

Question 1

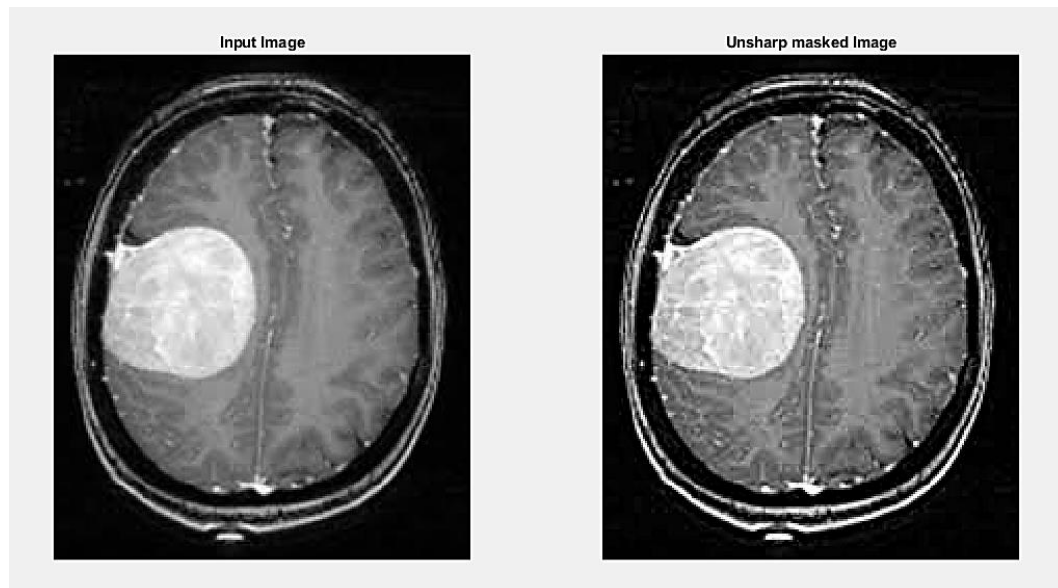
Sharpen the following image by applying the following:

Unsharp Masking

Code:

```
%Question 1 (a)
I1 = imread('tumor.jpg');
shrp1 = imsharpen(I1, "Radius", 2, "Amount", 1);
subplot(1,2,1);imshow(I1);title('Input Image')
subplot(1,2,2);imshow(shrp1);title('Unsharp masked Image')
|
```

Output:

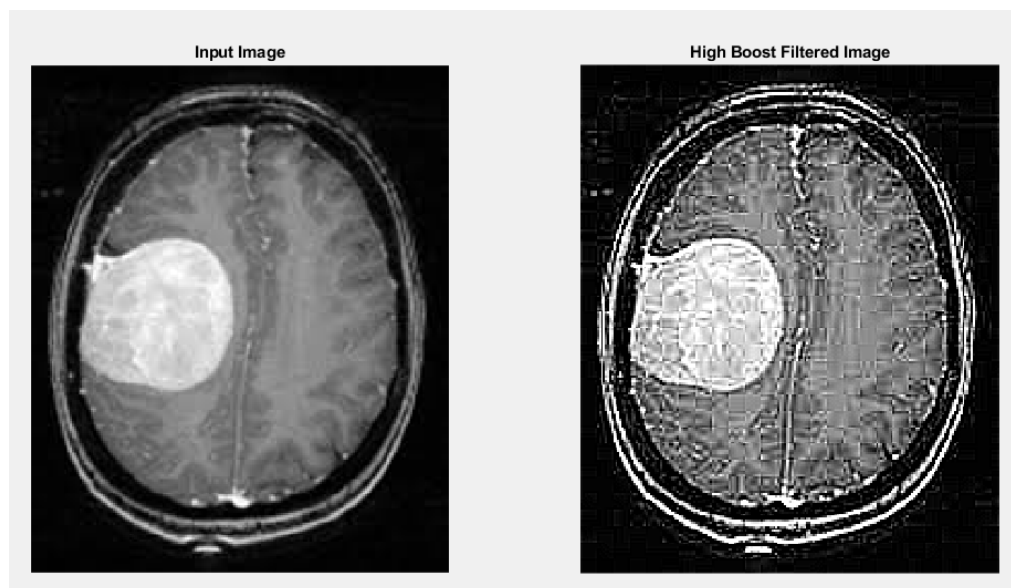


High Boost Filtering

Code:

```
%Question 1 (b)
laplacianKernel = [-1, -1, -1; -1, 8, -1; -1, -1, -1];
deltaFunction = [0, 0, 0; 0, 1, 0; 0, 0, 0];
scaleFactor = 1;
% Whatever... From 0 to 255, including fractional numbers like 0.5
kernel = laplacianKernel + scaleFactor * deltaFunction;
% Normalize so that the mean intensity doesn't change
kernel = kernel / sum(kernel(:));
% Filter the image
filteredImage = imfilter(I1, kernel);
subplot(1,2,1);imshow(I1);title('Input Image')
subplot(1,2,2);imshow(filteredImage);title('High Boost Filtered Image')
```

Output:



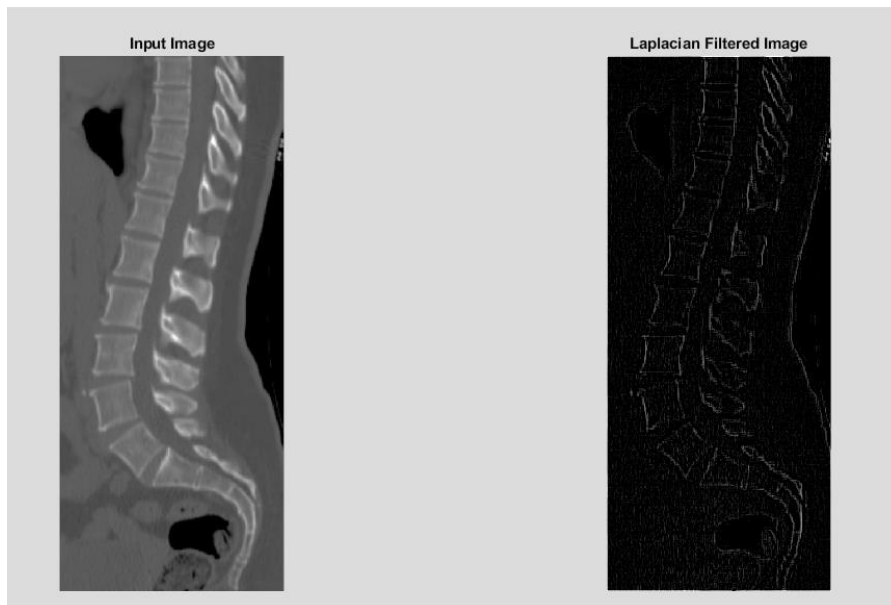
Question 2

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

Code:

```
%%  
%Question 2  
I2 = imread("Spine_CT.jpg");  
lap=[-1 -1 -1; -1 8 -1; -1 -1 -1];  
lap1 = imfilter(I2, lap);  
subplot(1,2,1);imshow(I2);title('Input Image')  
subplot(1,2,2);imshow(lap1);title('Laplacian Filtered Image')
```

Output:



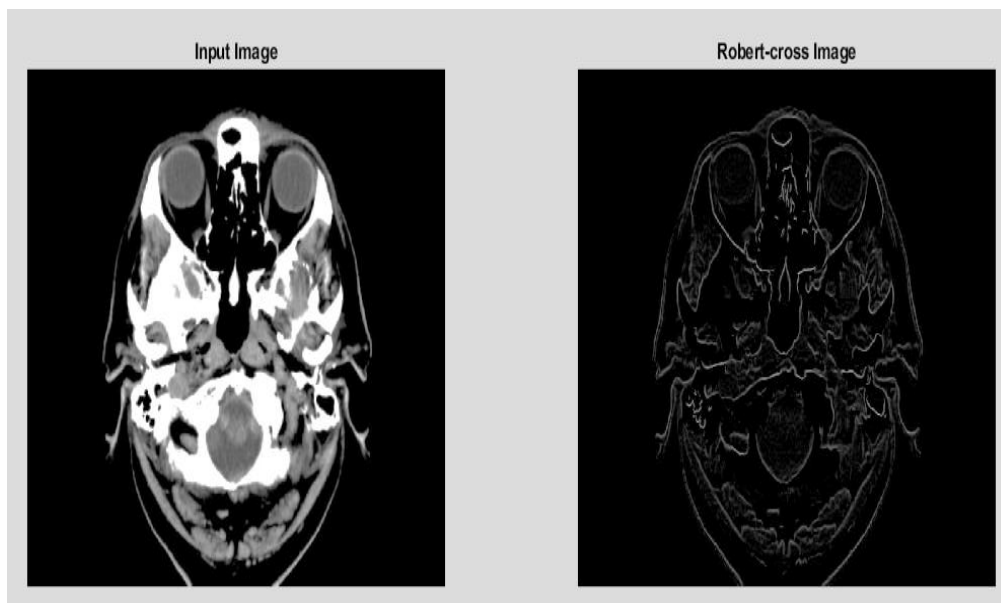
Question 3

Use **Roberts-cross** operators to detect the edge of the following image.

Code:

```
%%  
I3 = imread('Head_CT_Scan.jpg');  
  
x_mask = [1 0; 0 -1];  
y_mask = [0 1; -1 0];  
  
rb1 = imfilter(I3,x_mask, 'conv');  
rb2 = imfilter(I3,y_mask, 'conv');  
  
add1 = imadd(rb1, rb2);  
  
subplot(1,2,1);imshow(I3);title('Input Image')  
subplot(1,2,1);imshow(I3);title('Robert-cross Image')
```

Output:



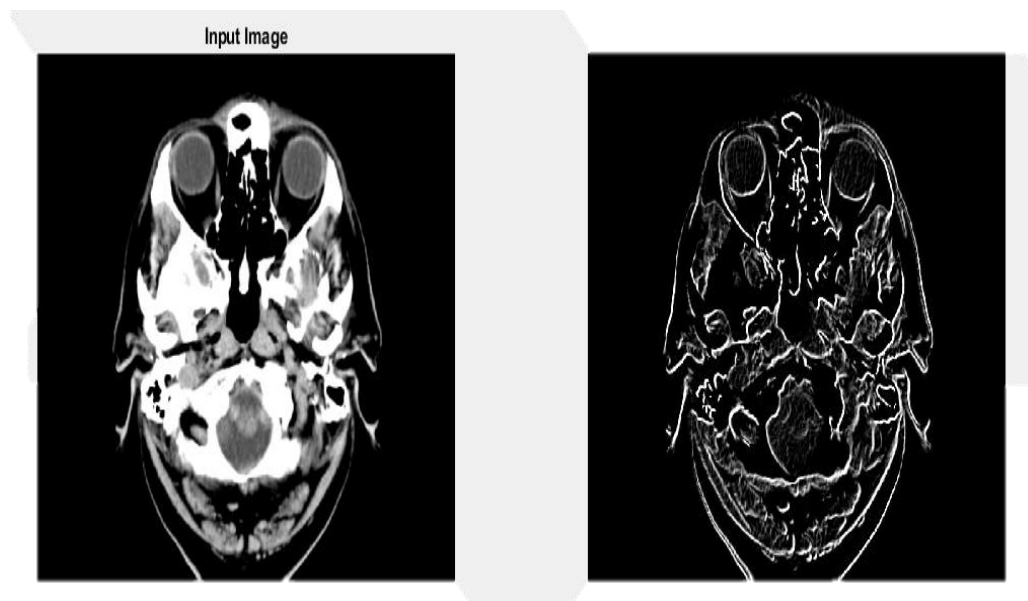
Question 4

Use **Sobel** operators to detect the edge of the image from problem 3.

Code:

```
%%  
Question 4  
x_mask = [-1 -2 -1; 0 0 0; 1 2 1];  
y_mask = [-1 0 1; -2 0 2; -1 0 1];  
  
sobel1 = imfilter(I3, x_mask, 'conv');  
sobel2 = imfilter(I3, y_mask, 'conv');  
  
add2 = imadd(sobel1, sobel2);  
  
subplot(1,2,1);imshow(I3);title('Input Image')  
subplot(1,2,1);imshow(I3);title('Robert-cross Image')
```

Output:



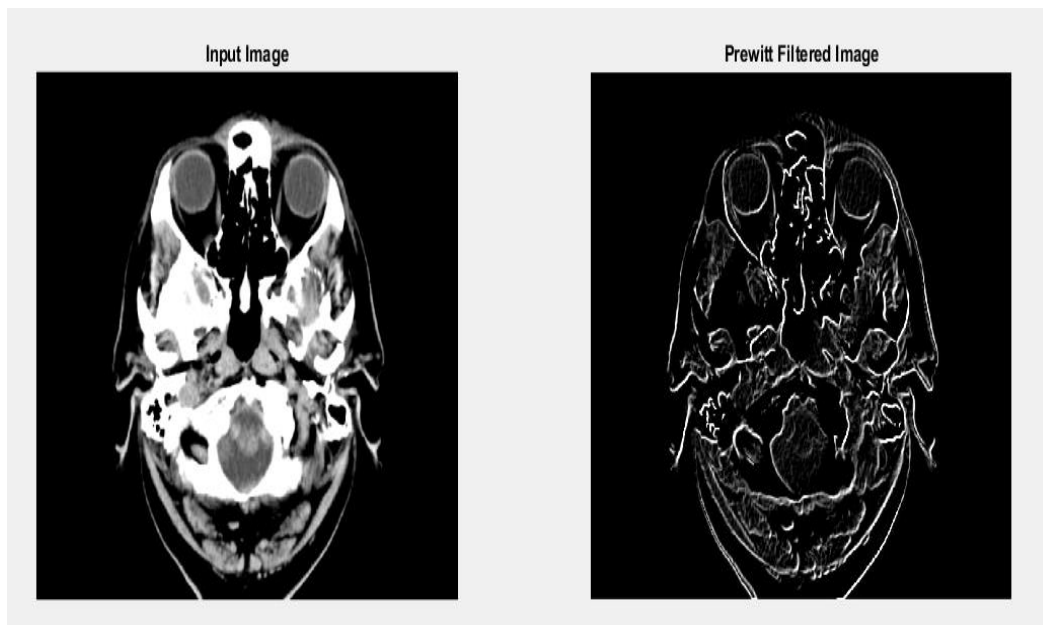
Question 5

Use **Prewitt** operators to detect the edge of the image from problem 3.

Code:

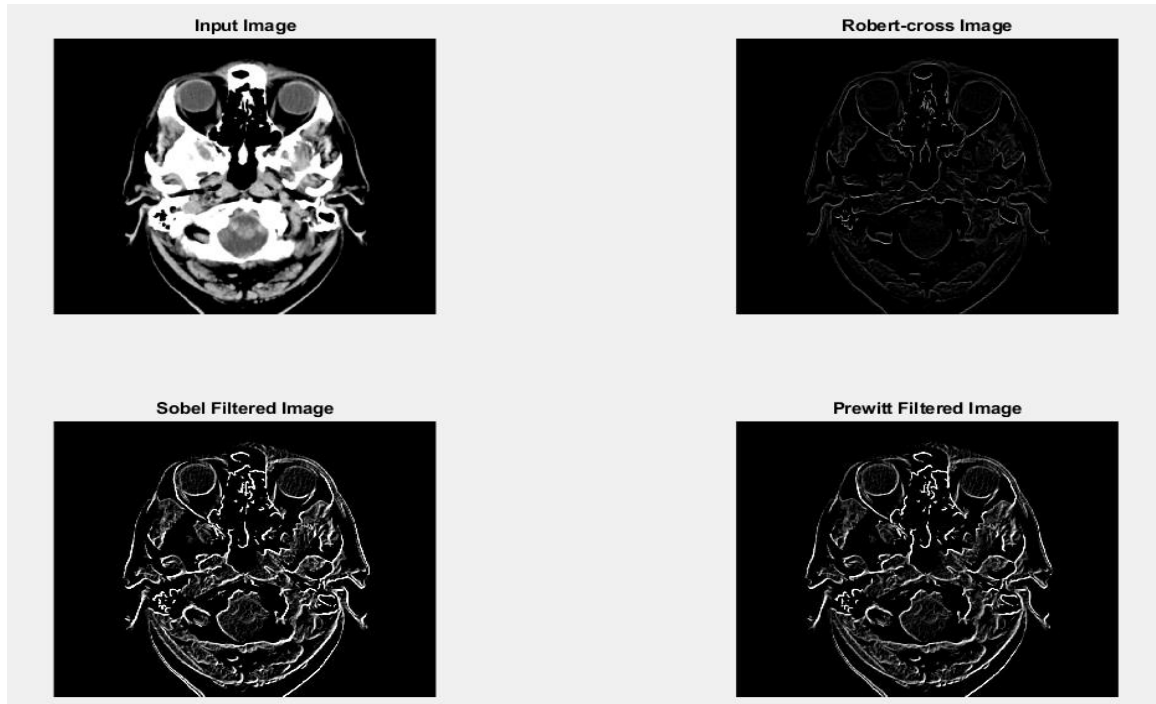
```
%%  
%Question 5  
  
x_mask = [-1 -1 -1; 0 0 0; 1 1 1];  
y_mask = [-1 0 1; -1 0 1; -1 0 1];  
  
pre1 = imfilter(I3, x_mask, 'conv');  
pre2 = imfilter(I3, y_mask, 'conv');  
  
add3 = imadd(sobel1, sobel2);  
  
subplot(1,2,1);imshow(I3);title('Input Image')  
subplot(1,2,1);imshow(I3);title('Robert-cross Image')
```

Output:



Question 6

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



As we can see from the output, the Prewitt operator performs better in my opinion if we want to extract edges. As a result, the center region is smoothed while the edges are highlighted. It helps with improved object recognition.