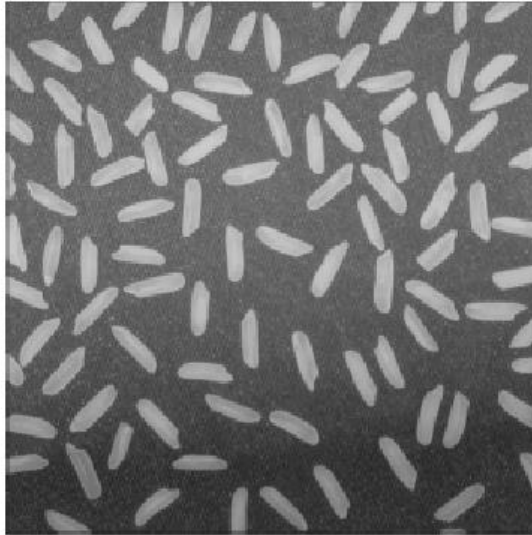


Elif KIR
Digital Image Processing
Assignment 3

```
%%Q1%%  
%%a%%  
img = imread('rice.tif');  
imshow(img)
```

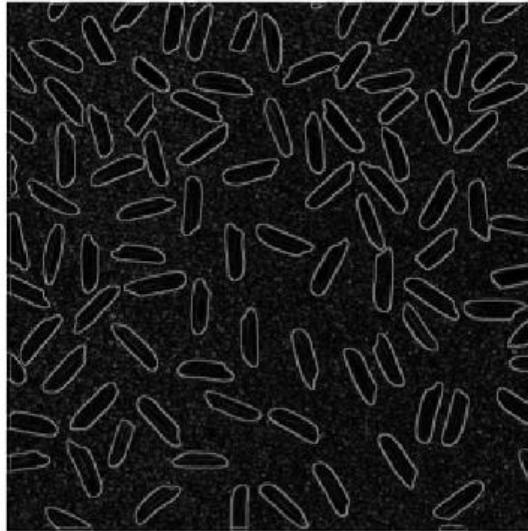


```
H2=[-1,-2,-1;0,0,0;1,2,1]
```

```
H2 = 3x3  
    -1    -2    -1  
     0     0     0  
     1     2     1
```

```
H2_v=imfilter(im2double(img),H2,'conv');  
H2_h=imfilter(im2double(img),H2',"conv");  
H2_F=(abs(H2_h)+abs(H2_v));  
H2_F=round(uint8((H2_F-min(H2_F(:)))/(max(H2_F(:))-min(H2_F(:)))*256));  
figure;  
imshow(H2_F); title ('Sobel operator Magnitude')
```

Sobel operator Magnitude



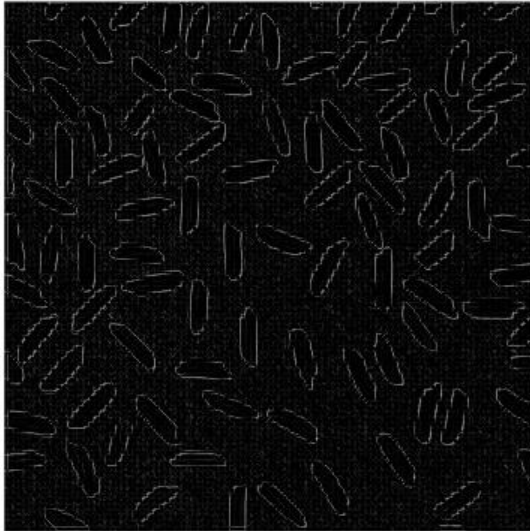
```
%%b%%
```

```
H=[-1,0;0 1],
```

```
H = 2x2
    -1     0
     0     1
```

```
H1_v=conv2(img, H, 'same');
H1_h=conv2(img, H', 'same');
H1_F=sqrt(H1_v.^2+H1_h.^2);
H1_F=round(uint8((H1_F-min(H1_F(:)))/(max(H1_F(:))-min(H1_F(:)))*256));
figure;
imshow(H1_F); title ('Robert operator Magnitude')
```

Robert operator Magnitude



```
figure;
```

```
%C%
```

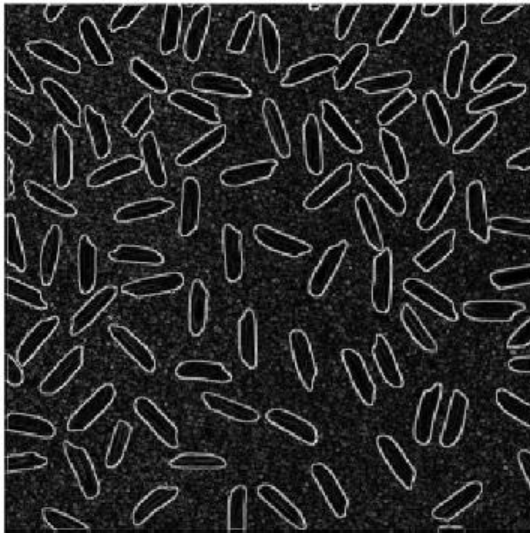
```
H4=[-1,-2,-1;0,0,0;1,2,1]
```

```
H4 = 3x3
```

```
    -1    -2    -1  
     0     0     0  
     1     2     1
```

```
H4_v=imfilter(im2double(img),H4,'replicate','full');  
H4_h=imfilter(im2double(img),H4',"replicate",'full');  
H4_F=(abs(H4_h)+abs(H4_v));  
H4_F(1,:)=[];  
H4_F(1,:)=[];  
H4_F(:,1)=[];  
H4_F(:,1)=[];  
H4_F=round(uint8((H4_F-min(H4_F(:)))/(max(H4_F(:))-min(H4_F(:)))*256));  
figure;  
imshow(H4_F); title ('Sobel operator Magnitude with replicate padding')
```

Sobel operator Magnitude with replicate padding

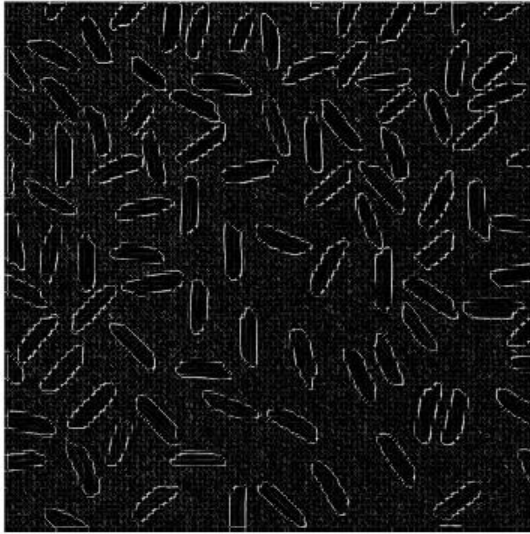


```
H3=[-1,0;0 1],
```

```
H3 = 2x2
     -1     0
      0     1
```

```
H3_v=imfilter(im2double(img),H3,'replicate','full');
H3_h=imfilter(im2double(img),H3','replicate','full');
H3_F=sqrt(H3_v.^2+H3_h.^2);
H3_F(1,:)=[];
H3_F(:,1)=[];
H3_F=round(uint8((H3_F-min(H3_F(:)))/(max(H3_F(:))-min(H3_F(:)))*256));
figure;
imshow(H3_F); title ('Robert operator Magnitude with replicate padding')
```

Robert operator Magnitude with replicate padding

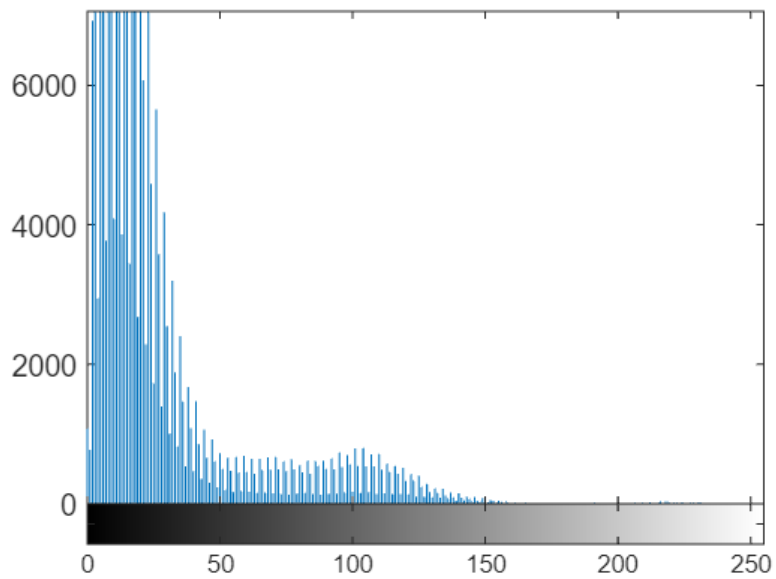


```
figure;
```

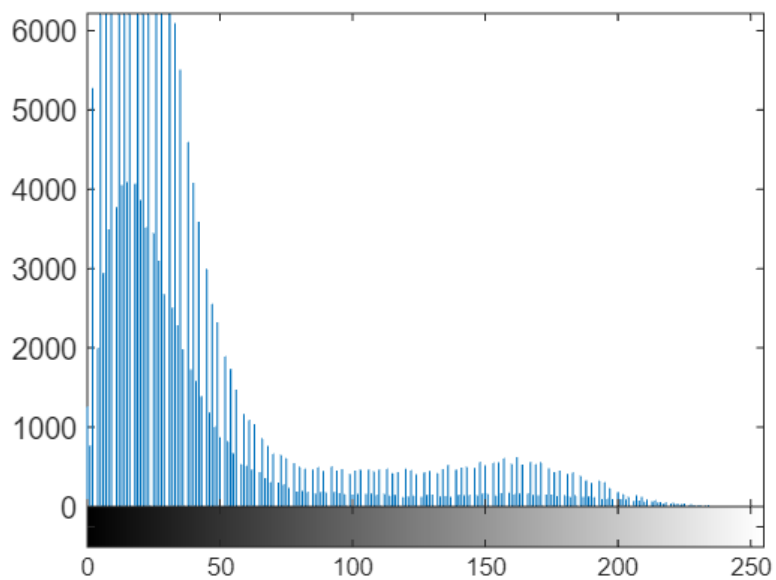
```
imhist(H2_F)
```

%When we use replicate padding instead of default zero padding, we saw that for both operator, it works better. While this type of padding preserves the variability in intense levels of original image so the contrast levels, zero padding affects the distribution of intensity level in favor of zero(darker).This reduce the contrast and so visibility in the output while we scaling back. We can see that with the help of histograms of outputs of two padding as below:

Elif KIR
Digital Image Processing
Assignment 3



```
imhist(H4_F)
```



```
%%Q2%%  
  
%%a%%  
%avg%  
Kernel=3;  
w= fspecial('average',Kernel);
```

Elif KIR
Digital Image Processing
Assignment 3

```
img2=imread('cameraman.tif');  
imshow(img2);
```



```
g1=imfilter(img2,w,'conv',0,'same');  
imshow(g1)
```



```
%median%  
g2=medfilt2(img2,[3 3]);
```

```
imshow(g2)
```



```
%%b%%  
%avg%  
Kernel2=9;  
w_9= fspecial('average',Kernel2);  
g3=imfilter(img2,w_9,'conv',0,'same');  
imshow(g3)
```



Elif KIR
Digital Image Processing
Assignment 3

```
%median%  
g4=medfilt2(img2,[9 9]);  
imshow(g4)
```



```
%%C%%  
  
w_1_9=9*w_9(1,:);  
w_9_1=9*w_9(:,1);  
g5=uint8(conv2(img2,w_1_9,"same"));  
g6=round(uint8(conv2(g5,w_9_1,"same")));  
imshow(g6)
```

Elif KIR
Digital Image Processing
Assignment 3



%Mean Square Error

```
MSE_avg_3 = immse(img2,g1) %3 by 3 avg
```

```
MSE_avg_3 = 198.2323
```

```
MSE_median_3 = immse(img2,g2) %3 by 3 median
```

```
MSE_median_3 = 123.6112
```

```
MSE_avg_9 = immse(img2,g3) %3 by 3 avg
```

```
MSE_avg_9 = 647.5050
```

```
MSE_median_9 = immse(img2,g4) %9 by 9 median
```

```
MSE_median_9 = 452.4152
```

```
MSE_2step_avg = immse(img2,g6) %2step average
```

```
MSE_2step_avg = 647.4932
```

```
MSEs =[MSE_avg_3, MSE_median_3, MSE_avg_9, MSE_median_9,MSE_2step_avg]
```

```
MSEs = 1x5
```

Elif KIR
Digital Image Processing
Assignment 3

```
198.2323 123.6112 647.5050 452.4152 647.4932
```

```
sorted_MSEs=sort(MSEs,"ascend")
```

```
sorted_MSEs = 1x5
```

```
123.6112 198.2323 452.4152 647.4932 647.5050
```

```
table(sorted_MSEs)
```

```
ans = table table
```

	sorted_MSEs				
1	123.6112	198.2323	452.4152	647.4932	647.5050

```
%%Q3%%
```

```
Im1=imnoise(img2,"gaussian",0,0.25);
```

```
Im2=imnoise(img2,"salt & pepper",0.35);
```

```
imshow(Im1)
```

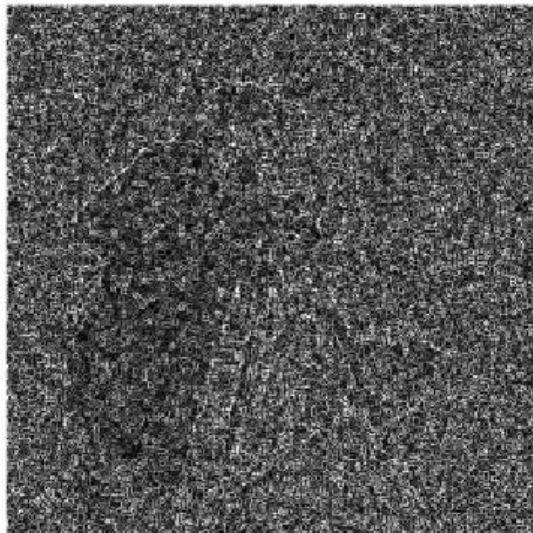


```
imshow(Im2)
```

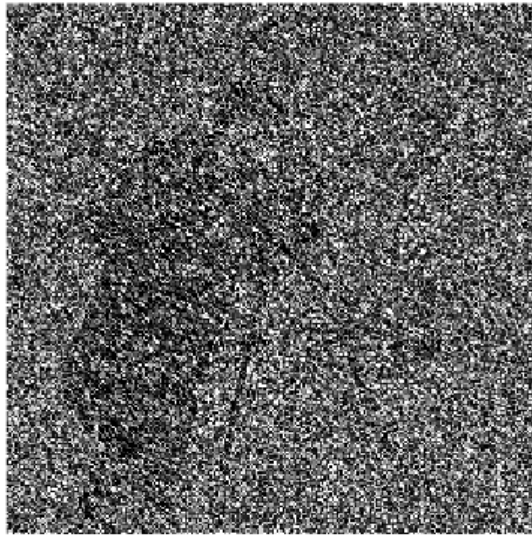


```
%%Q1a-gaussian%%  
H2_v=imfilter(im2double(Im1),H2,'conv');  
H2_h=imfilter(im2double(Im1),H2',"conv");  
H2_F=(abs(H2_h)+abs(H2_v));  
H2_F=round(uint8((H2_F-min(H2_F(:)))/(max(H2_F(:))-min(H2_F(:)))*256));  
figure;  
imshow(H2_F); title ('Sobel operator Magnitude with gaussian noise')
```

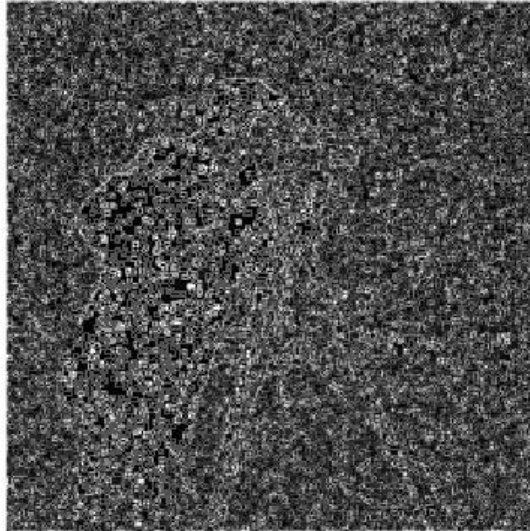
Sobel operator Magnitude with gaussian noise



```
%%Q1b-gaussian%%  
H1_v=conv2(Im1, H, 'same');  
H1_h=conv2(Im1, H, 'same');  
H1_F=sqrt(H1_v.^2+H1_h.^2);  
H1_F=round(uint8((H1_F-min(H1_F(:)))/(max(H1_F(:))-min(H1_F(:)))*256));  
figure;  
imshow(H1_F); title ('Robert operator Magnitude with gaussian noise')
```

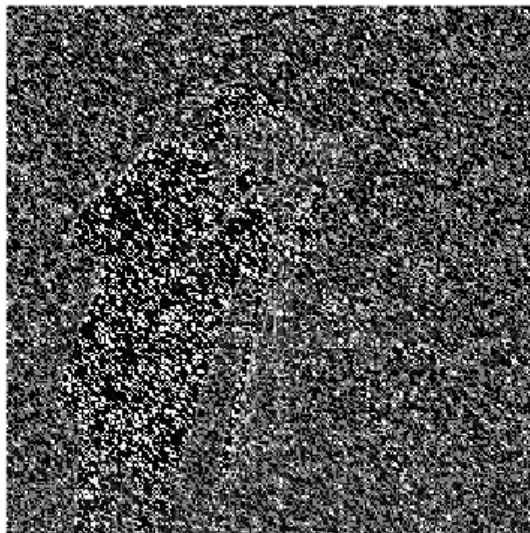


```
figure;  
  
%%Q1a-salt-pepper%%  
H2_v=imfilter(im2double(Im2),H2,'conv');  
H2_h=imfilter(im2double(Im2),H2',"conv");  
H2_F=(abs(H2_h)+abs(H2_v));  
H2_F=round(uint8((H2_F-min(H2_F(:)))/(max(H2_F(:))-min(H2_F(:)))*256));  
figure;  
imshow(H2_F); title ('Sobel operator Magnitude with salt&pepper noise')
```

```
%Q1b-salt-pepper%  
H1_v=conv2(Im2, H, 'same');  
H1_h=conv2(Im2, H, 'same');  
H1_F=sqrt(H1_v.^2+H1_h.^2);  
H1_F=round(uint8((H1_F-min(H1_F(:)))/(max(H1_F(:))-min(H1_F(:)))*256));  
figure;  
imshow(H1_F); title ('Robert operator Magnitude with salt&pepper noise')
```

Robert operator Magnitude with salt&pepper noise



```
figure;
```

Elif KIR
Digital Image Processing
Assignment 3

```
%%Q2a-gaussian%%  
%avg%  
g1=imfilter(Im1,w,'conv',0,'same');  
imshow(g1)
```



```
%median%  
g2=medfilt2(Im1,[3 3]);  
imshow(g2)
```

Elif KIR
Digital Image Processing
Assignment 3



```
%%Q2b-gaussian%  
%avg%  
g3=imfilter(Im1,w_9,'conv',0,'same');  
imshow(g3)
```



```
%median%  
g4=medfilt2(Im1,[9 9]);
```



```
imshow(g4)
```



```
%%Q2a-salt&pepper%%  
%avg%  
g1=imfilter(Im2,w,'conv',0,'same');  
imshow(g1)
```



Elif KIR
Digital Image Processing
Assignment 3

```
%median%  
g2=medfilt2(Im2,[3 3]);  
imshow(g2)
```



```
%%Q2b-salt&pepper%%  
%avg%  
g3=imfilter(Im2,w_9,'conv',0,'same');  
imshow(g3)
```



```
%median%  
g4=medfilt2(Im2,[9 9]);  
imshow(g4)
```

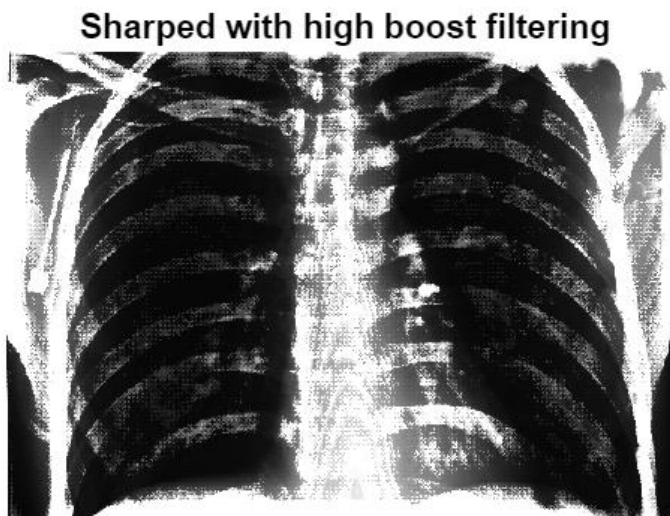


Comment: Obviously, median filter works better in salt&pepper noise.

```
%%Q4%%  
%%first technique%% (high boost filtering)  
img3=imread('chestXray-dark.tif');  
imshow(img3)
```

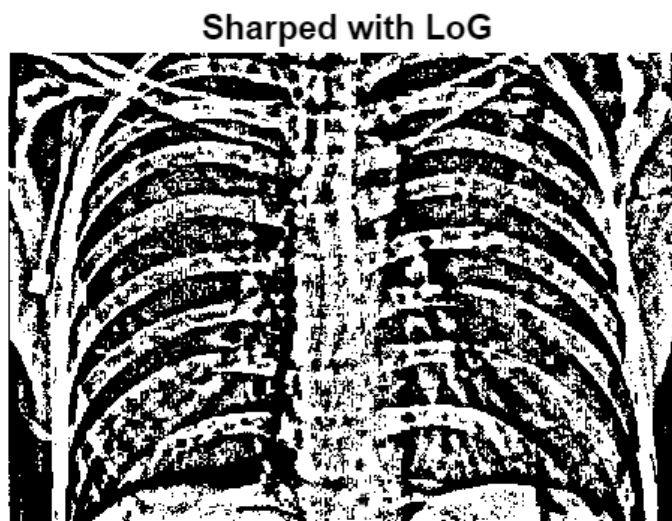


```
Kernel3=40;  
w_40= fspecial('average',Kernel3);  
img3_avg=imfilter(img3,w_40,'conv',0,'same');  
mask=img3-img3_avg;  
img3_boosted=img3+18*mask;  
imshow(img3_boosted);title('Sharped with high boost filtering')
```



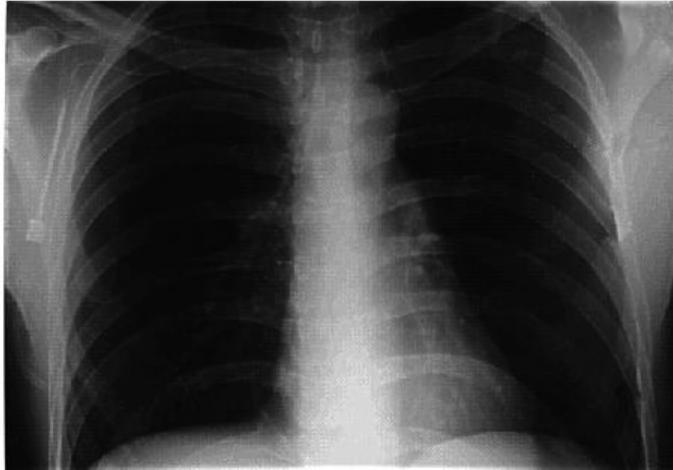
%%second technique% (Laplacian of Gaussian)

```
sigma=0.1;  
H =fspecial('log',20, sigma);  
H1=imfilter(img3,H,'replicate');  
figure;  
imshow(H1)  
title('Sharped with LoG')
```



```
%%third tecnique%% (Laplacian)  
a=ones(11);  
a(6,6)=-120;  
a=-a*1/121;  
H11=uint8(conv2(img3,a,"same"));  
H12=H11+img3;  
imshow(H12); title('Sharped with Laplacian')
```

Sharped with Laplacian



I think LoG and high-boost filtering worked better to sharp ribs and bones.