**Canny-Edge Algorithm using PYTHON**

**CODING**

import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('messi5.jpg',0)

edges = cv2.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(edges,cmap = 'gray')

plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()

**EXPLANATION :**

First argument is our input image. Second and third arguments are our min Val and max Val respectively. Third argument is aperture size. It is the size of Sobel kernel used for find image gradients. By default it is 3. Last argument is L2gradient which specifies the equation for finding gradient magnitude. If it is True, it uses the equation mentioned above which is more accurate, otherwise it uses this function: Edge\_Gradient (G) = |Gx| + |Gy| default, it is False.

**INPUT IMAGE :**

****

**OUTPUT IMAGE :**

****