Practical 8

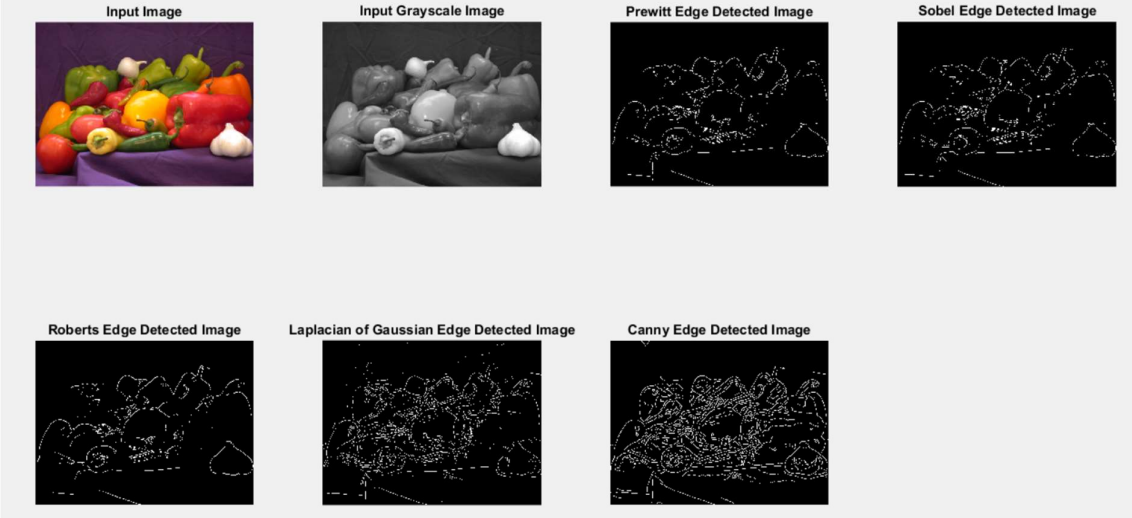
Aim: Write a program for edge detection.

(Apply Robert Operator, Prewitt Operator, Sobel Operator, Laplacian of Gaussion Operator and Canny Edge Detection methods to perform Edge Detection)

Code:

```
fprintf('92000103073 Raj Chhadia');
% Displaying Input Image
subplot(2,4,1);
    input_image = imread('peppers.png');
    imshow(input_image);
    title('Input Image');

% Convert the RGB image to the grayscale image
subplot(2,4,2);
    input_image = rgb2gray(input_image);
    imshow(input_image);
    title('Input Grayscale Image');

%Apply prewitt Operator
subplot(2,4,3);
    filtered_image1 = edge(input_image, 'prewitt');
    imshow(filtered_image1);
    title('Prewitt Edge Detected Image');

%Apply Sobel Operator
subplot(2,4,4);
    filtered_image2 = edge(input_image, 'sobel');
    imshow(filtered_image2);
    title('Sobel Edge Detected Image');

%Apply Roberts Operator
subplot(2,4,5);
    filtered_image3 = edge(input_image, 'roberts');
    imshow(filtered_image3);
    title('Roberts Edge Detected Image');

%Apply 'Laplacian of Gaussian Operator
subplot(2,4,6);
    filtered_image4 = edge(input_image, 'log');
    imshow(filtered_image4);
    title('Laplacian of Gaussian Edge Detected Image');

%Apply canny edge Detection algorithm
subplot(2,4,7);
    filtered_image4 = edge(input_image, 'canny');
    imshow(filtered_image4);
    title('Canny Edge Detected Image');
```

Output:



Practical 9

Aim: Write a program for smoothening and sharpening for 8-bit color image.

(Apply Laplacian of Gaussion Operator to Perform Edge Detection)

Code:

```
fprintf('92000103073 Raj Chhadia');
subplot(2,3,1);
    % Edge detection using Laplacian Filter.
    k=imread('peppers.png');
    imshow(input_image); title('Input Image');

subplot(2,3,2);
    % Convert rgb image to grayscale.
    kl=rgb2gray(k);
    imshow(kl); title('Input Grayscale Image');

subplot(2,3,3);
    % Display the noisy image.
    NI =imnoise(kl, 'gaussian');
    imshow(NI); title('Noisy Image');

subplot(2,3,4)
```

```
17 -  subplot(2,3,4)
18        % Create the gaussian Filter.
19 -      GI=fspecial('gaussian', 5, 1);
20        % Define the Laplacian Filter.
21 -      Lap=[0 -1 0; -1 4 -1; 0 -1 0];
22        % Convolve the noisy image
23        % with Gaussian Filter first.
24 -      Output1=conv2(NI, GI, 'same');
25        % Display the Gaussian of noisy _ image.
26 -      imshow(uint8(Output1)); title('Apply Gaussian Filter');
27
28 -  subplot(2,3,5)
29        % Convolve the resultant
30        % image with Laplacian filter.
31 -      Output2=conv2(Output1, Lap, 'same');
32        % Display the Laplacian of Gaussian resultant image.
33 -      imshow(Output2); title('Apply Laplacian Operator');
```

Output: