



Lab Report 4

MATLAB Image Processing

Digital Image Processing CSE438

Section: 03

Semester: Spring-2025

Submitted To:

Md. Asif Khan Rifat

Lecturer

Department of Computer Science
and Engineering

Submitted By:

Suddip Paul Arnab

2022-1-60-356

Date of submission: 24 April 2025

1. **Sharpen** the following image by applying the following and find out which one is better:
 - a) **Unsharp Masking**
 - b) **High Boost Filtering**

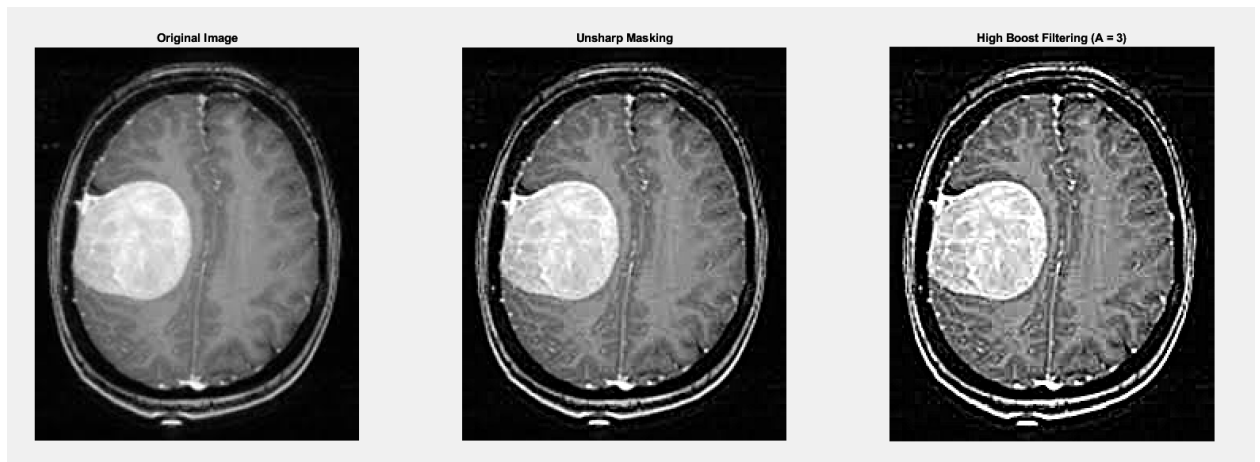
```
original_img = imread('Picture1.jpg');
original_img = im2double(original_img);

blurred_img = imgaussfilt(original_img, 2);

% a) Unsharp Masking
mask_unsharp = original_img - blurred_img;
unsharp_img = original_img + mask_unsharp;

% b) High Boost Filtering
A = 3;
highboost_img = original_img + A * mask_unsharp;

figure;
subplot(1, 3, 1), imshow(original_img), title('Original Image');
subplot(1, 3, 2), imshow(unsharp_img), title('Unsharp Masking');
subplot(1, 3, 3), imshow(highboost_img), title(['High Boost Filtering (A = ',
num2str(A), ')']);
```



2. **Sharpen** the following image using the concept of **Laplacian Filtering**.

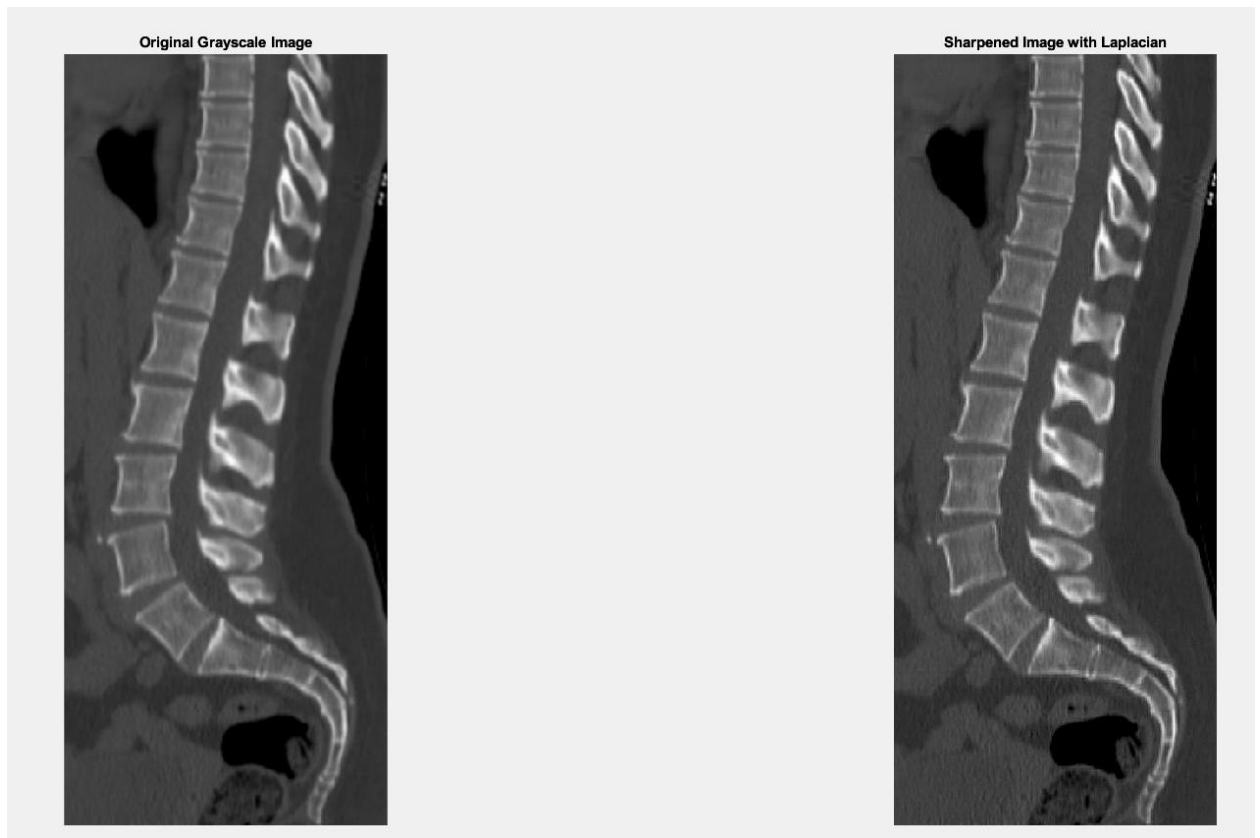
```
original_img = imread('Picture2.jpg');
gray_img = im2double(rgb2gray(original_img));

laplacian_filter = fspecial('laplacian', 0.9);

laplacian_img = imfilter(gray_img, laplacian_filter, 'replicate');

sharpened_img = gray_img - laplacian_img;
```

```
figure;
subplot(1, 2, 1), imshow(gray_img), title('Original Grayscale Image');
subplot(1, 2, 2), imshow(sharpened_img), title('Sharpened Image with Laplacian');
```



3. Use **Roberts-cross**, **Sobel**, and **Prewitt** operators to detect the edge of the following image.

```
img = imread('Picture3.jpg');
gray_img = rgb2gray(img);

roberts_edge = edge(gray_img, 'roberts');
sobel_edge = edge(gray_img, 'sobel');
prewitt_edge = edge(gray_img, 'prewitt');

figure;

subplot(2,2,1), imshow(gray_img), title('Original Grayscale Image');
subplot(2,2,2), imshow(roberts_edge), title('Roberts Edge Detection');
subplot(2,2,3), imshow(sobel_edge), title('Sobel Edge Detection');
subplot(2,2,4), imshow(prewitt_edge), title('Prewitt Edge Detection');
```



4. **Show** performance comparison among High Boost, Unsharp, Laplacian Roberts-cross, Sobel, Prewitt and Canny filtering for edge detection – find out which one is better for the given image.

```
original_img = imread('Picture3.jpg');
gray_img = im2double(rgb2gray(original_img));
```

```
%% 1. Unsharp Masking + Sobel
blurred = imgaussfilt(gray_img, 1);
mask = gray_img - blurred;
unsharp_img = gray_img + mask;
unsharp_sobel = edge(unsharp_img, 'sobel');
```

```
%% 2. High Boost Filtering + Sobel
k = 1.5; % High boost factor (>1)
high_boost_img = gray_img + k * mask;
highboost_sobel = edge(high_boost_img, 'sobel');
```

```
%% 3. Laplacian Filtering + Sobel
laplacian_filter = fspecial('laplacian', 0.2);
laplacian_img = imfilter(gray_img, laplacian_filter, 'replicate');
laplacian_sharpened = gray_img - laplacian_img;
laplacian_sobel = edge(laplacian_sharpened, 'sobel');
```

```
%% 4. Edge Detection Operators
roberts_edge = edge(gray_img, 'roberts');
```

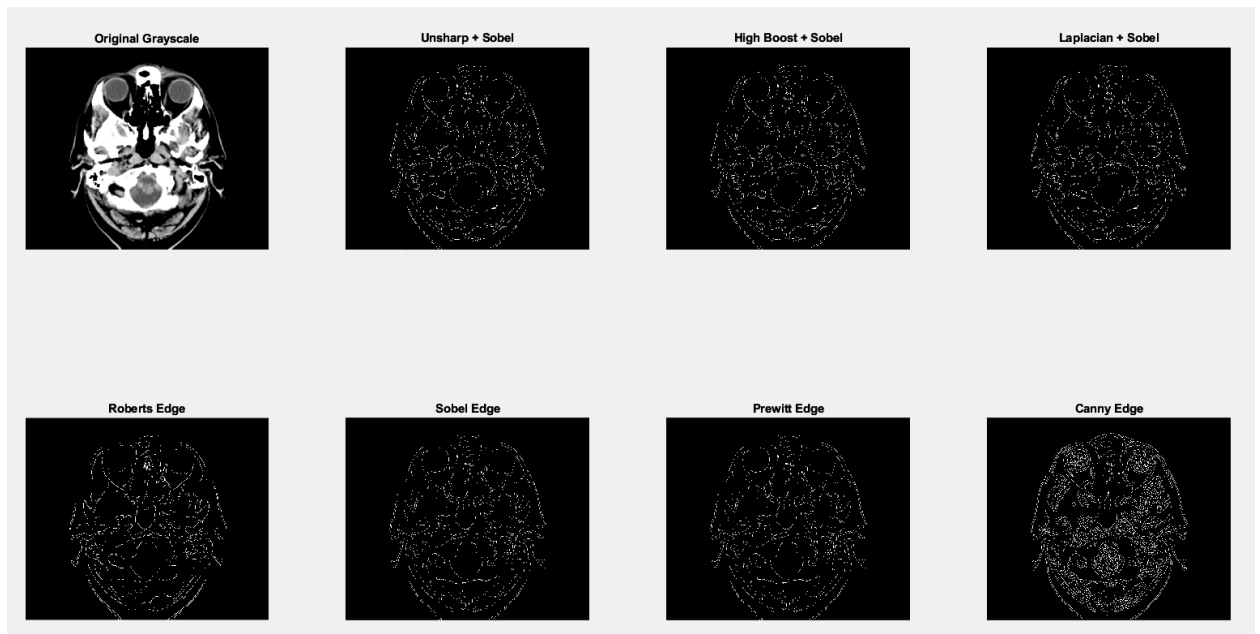
```

sobel_edge = edge(gray_img, 'sobel');
prewitt_edge = edge(gray_img, 'prewitt');
canny_edge = edge(gray_img, 'canny');

figure('Name','Edge Detection & Sharpening Comparison (4x2)',
'NumberTitle','off');

subplot(2,4,1), imshow(gray_img), title('Original Grayscale');
subplot(2,4,2), imshow(unsharp_sobel), title('Unsharp + Sobel');
subplot(2,4,3), imshow(highboost_sobel), title('High Boost + Sobel');
subplot(2,4,4), imshow(laplacian_sobel), title('Laplacian + Sobel');
subplot(2,4,5), imshow(roberts_edge), title('Roberts Edge');
subplot(2,4,6), imshow(sobel_edge), title('Sobel Edge');
subplot(2,4,7), imshow(prewitt_edge), title('Prewitt Edge');
subplot(2,4,8), imshow(canny_edge), title('Canny Edge');

```



According to outputs of image, we found that Canny Edge Detector is better for detecting edge because it is most detailed and noise resistant.