Aim: To study Edge detection with Canny.

Objective: Perform Canny Edge detector using Noise reduction using Gaussian filter, Gradient calculation along the horizontal and vertical axis, Non-Maximum suppression of false edges, Double thresholding for segregating strong and weak edges, Edge tracking by hysteresis.

Theory:

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

What are the three stages of the Canny edge detector To fulfill these objectives, the edge detection process included the following stages.

- Stage One - Image Smoothing.
- Stage Two - Differentiation.
- Stage Three - Non-maximum Suppression.

The basic steps involved in this algorithm are:

- Noise reduction using Gaussian filter.
- Gradient calculation along the horizontal and vertical axis.
- Non-Maximum suppression of false edges.
- Double thresholding for segregating strong and weak edges.
- Edge tracking by hysteresis.

Now let us understand these concepts in detail:

1. Noise reduction using Gaussian filter This step is of utmost importance in the Canny edge detection. It uses a Gaussian filter for the removal of noise from the image, it is because this noise can be assumed as edges due to sudden intensity change by the edge detector. The sum of the elements in the Gaussian kernel is 1, so the kernel should be normalized before applying convolution to the image. In

this Experiment, we will use a kernel of size 5 X 5 and sigma = 1.4, which will blur the image and remove the noise from it. The equation for Gaussian filter kernel is s because this noise can be assumed as edges due to sudden intensity change by the edge detector. The sum of the elements in the Gaussian kernel is 1, so the kernel should be normalized before applying convolution to the image. In this Experiment, we will use a kernel of size 5 X 5 and sigma = 1.4, which will blur the image and remove the noise from it. The equation for Gaussian filter kernel is

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

2. Gradient calculation When the image is smoothed, the derivatives Ix and Iy are calculated w.r.t x and y axis. It can be implemented by using the Sobel-Feldman kernels convolution with image as given: after applying these kernel we can use the gradient magnitudes and the angle to further process this step. The magnitude and angle can be calculated as

3. Non-Maximum Suppression This step aims at reducing the duplicate merging pixels along the edges to make them uneven. For each pixel find two neighbors in the positive and negative gradient directions, supposing that each neighbor occupies the angle of pi /4, and 0 is the direction straight to the right. If the magnitude of the current pixel is greater than the magnitude of the neighbors, nothing changes, otherwise, the magnitude of the current pixel is set to zero.

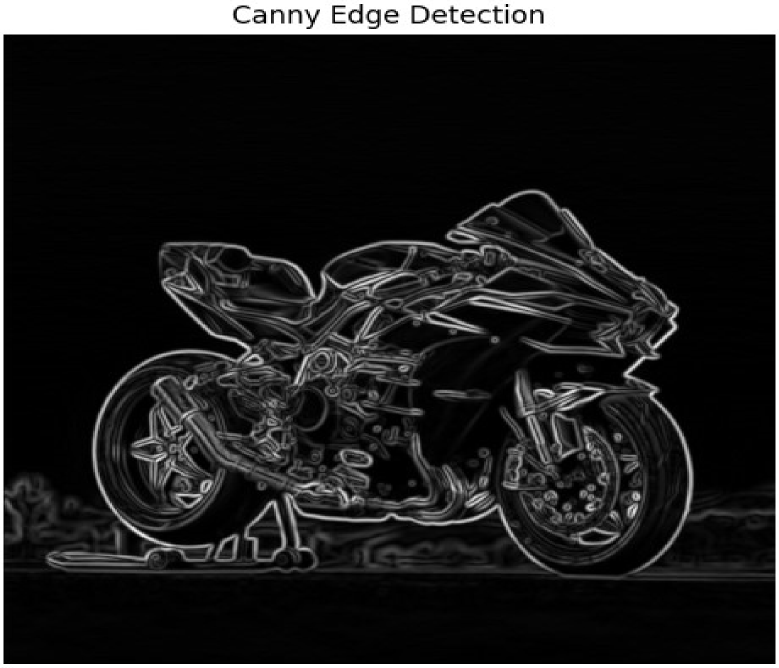
4. Double Thresholding The gradient magnitudes are compared with two specified threshold values, the first one is lower than the second. The gradients that are smaller than the low threshold value are suppressed, the gradients higher than the

high threshold value are marked as strong ones and the corresponding pixels are included in the final edge map. All the rest gradients are marked as weak ones and pixels corresponding to these gradients are considered in the next step.

Output:

Input Image

Canny Edge Detection



Conclusion:

In conclusion, the study of Canny edge detection reveals its significance in computer vision and image processing. The algorithm's multi-stage process, involving blurring, gradient calculation, non-maximum suppression, and thresholding, effectively identifies edges in images. While it demands parameter tuning, Canny's accuracy and versatility make it a cornerstone for various applications, despite some limitations. Overall, understanding Canny edge detection is vital for harnessing its power in real-world image analysis tasks.