梯度的和代替该点梯度；
3. 用x向梯度和y向梯度的平方根代替该点的梯度；
4. 用x向梯度和y向梯度绝对值的和代替该点的梯度。

Roberts算子：对垂直边缘的检测效果好于斜向边缘、对具有陡峭的低噪声的图像效果最好、定位精度高、对噪声敏感，但无法抑制噪声的影响、提取边缘的结果是边缘比较粗、边缘定位不是很准确

Prewitt算子：边缘检测结果在水平方向和垂直方向均比Roberts算子明显、适合用来识别噪声较多/灰度渐变的图像、对噪声有抑制作用（抑制噪声的原理是通过像素平均），但像素平均相当于对图像的低通滤波，所以Prewitt算子对边缘的定位不如Roberts算子

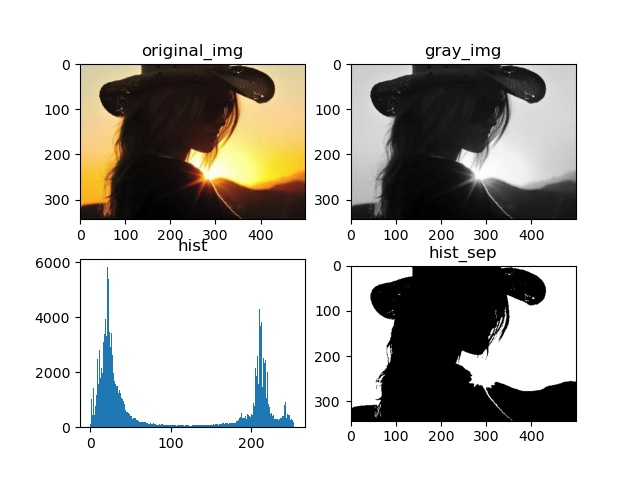
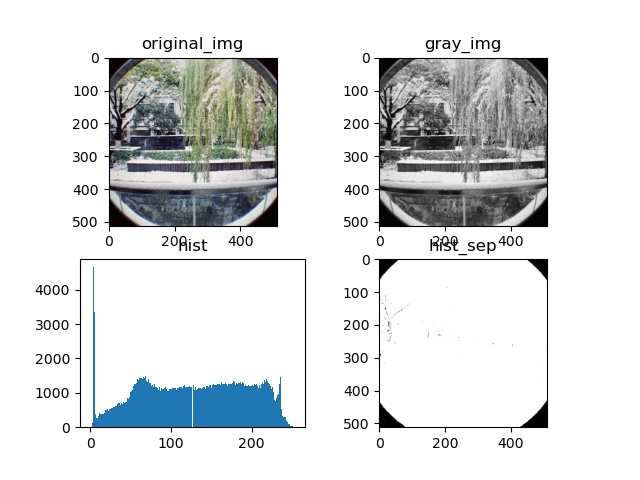
**作业二： Hough线检测**

任务分为两个部分，第一个部分是对作业一中两个算子获得的边缘提取结果进行Hough先检测，第二个部分是用3组不同的参数进行Hough线检测，使用两张图片共计12个实验结果需要展示。为了提取更长的Hough线，我选择对HoughLinesP函数中的minLineLength进行调整，选择了10、50、100三个数值，其他参数的设置为cv2.HoughLinesP(result, 1, 1 \* np.pi / 180, 10, minLineLength=?, maxLineGap=5)。下面的结果展示将把同一个算子的结果放在一起进行对比，所以先展示roberts算子的结果。

实验一种选择了四种不同的策略对prewitt算子进行了边缘检测，这里采取了和封装函数策略相同的sqrt进行了实验。

很明显，当minLineLength越大时，检测到的Hough线越长且越少，很容易理解。

**作业三：图像二值分割**

选择了双峰，实验结果如下：