Digital Image Processing

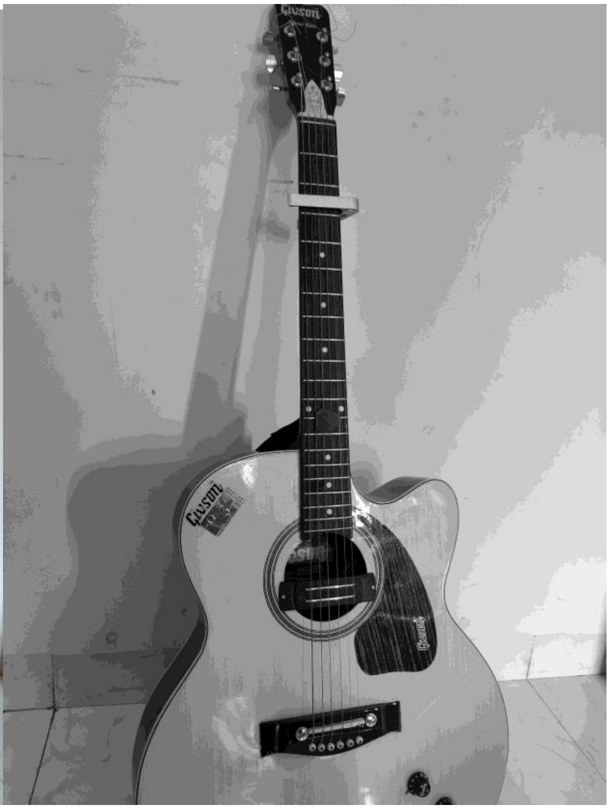
Assignment 1

SUBMITTED BY-:

Kshitij Chhabra 2017A8PS06916

RGB TO GREY CONVERSION

```
% Reading the Image and converting to Grey scale and Storing
image=imread('IMG_20210214_160213.jpg');
grayscale=rgb2gray(image);
imwrite(grayscale,'2017A8PS0691G.jpg');
```

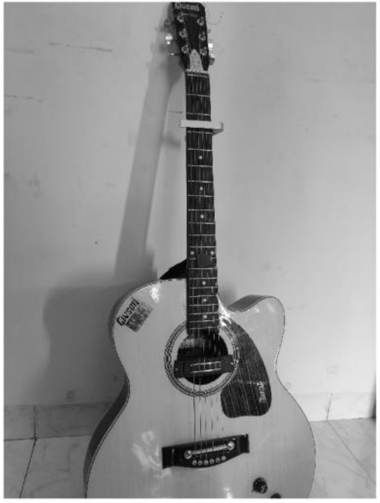


Question 1

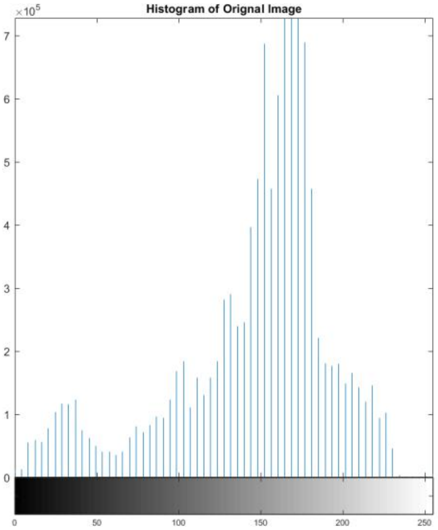
Code

```matlab
clc;
clear;
% Reading the grayScale Image
grayScale=imread('2017A8PS0691G.jpg');

% Plotting to compare the results

% Orignal Image
figure;
subplot(1,2,1);
imshow(grayScale),title("Orignal Image");
subplot(1,2,2);
imhist(grayScale,63),title("Histogram of Orignal Image");

% Equalising the image
Equalised_grayScale = histeq(grayScale);

% Equalised Image
figure;
subplot(1,2,1);
imshow(Equalised_grayScale),title("Equalised Image");
subplot(1,2,2);
imhist(Equalised_grayScale,64), title("Histogram of Equalised Image");
```

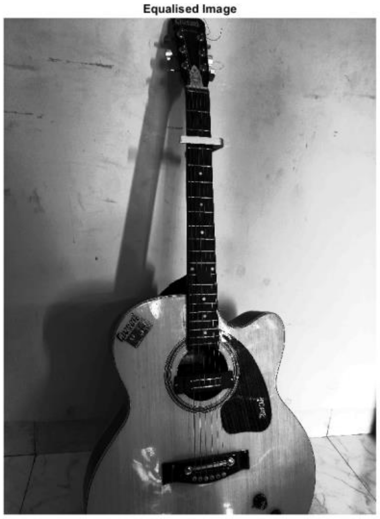
Orignal Image



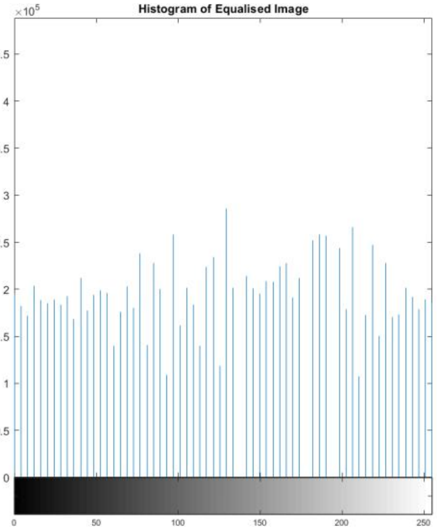
Histogram of Orignal Image



Equalised Image



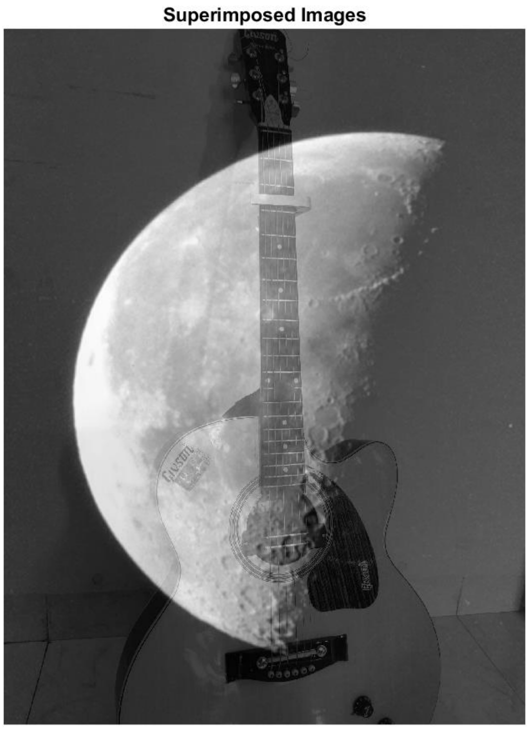
Histogram of Equalised Image



Question 2

Code

```
clc;
clear all;

% Loading the gray scale image and the moon image
grayScale=imread('2017A8PS0691G.jpg');
moon=imread('moon.tif');

% Resizing the moon image so that it is of same dimensions as
our object
% image
moon_resize=imresize(moon, [4000,3000]);
% Reducing the brightness of object for better appeal
grayScale=grayScale./2;

% Adding the two images
superimposed=imadd(grayScale,moon_resize,'uint16');
% Showing the output of two images
imshow(superimposed,[]),title("Superimposed Images");
```

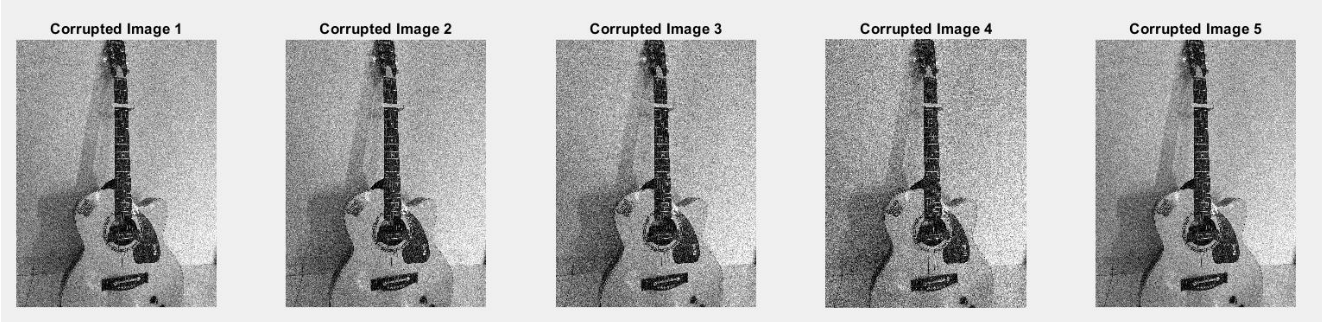


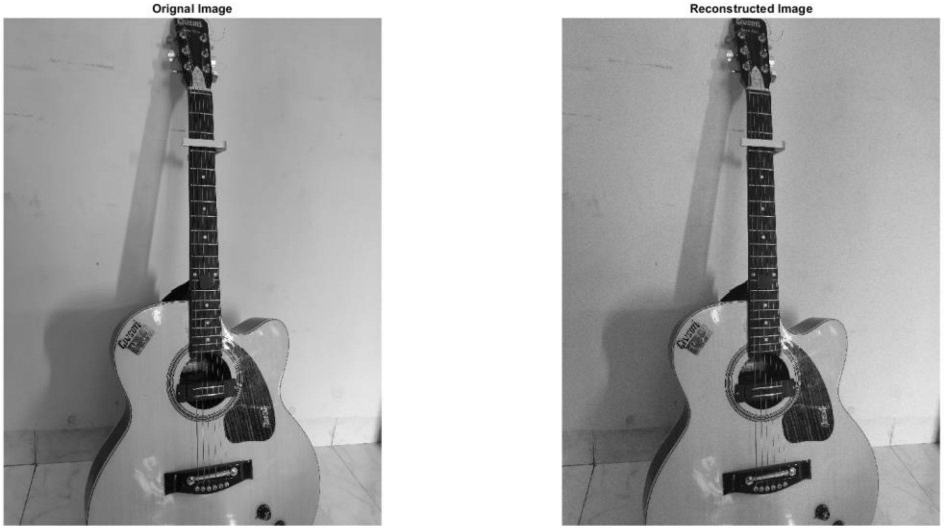
Question 3

Code

```
clc;
clear all;

% Loading the gray scale image
grayScale=imread('2017A8PS0691G.jpg');

% Corrupting the Image with gaussian noise with defaulalt
parameters
CorruptedImage1=imnoise(grayScale,'gaussian');
% Corrupting the Image with gaussian noise with mean 0 and
variance 0.012
CorruptedImage2=imnoise(grayScale,'gaussian',0,0.012);
% Corrupting the Image with gaussian noise with mean 0 and
variance 0.015
CorruptedImage3=imnoise(grayScale,'gaussian',0,0.015);
% Corrupting the Image with gaussian noise with mean 0 and
variance 0.02
CorruptedImage4=imnoise(grayScale,'gaussian',0,0.02);
% Corrupting the Image with gaussian noise with mean 0 and
variance 0.025
CorruptedImage5=imnoise(grayScale,'gaussian',0,0.025);


% Plotting the Corrupted Images
figure;
subplot(151)
imshow(CorruptedImage1),title("Corrupted Image 1");
subplot(152)
imshow(CorruptedImage2),title("Corrupted Image 2");
subplot(153)
imshow(CorruptedImage3),title("Corrupted Image 3");
subplot(154)
imshow(CorruptedImage4),title("Corrupted Image 4");
subplot(155)
imshow(CorruptedImage1),title("Corrupted Image 5");


% Reconstructing the Orignal Image by taking average across all
the images
ReconstructedImage_Raw=((CorruptedImage1./5)+(CorruptedImage2./5
)+(CorruptedImage3./5)+(CorruptedImage4./5)+(CorruptedImage5./5)
);
% Applying Wiener Filter
Im_Weiner=wiener2(ReconstructedImage_Raw);
```

```
% Comparing the Orignal Image to the Reconstructed Image
figure;
subplot(121)
imshow(grayScale),title("Orignal Image");
subplot(122)
imshow(Im_Weiner),title("Reconstructed Image");
```

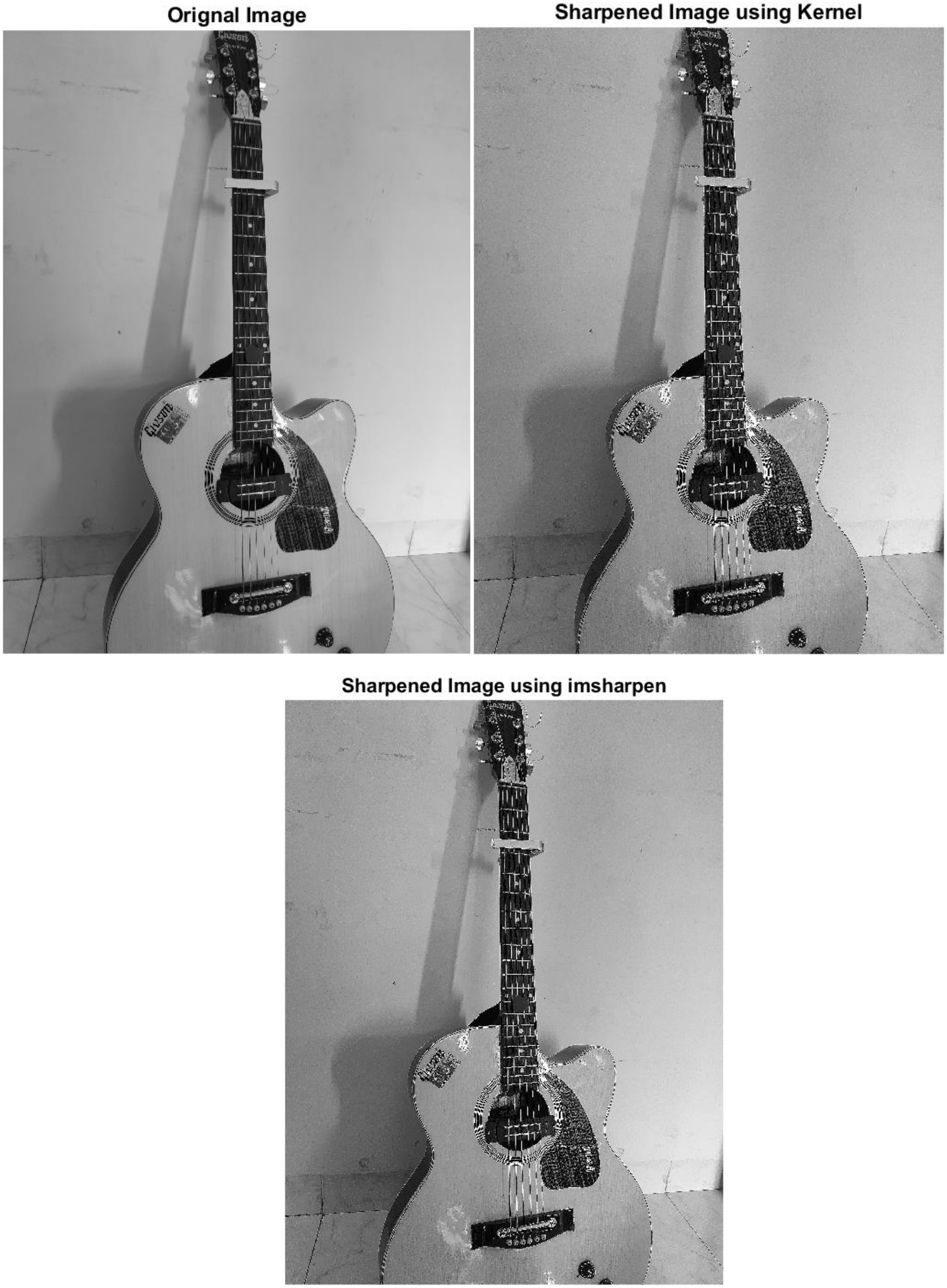




Question 4

Code

```
clc;
clear all;

% Loading the gray scale image
grayScale=imread('2017A8PS0691G.jpg');

% Unsharpening Mask
h=[-1 -1 -1;
    -1 9 -1;
    -1 -1 -1];

% Sharpening the image using the kernel
SharpenedImage_kernel=imfilter(grayScale,h,'replicate');
% Sharpening the image using matlab imsharpen command
SharpenedImage_fx=imsharpen(grayScale,'Radius',1.2,'Amount',8,'Threshold',0);


% Comparing the results
figure;
subplot(131);
imshow(grayScale),title("Orignal Image");
subplot(132)
imshow(SharpenedImage_kernel),title("Sharpened Image using Kernel");
subplot(133)
imshow(SharpenedImage_fx),title("Sharpened Image using imsharpen");
```



Orignal Image

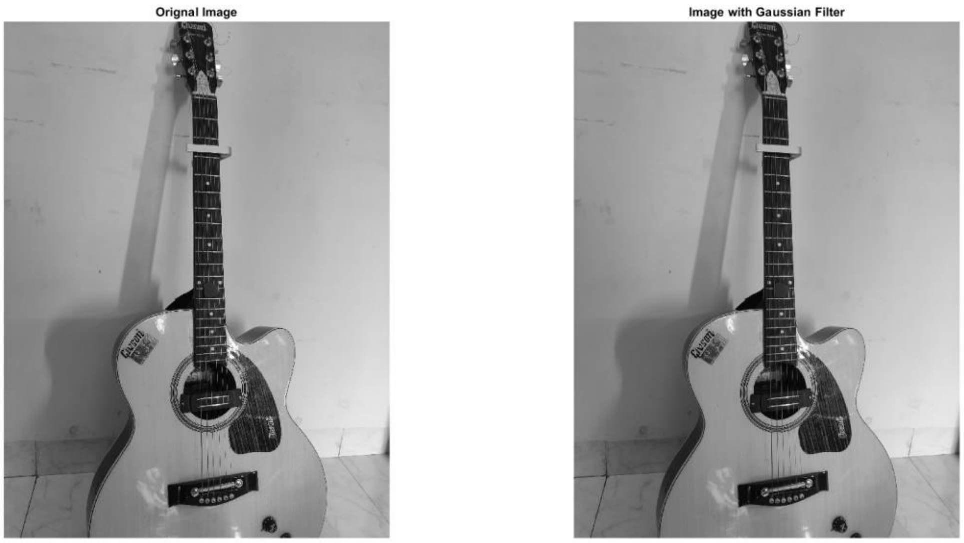
Sharpened Image using Kernel

Sharpened Image using imsharpen

Question 5

Code

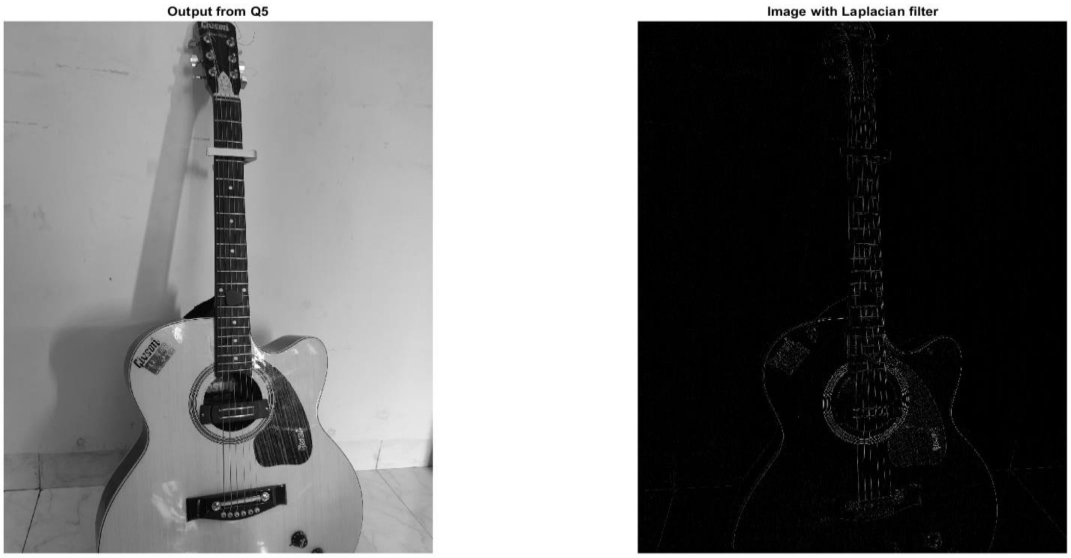
```
clc;
clear all;

% Loading the gray scale image
grayScale=imread('2017A8PS0691G.jpg');

% Defining the filter wit a 3x3 kernel
filter=fspecial('gaussian', [3 3],1.2);

% Applying the transformation
image_gaussian=imfilter(grayScale,filter,'replicate');

% Plotting and Comparing the Results
figure;
subplot(121);
imshow(grayScale),title("Orignal Image");
subplot(122);
imshow(image_gaussian),title("Image with Gaussian Filter");

%Saving the image
imwrite(image_gaussian,'image_gaussian.jpg')
```



Question 6

Code

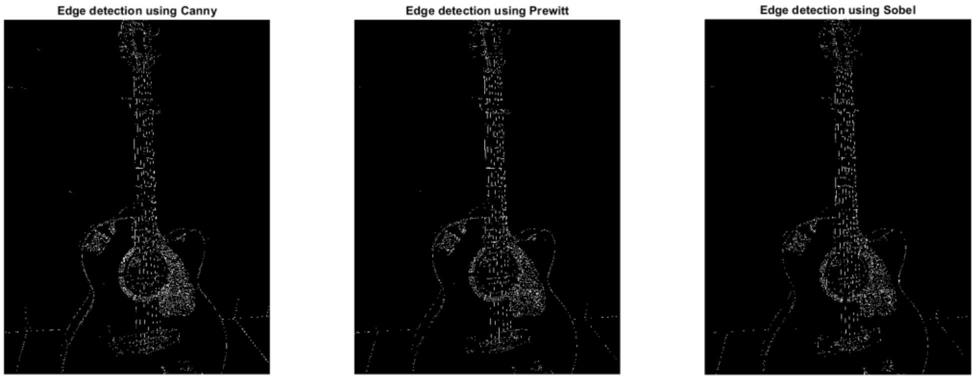
```
clc;
clear all;

% Loading the image from question 5
grayScale=imread('image_gaussian.jpg');

% Defining the Laplacian Filter
H = fspecial('laplacian',0.8);

% Applying the filter
im_edge=imfilter(grayScale,H,'replicate');

% Plotting and Comparing the Results
figure;
subplot(121);
imshow(grayScale),title("Output from Q5");
subplot(122);
imshow(im_edge,[]),title("Image with Laplacian filter");
```



Question 7

Code

```
clc;
clear all;

% Loading the gray scale image
grayScale=imread('2017A8PS0691G.jpg');

% Applying various edge detection techniques
CannyEdge=edge(grayScale,'Canny',0.1);
PreWittEdge=edge(grayScale,'prewitt');
SobelEdge=edge(grayScale,'sobel');

% Plotting to compare the results
figure;
subplot(131);
imshow(CannyEdge,[]),title("Edge detection using Canny");
subplot(132);
imshow(PreWittEdge,[]),title("Edge detection using Prewitt");
subplot(133);
imshow(SobelEdge,[]),title("Edge detection using Sobel");
```



Edge detection using Canny

Edge detection using Prewitt

Edge detection using Sobel

Question 8

Code

```
clc;
clear all;

% Loading the gray scale image
grayScale=imread('2017A8PS0691G.jpg');

% Embossing kernel
h=[-1,-1,-1,-1,0;
    -1,-1,-1,0,1;
    -1,-1,0,1,1;
    -1,0,1,1,1;
    0,1,1,1,1];

% Applying the embossing kernel
embossing=imfilter(grayScale,h,'replicate');

% Plotting and Comparing the Results
figure;
subplot(121);
imshow(grayScale),title("Orignal Image");
subplot(122);
imshow(embossing),title("Image with Embossing Kernel");
```