Date : 08/06/2023

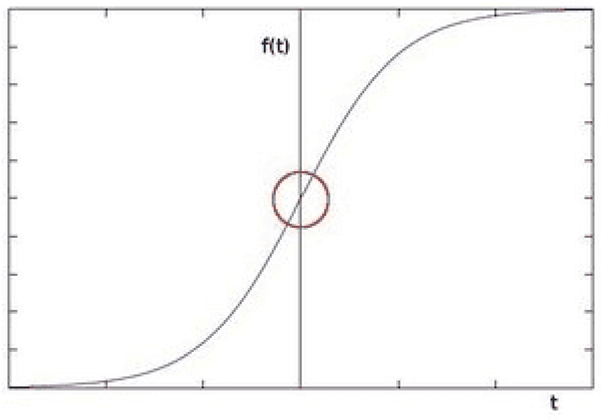
AIM: Implement program for Edge detection.

Theory :

Edge detection is an image-processing technique that is used to identify the boundaries (edges) of objects or regions within an image. Edges are among the most important features associated with images.

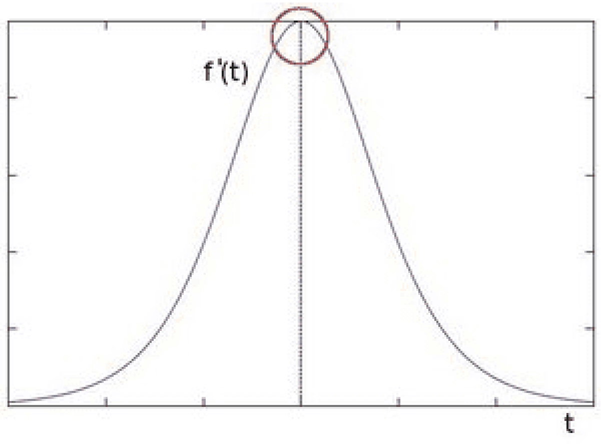
Blurring smoothens the intensity variation near the edges, making it easier to identify the predominant edge structure within the image. We use the `GaussianBlur()` function for blurring the image. We supply the size of the convolution kernel which specifies the degree of blurring.

Sobel Edge Detection is one of the most widely used algorithms for edge detection. The Sobel Operator detects edges marked by sudden changes in pixel intensity, as shown in the figure below.



Pixel intensity as a function of t

The rise in intensity is even more evident when we plot the first derivative of the intensity function.



First Derivative of Pixel intensity as a function of t

The above plot demonstrates that edges can be detected in areas where the gradient is higher than a particular threshold value. In addition, a sudden change in the derivative will also reveal a change in the pixel intensity.

Syntax :

`Sobel(src, ddepth, dx, dy, ksize)`

src	input image.
dst	output image of the same size and the same number of channels as src .
ddepth	output image depth, see **combinations**; in the case of 8-bit input images it will result in truncated derivatives.
dx	order of the derivative x.
dy	order of the derivative y.
ksize	size of the extended Sobel kernel; it must be 1, 3, 5, or 7.

Canny Edge Detection is one of the most popular edge-detection methods in use today because it is so robust and flexible. The algorithm itself follows a three-stage process for extracting edges from an image. Add to it image blurring, a necessary preprocessing step to reduce noise.
`Canny(image, threshold1, threshold2, apertureSize, L2gradient)`

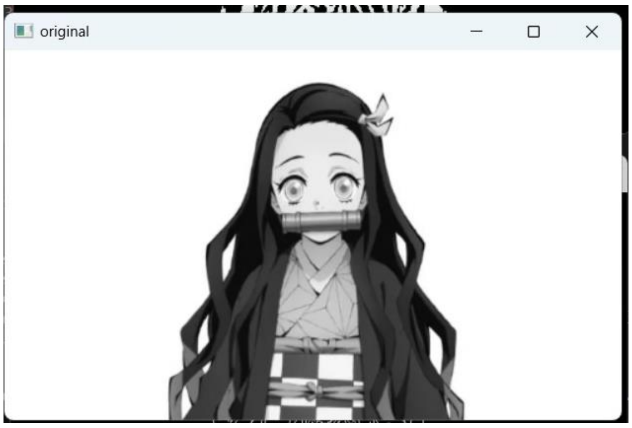
image	8-bit input image.
threshold1	first threshold for the hysteresis procedure.
threshold2	second threshold for the hysteresis procedure.
apertureSize	aperture size for the Sobel operator.
L2gradient	a flag

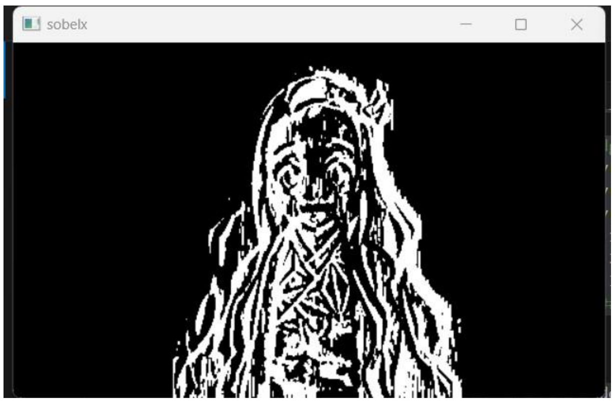
CODE :

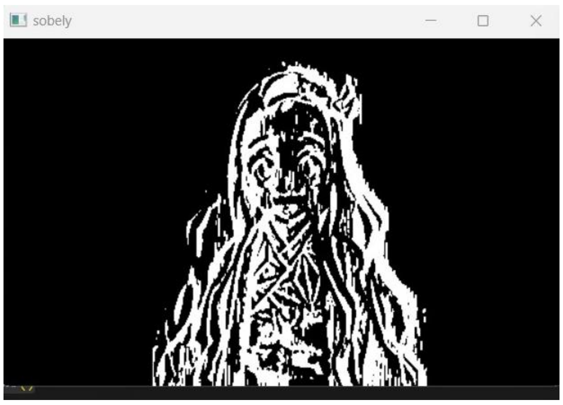
```
# sobel and canny edge detection

import cv2

img = cv2.imread('C:\\Users\\HP\\OneDrive\\Desktop\\sam\\sem 
6\\IPV\\pract1\\expt2\\nezuko.png',flags=0)
img = cv2.resize(img,(500,300))
cv2.imshow("original",img)
######### soble edge detection #########

img_blur = cv2.GaussianBlur(img,(3,3), 0, 0)
```

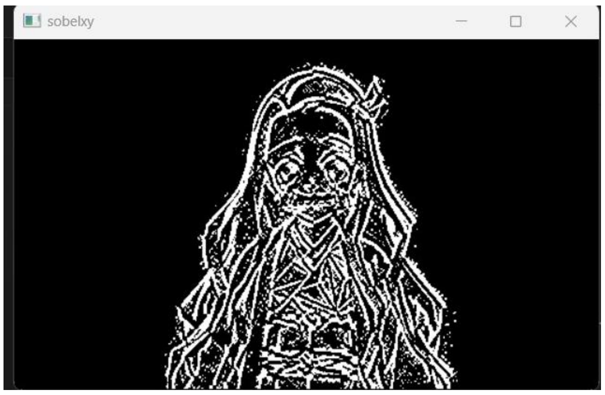
```
sobelx = cv2.Sobel(img_blur, ddepth=cv2.CV_64F,dx= 1, dy=0,ksize=3) # ddepth
specifies the precision of the output image
sobely = cv2.Sobel(img_blur, ddepth=cv2.CV_64F,dx= 0,dy= 1,ksize=3)
sobelxy = cv2.Sobel(img_blur, ddepth=cv2.CV_64F,dx= 1,dy= 1,ksize=3)

cv2.imshow("sobelx",sobelx)
cv2.imshow("sobely",sobelx)
cv2.imshow("sobelxy",sobelxy)

######### canny edge detection #########
canny1 = cv2.Canny(img,100,200, apertureSize=5,L2gradient = True)
canny2 = cv2.Canny(img,100,200, apertureSize=5)
canny3 = cv2.Canny(img,100,200,L2gradient = True)
cv2.imshow("canny 1 ",canny1)
cv2.imshow("canny 2 ",canny2)
cv2.imshow("canny 3 ",canny3)
#######################################
cv2.waitKey(0)
cv2.destroyAllWindows()
```

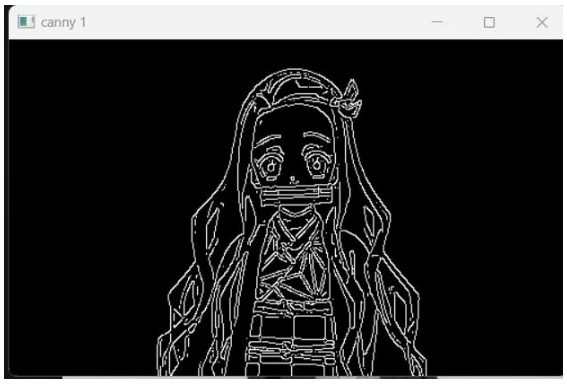
OUTPUT:

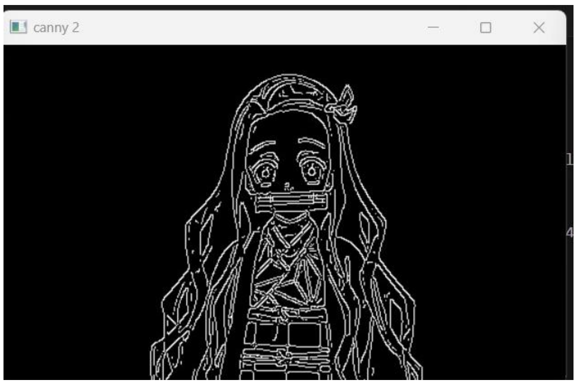


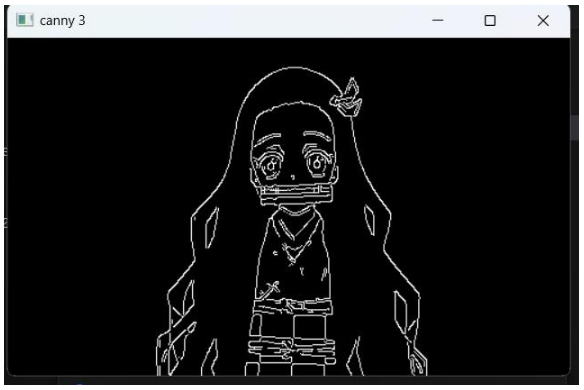












Conclusion :

Sobel edge detection uses 1st order derivative , while Canny edge detection uses 2nd order derivative.