

---

```

clear all; close all; clc;

% Read and Convert Image to Grayscale
I = imread('https://upload.wikimedia.org/wikipedia/en/7/7d/Lenna_%28test_image%29.png');
I_gray = rgb2gray(I);

% Sobel Edge Detection
sobel_x = [-1 0 1; -2 0 2; -1 0 1];
sobel_y = [-1 -2 -1; 0 0 0; 1 2 1];

gx = imfilter(double(I_gray), sobel_x, 'same');
gy = imfilter(double(I_gray), sobel_y, 'same');

% Magnitude of Gradient and Thresholding
BW1 = sqrt(gx.^2 + gy.^2) > 50;

% Prewitt Edge Detection
prewitt_x = [-1 0 1; -1 0 1; -1 0 1];
prewitt_y = [-1 -1 -1; 0 0 0; 1 1 1];

gx = imfilter(double(I_gray), prewitt_x, 'same');
gy = imfilter(double(I_gray), prewitt_y, 'same');
BW3 = sqrt(gx.^2 + gy.^2) > 50;

% Roberts Edge Detection
roberts_x = [1 0; 0 -1];
roberts_y = [0 1; -1 0];

gx = imfilter(double(I_gray), roberts_x, 'same');
gy = imfilter(double(I_gray), roberts_y, 'same');
BW4 = sqrt(gx.^2 + gy.^2) > 50;

% Laplacian of Gaussian (LoG) Edge Detection
sigma = 2;
size = 6 * sigma;
x = -size/2:size/2;
y = x;
[X, Y] = meshgrid(x, y);

% Manually defined LoG filter
LoG = (X.^2 + Y.^2 - 2*sigma^2) .* exp(-(X.^2 + Y.^2) / (2*sigma^2));

BW5 = imfilter(double(I_gray), LoG, 'same') > 0;

% Zero-Crossing Edge Detection
BW6 = edge(I_gray, 'zerocross');

% Canny Edge Detection with Post-processing
gx = imfilter(double(I_gray), sobel_x, 'same');
gy = imfilter(double(I_gray), sobel_y, 'same');
gmag = sqrt(gx.^2 + gy.^2);

```

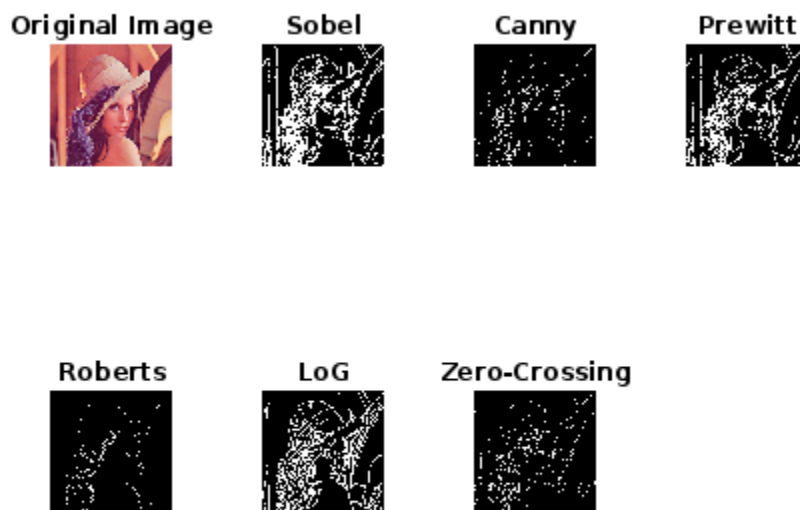
---

---

```
% Normalize gradient magnitude and apply threshold
BW2 = gmag > 0.1 * max(gmag(:));

% Morphological operations: Thin and remove small objects
BW2 = bwmorph(BW2, 'thin', Inf);
BW2 = bwareaopen(BW2, 30);

% Display Results
tiledlayout(2,4);
nexttile; imshow(I); title('Original Image');
nexttile; imshow(BW1); title('Sobel');
nexttile; imshow(BW2); title('Canny');
nexttile; imshow(BW3); title('Prewitt');
nexttile; imshow(BW4); title('Roberts');
nexttile; imshow(BW5); title('LoG');
nexttile; imshow(BW6); title('Zero-Crossing');
```



*Published with MATLAB® R2024b*