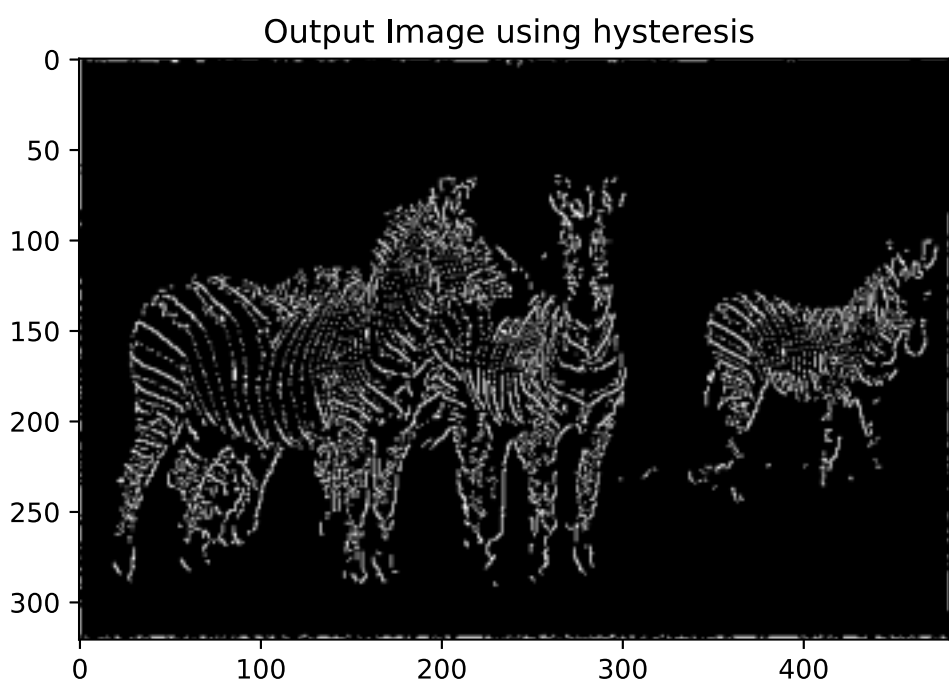
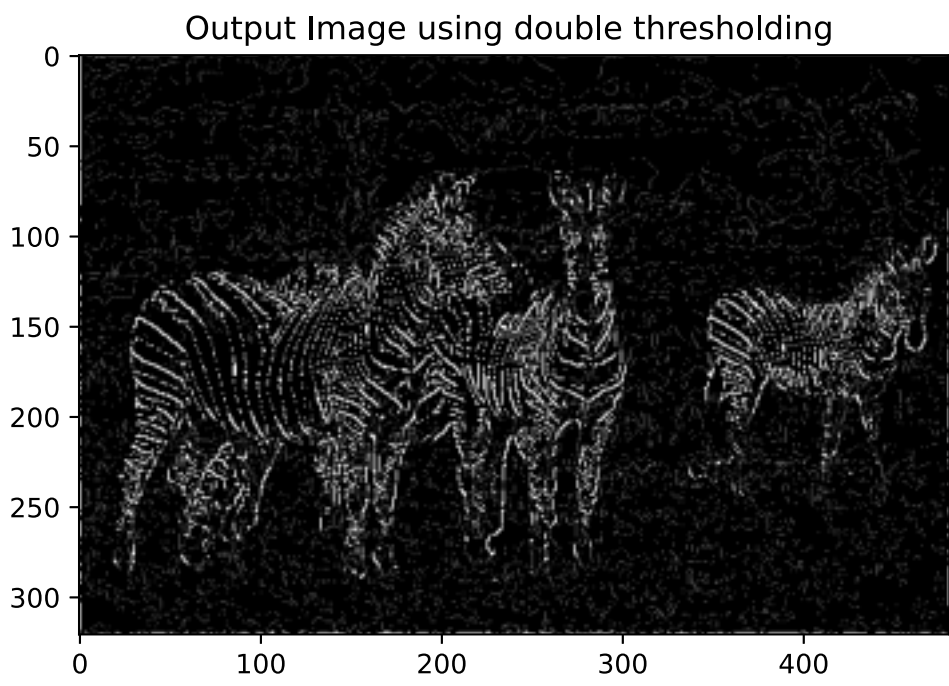
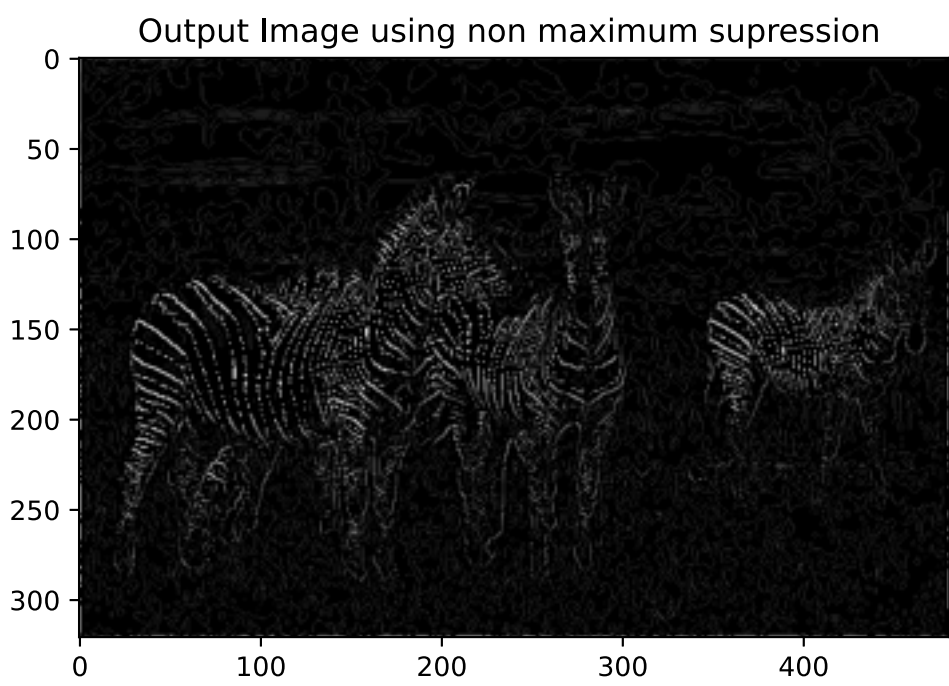
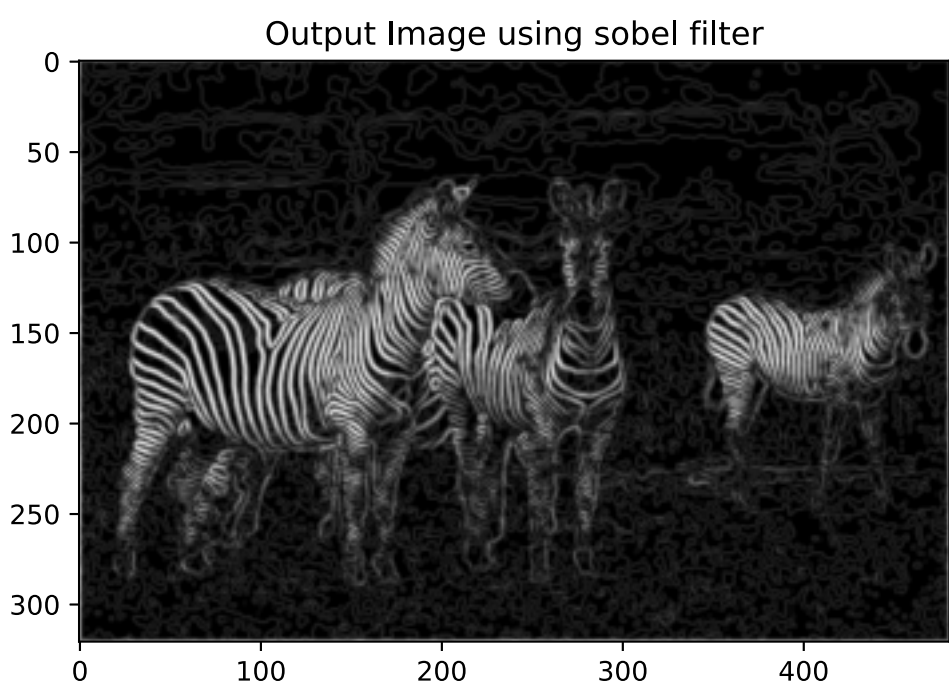
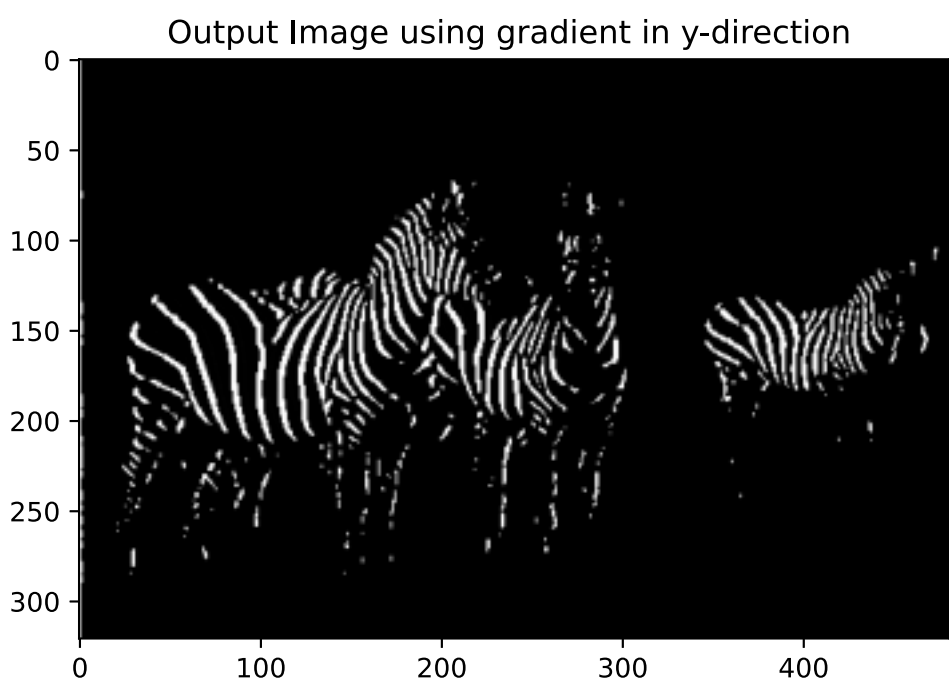
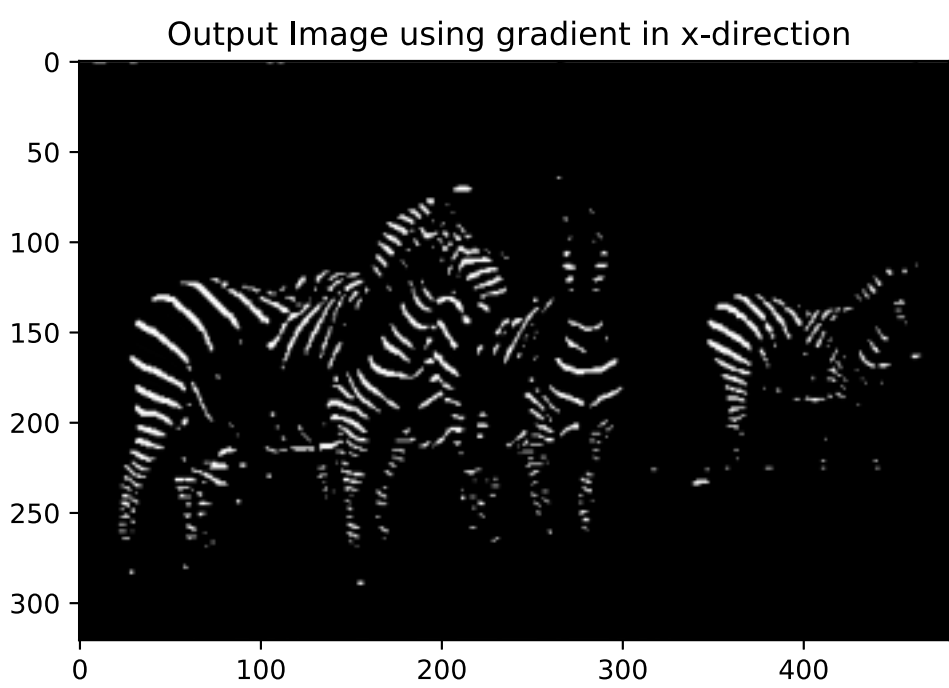
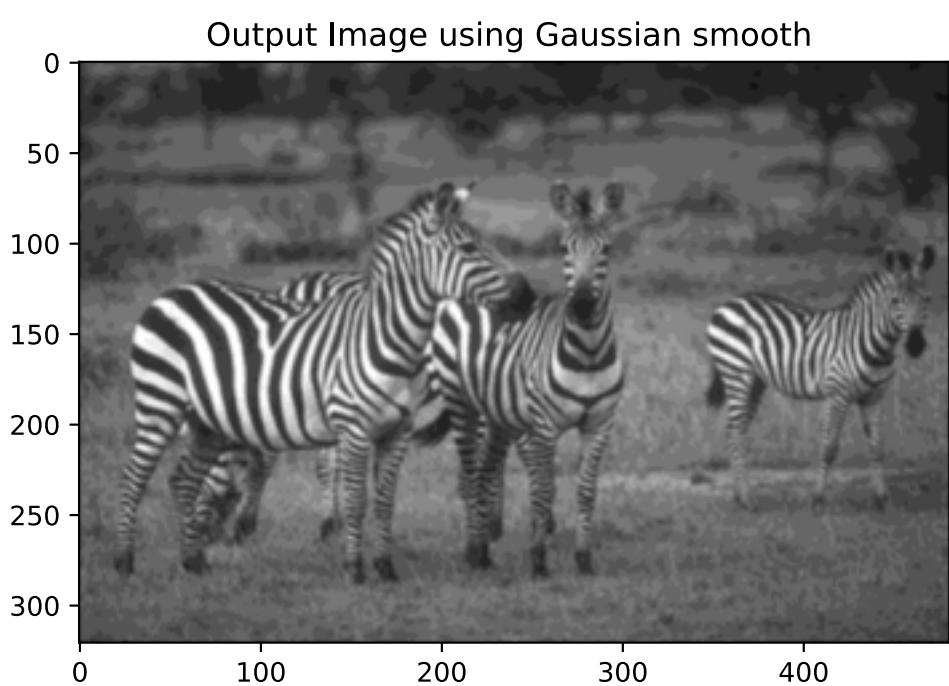


```
In [1]: from canny_detector import canny_detector

In [48]: img_path = "C:/Users/default.DESKTOP-IU77C8K/OneDrive - jbnu.ac.kr/uni_D/first sem/computer science/lab 2/zebra.jpg"

In [50]: canny = canny_detector(img_path=img_path,highThresholdRatio=0.25,lowThresholdRatio=0.5, double_thresh=True, threshold_ = 45,gaussian_size=3,gaussian_sigma=1.3)
res = canny.detect_edge()
```



```
In [21]: import cv2 as cv
from PIL import Image
import numpy as np

max_lowThreshold = 200
window_name = 'Edge Map'
title_trackbar = 'Min Threshold:'
ratio = 3
kernel_size = 5
def CannyThreshold(val):
    low_threshold = val
    img_blur = cv.blur(src, (3,3))
    detected_edges = cv.Canny(img_blur, low_threshold, low_threshold*ratio, kernel_size)
    mask = detected_edges != 0
    dst = src * (mask[:, :,None].astype(src.dtype))
    cv.imshow(window_name, dst)

src = cv.imread(img_path)
cv.namedWindow(window_name)
cv.createTrackbar(title_trackbar, window_name , 0, max_lowThreshold, CannyThreshold)
CannyThreshold(10)
cv.waitKey()
```

Out[21]: -1