**Canny Edge Detection:**

Canny edge detection algorithm uses image gradient for edge detection in the given image.

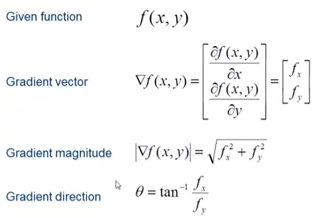
It is a multi-steps algorithm with the following steps:

1. Image transformation: Perform the necessary transformations like scaling, cropping, masking etc. to get the resultant output image on which edge detection has to be done.
2. Noise reduction: In this step we reduce the noise from image by applying various filters like Gaussian filter, averaging etc. that we have learnt in the previous tutorials.
3. Intensity gradient of the image: It is most important step of this algorithm. Canny edge detection uses intensity gradient of image for edge detection as on edges there will be a drastic change in intensity as compared to other regions in the image.

To find the gradient of image we find the partial derivation of image in both horizontal and vertical direction.

The final magnitude of gradient will be the resultant of both x and y derivative images.

The angle of gradient at any point is always perpendicular to the tangent on edge from that point.

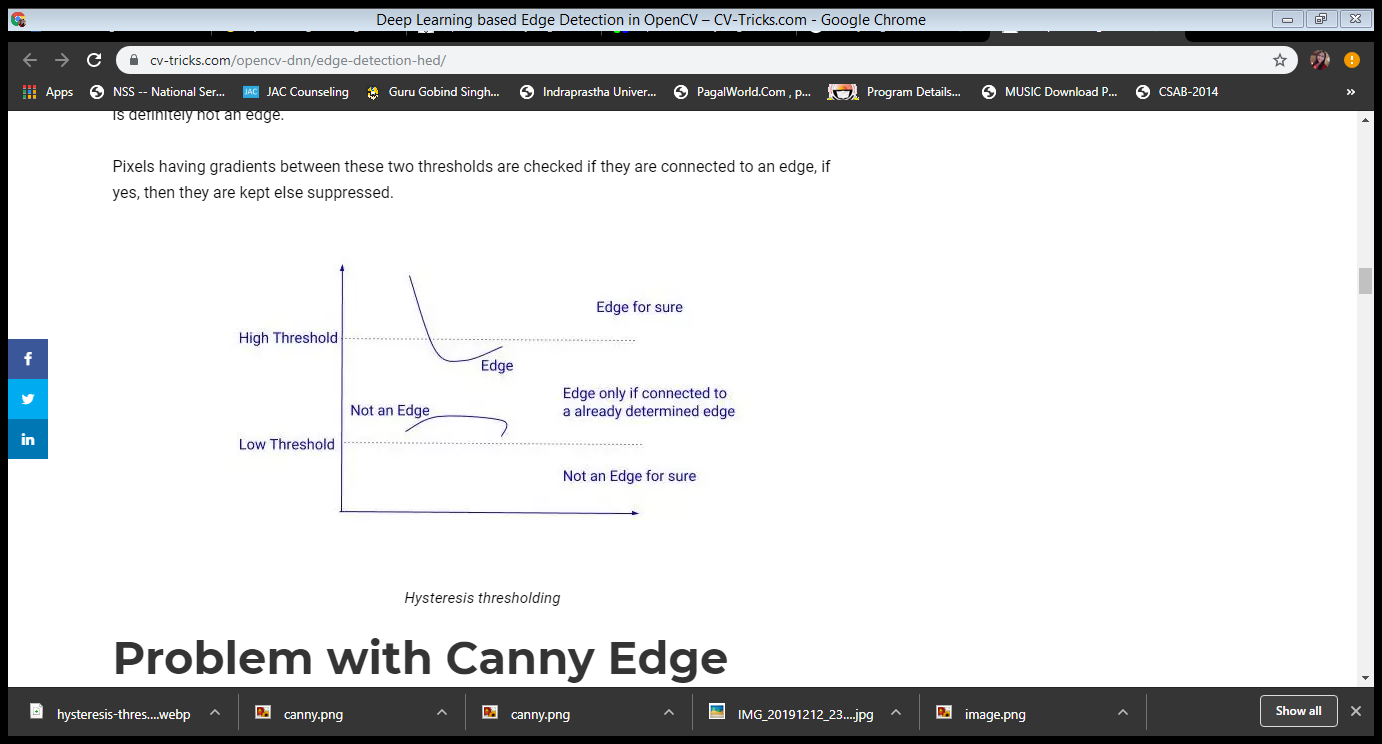


1. Find maximum gradient points: In this step we check at every pixel if the gradient is high or not. Also check local optimal at every pixel to avoid any unwanted pixel to be marked as an edge. This is done by checking the magnitude of gradient of the neighboring pixel in the direction of gradient.
2. Hysteresis Thresholding**:** In this step, we perform thresholding on the resultant gradient image by the following rules:

High threshold= A very high value is chosen in such a way that any pixel having gradient value higher than this value is definitely an edge.

Low threshold= A low value is chosen in such a way that any pixel having gradient value below this value is definitely not an edge.

Pixels having gradients between these two thresholds are checked if they are connected to an edge, if yes, then they are kept else suppressed.



cv2.Canny( image, low\_threshold, high\_threshold ) function is used in openCV to perform Canny edge detection algorithm. The function has three arguments:

1. Image: it signifies the source image on which smoothing has to be done.
2. low\_threshold: it signifies the lower bound of hysteresis thresholding.
3. high\_threshold: it signifies the higher bound of hysteresis thresholding.

The function returns an image after edge detection.

Python code:

# Importing libraries

import cv2

#read input image

img = cv2.imread('eagle.jpg',0);

edge = cv2.Canny(img, 100, 200); #low\_threshold = 100

#high\_threshold = 200

cv2.imshow('Original',img);

cv2.imshow('Output',edge);

#wait for 10 sec

cv2.waitKey(10000);

cv2.destroyAllWindows();

OUTPUT:

