Edge Detection and Image Segmentation

Edge detection is a technique in image processing that identifies points in a digital image where the image brightness changes sharply or has discontinuities. These points are called edges.

Edges often correspond to important features in an image, like the boundaries of objects, changes in surface orientation, or texture changes.

Why is Edge Detection Needed?

- **To identify object boundaries:** Edges help separate objects from the background or from each other.
- **To simplify image analysis:** Instead of processing every pixel, focusing on edges reduces data while keeping important structural information.
- **To aid in object recognition:** Detecting edges is often the first step in recognizing shapes and objects within images.
- **To assist in segmentation:** Edges help divide an image into meaningful regions.
- **In real-world applications:** Edge detection is essential in fields like medical imaging (finding tumors), autonomous vehicles (detecting lanes), and facial recognition.

Gradient

In image processing, a gradient represents the change in intensity (brightness) of pixels across an image. It's a fundamental concept used to detect edges, boundaries, and shapes.

$$\text{Gradient} = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

The image gradient is a vector that has:

- **Magnitude**: how strong the intensity changes.
- **Direction**: where the intensity increases the most.

Common Algorithms:

Sobel Operator

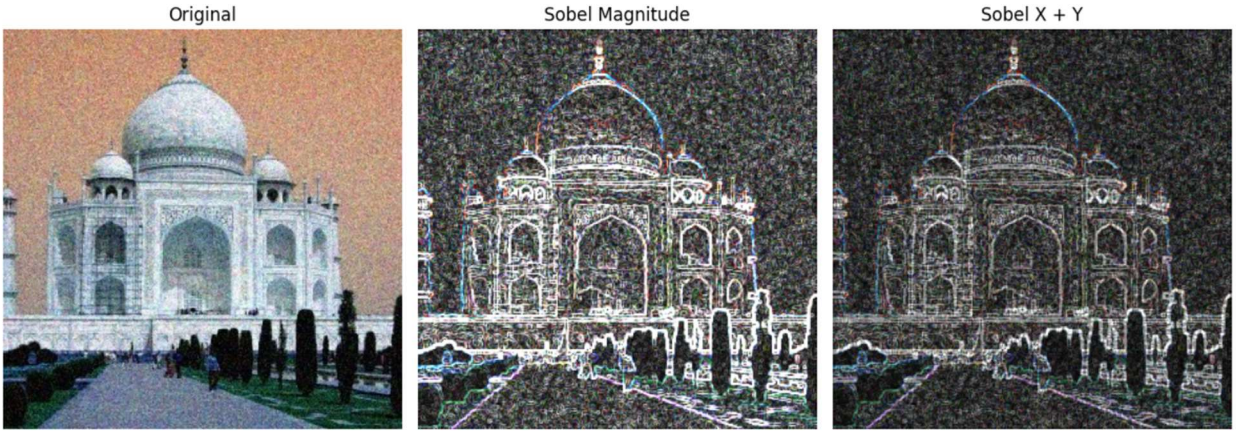
The **Sobel Operator** is an edge detection method used in image processing to find the gradient (rate of change) of image intensity at each pixel. It highlights regions of high spatial frequency that correspond to edges. It uses two small 3x3 convolution kernels (filters):

- One to detect changes in the horizontal direction (Gx)

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

- One to detect changes in the vertical direction (Gy)

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$



Prewitt Operator:

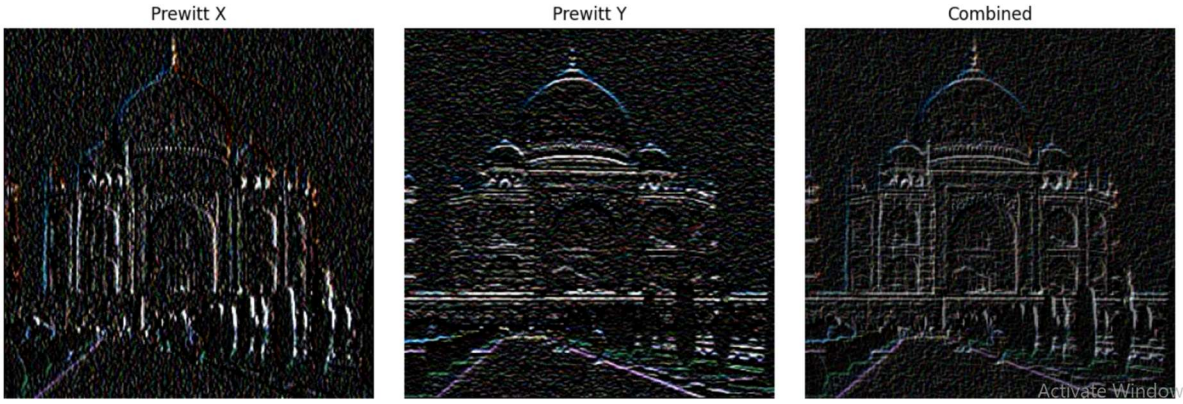
The Prewitt operator is similar to the Sobel operator. It's used to estimate the gradient (edges) of the image intensity function. Like Sobel, it uses two fixed 3×3 convolution kernels to approximate the derivatives in the horizontal and vertical directions.

Horizontal Kernel (Gx) — detects vertical **edges**:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

Vertical Kernel (Gy) — detects horizontal edges:

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$



Scharr Operator

The **Scharr operator** is an edge detection algorithm, similar to the **Sobel operator**, but it provides **better rotational symmetry** and **improved gradient approximation**, especially when using small kernel sizes (like 3×3).

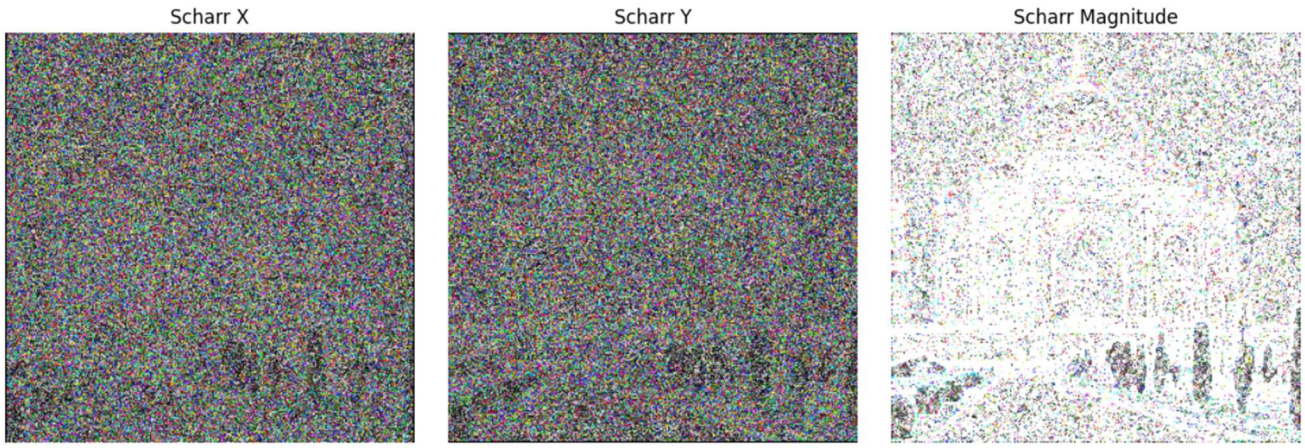
It is particularly useful when precise gradient computation is needed.

- **Horizontal (Gx):**

$$G_x = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix}$$

- **Vertical (Gy):**

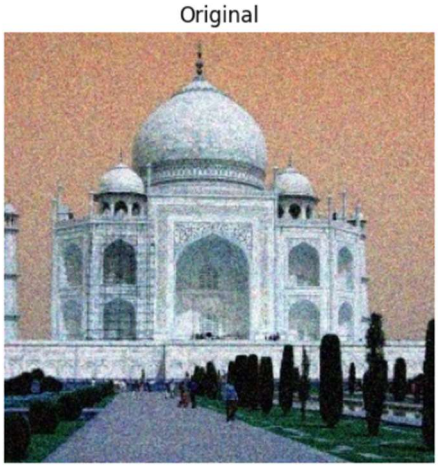
$$G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix}$$

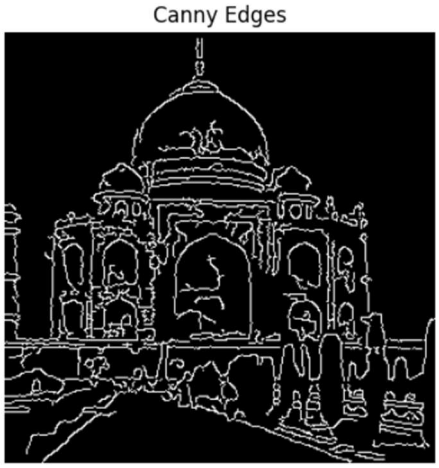


Canny Edge Detector

The Canny Edge Detector is a multi-stage algorithm that detects a wide range of edges in images. Unlike Sobel and Prewitt, which are based on convolution with fixed kernels, Canny is a more sophisticated pipeline involving several steps.

- Noise Reduction: Apply a Gaussian blur to reduce noise.
- Gradient Calculation: Compute intensity gradients using something like Sobel operators.
- Non-Maximum Suppression: Thin out the edges by keeping only the local maxima in the direction of the gradient.
- Double Thresholding:
- Edge Tracking by Hysteresis: Weak edges connected to strong edges are kept; others are discarded.





No Fixed kernel

Step	Purpose	Kernel Example
Gaussian Filtering	Noise reduction	5x5 Gaussian kernel
Gradient Calculation	Edge intensity/direction	Sobel Gx and Gy
Non-Max Suppression	Thinning edges	No kernel (logic-based)
Hysteresis Thresholding	Final edge linking	No kernel (logic-based)

Summary

Operator	Kernel Size	Noise Sensitivity	Edge Quality	Gradient Accuracy	Thin Edges	When to Use

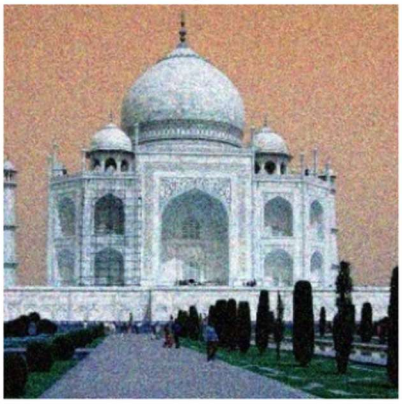
Sobel	3×3 (or more)	Medium	Good	Moderate	No	General-purpose; when moderate noise and reasonable edge quality are acceptable
Prewitt	3×3	High	Fair	Low	No	Lightweight tasks; fast prototyping; educational/demo use
Scharr	3×3 (fixed)	Medium	Better than Sobel	High	No	When high precision gradient computation is needed; better rotational symmetry
Canny	Multi-stage	Low (with Gaussian blur)	Excellent	Very High	Yes	Critical applications like object detection, medical imaging, and lane detection

Laplacian of Gaussian (LoG):

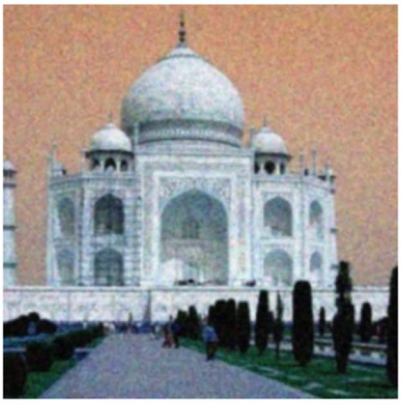
The Laplacian of Gaussian (LoG) is a second-order edge detection method that:

- Smooths the image using a Gaussian filter to reduce noise.
- Applies the Laplacian operator to detect regions of rapid intensity change (edges) by finding zero-crossings in the second derivative.

Original



Gaussian Blur



Laplacian of Gaussian (LoG)