**Project5-Canny**

一、运行环境

操作系统：Windows11

Python版本：3.7

第三方库：numpy 1.21.6、matplotlib 3.5.3、scipy 1.7.3、pillow 9.4.0

二、过程分析

（1）findDerivatives函数

①计算边缘梯度：先使用Sobel算子计算出x方向与y方向的梯度，再用高斯函数对其进行平滑处理；

②计算边缘强度：计算x方向与y方向的梯度平方和后再进行开根号处理；

③计算边缘方向：通过x方向与y方向的梯度求出边缘方向。

（2）nonMaxSup函数

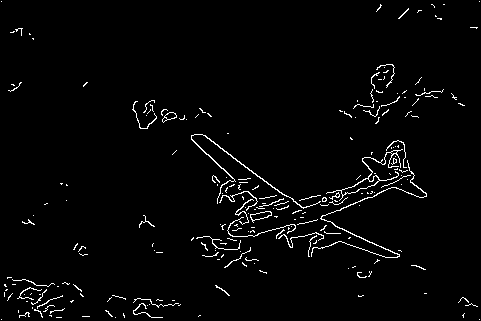
①非极大值抑制：沿着梯度方向在八邻域内搜索最大值。

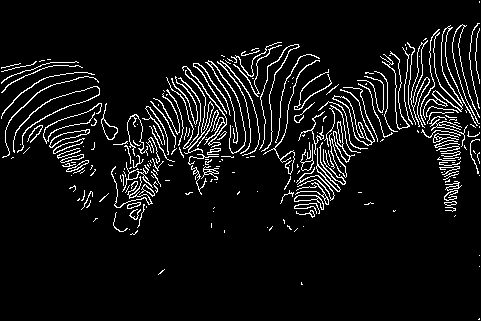
（3）edgeLink函数（已在pyc文件中给出）

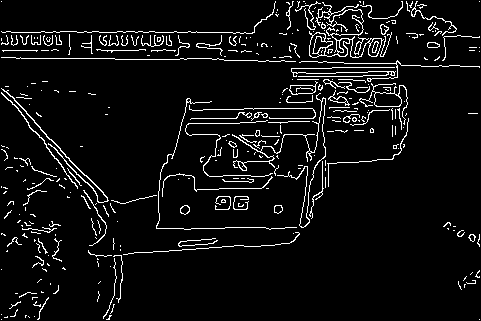
①阈值滞后处理：应用双阈值检测来确定真实的和潜在的边缘；

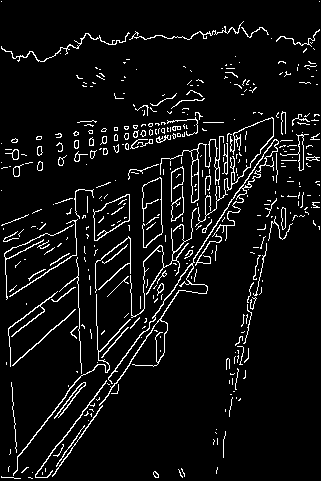
②边缘连接：通过抑制孤立的弱边缘最终完成边缘检测。

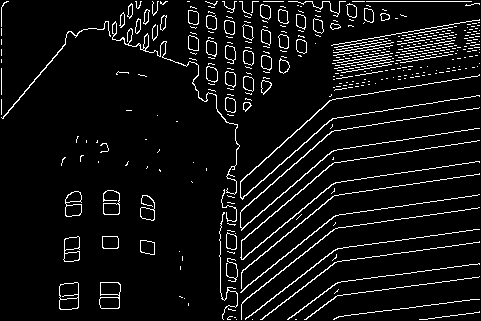
三、运行结果

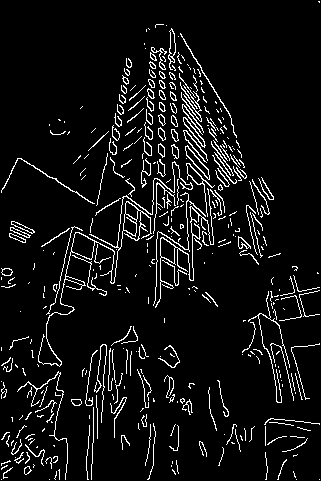
 

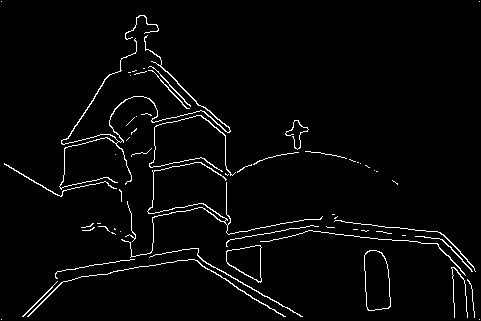
 

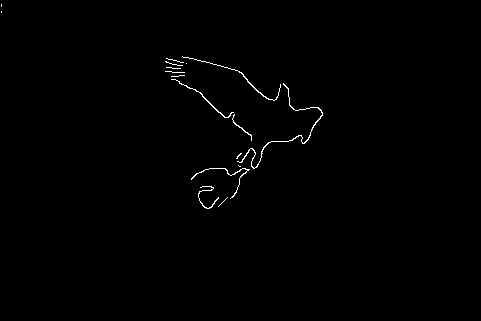
 

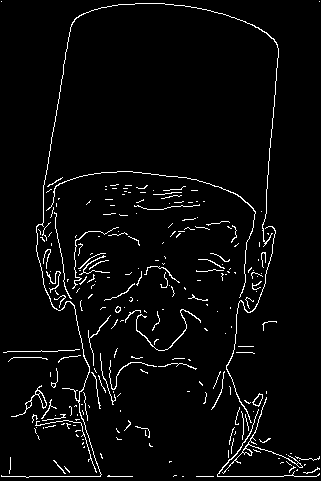
 

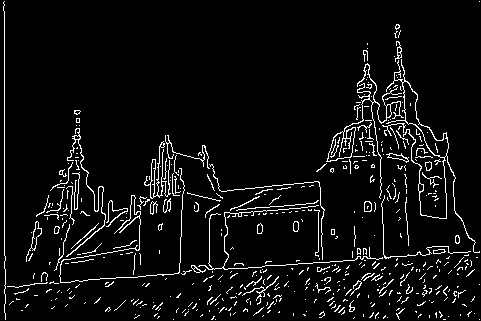
 

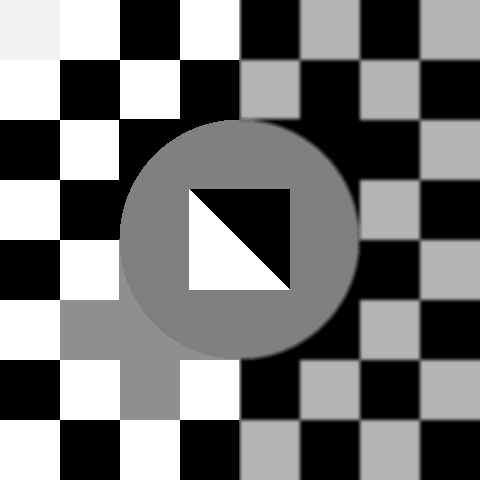
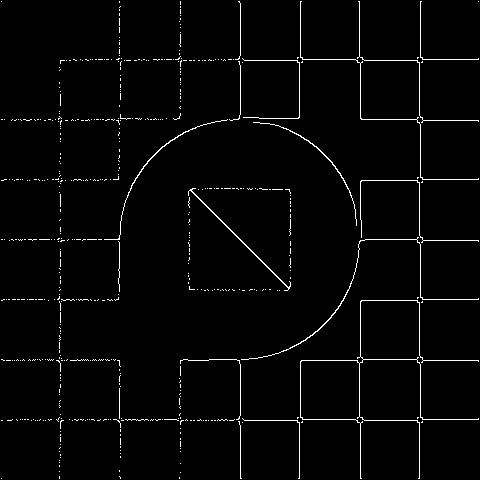
 

四、补充代码

①findDerivatives

|  |
| --- |
| Magx = signal.convolve(I\_gray, dx, mode=**'same'**) Magx = signal.convolve(Magx, gaussian, mode=**'same'**)  Magy = signal.convolve(I\_gray, dy, mode=**'same'**) Magy = signal.convolve(Magy, gaussian, mode=**'same'**)  Mag = np.sqrt(np.square(Magx) + np.square(Magy))  Ori = np.arctan(np.divide(Magy, np.where(Magx == 0, 1e-10, Magx))) |

②nonMaxSup

|  |
| --- |
| H, W = Mag.shape **for** i **in** range(1, H - 1):  **for** j **in** range(1, W - 1):  **if** grad\_Ori[i, j] == 0 **or** grad\_Ori[i, j] == 180 **or** grad\_Ori[i, j] == -180:  suppressed[i, j] = 1 **if** Mag[i, j] > max([Mag[i, j - 1], Mag[i, j + 1]]) **else** 0  **elif** grad\_Ori[i, j] == 45 **or** grad\_Ori[i, j] == -135:  suppressed[i, j] = 1 **if** Mag[i, j] > max([Mag[i - 1, j + 1], Mag[i + 1, j - 1]]) **else** 0  **elif** grad\_Ori[i, j] == 90 **or** grad\_Ori[i, j] == -90:  suppressed[i, j] = 1 **if** Mag[i, j] > max([Mag[i - 1, j], Mag[i + 1, j]]) **else** 0  **else**:  suppressed[i, j] = 1 **if** Mag[i, j] > max([Mag[i - 1, j - 1], Mag[i + 1, j + 1]]) **else** 0 |