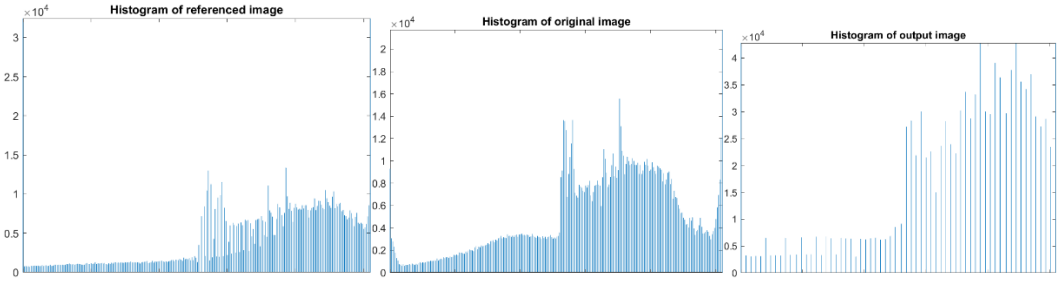
Lab 2

1. Adjust the histogram of the following image to match the reference image using histogram matching. Show the histogram of original, reference, and output images.

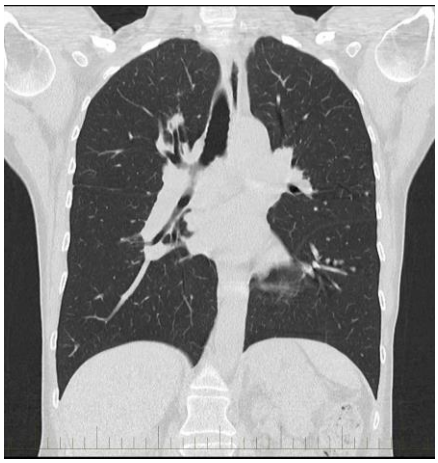
Code:

```
%Qus 1
img1 = imread('tree.png');
ref = imread('tree_reference.png');
res1 = imhistmatch(img1,ref);
figure,imhist(img1);title('Histogram of original image');
figure,imhist(ref);title('Histogram of referenced image');
figure,imhist(res1);title('Histogram of output image');
```

Output:



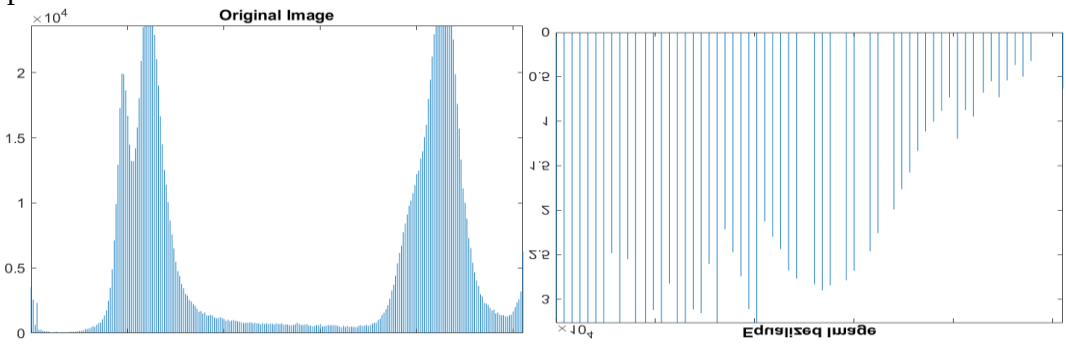
2. Change the contrast of the image using histogram equalization. Show the histogram of both input and output images.



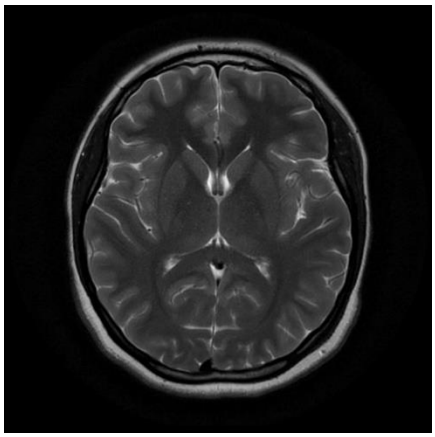
Code:

```
%Qus 2
img2 = imread('CT.jpg');
target = 256:-4:4;
heq1 = histeq(img2,target);
figure,imhist(img2);title('Original Image')
figure,imhist(heq1);title('Equalized Image')
```

Output:

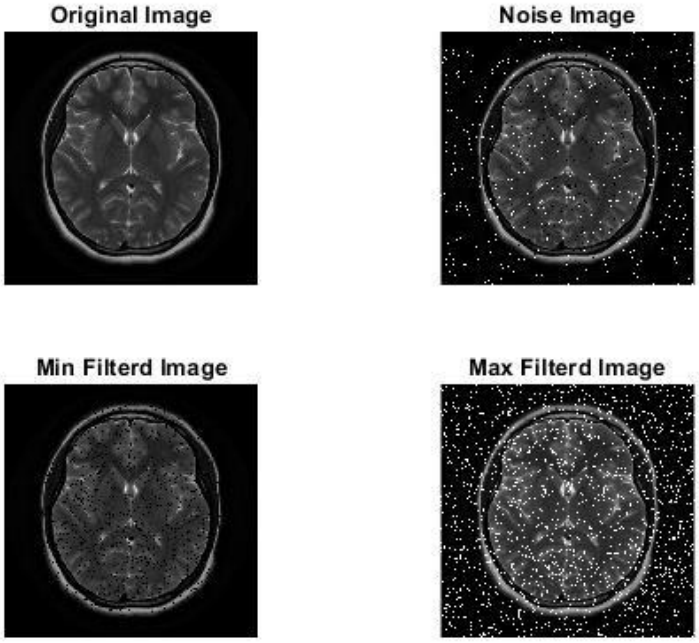


3. Apply salt and pepper noise to the following image and remove the noise using min and max filtering technique. Show input and output side by side.

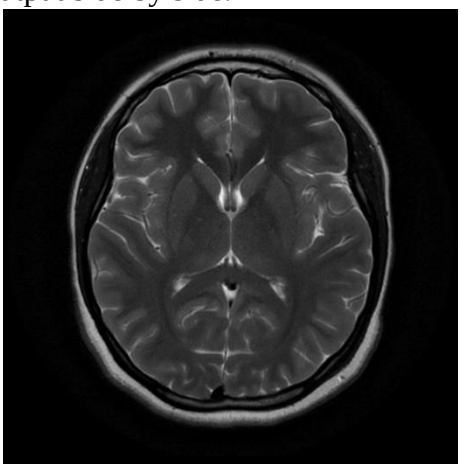


Code:

```
%Qus 3
img3 = imread('MRI.jpg');
noise1 = imnoise(img3,'salt & pepper',0.035);

minf = @(x) min(x(:));
min_image = nlfilter(noise1,[2,2],minf);

maxf = @(x) max(x(:));
max_image = nlfilter(noise1,[2,2],maxf);

subplot(2,2,1);imshow(img3);title('Original Image');
subplot(2,2,2);imshow(noise1);title('Noise Image');
subplot(2,2,3);imshow(min_image);title('Min Filterd Image');
subplot(2,2,4);imshow(max_image);title('Max Filterd Image');
```



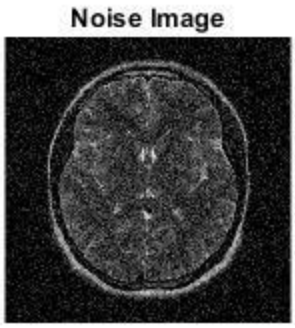
4. Apply Gaussian noise to the following image and remove the noise using Gaussian filtering. Show input and output side by side.

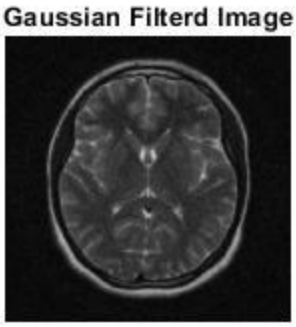


Code:

```
%Qus 4
noise2 = imnoise(img3,'gaussian',0.01);
guss1 =  imgaussfilt(noise2,2);
subplot(2,2,1);imshow(img3);title('Original Image');
subplot(2,2,2);imshow(noise2);title('Noise Image');
subplot(2,2,3);imshow(guss1);title('Gaussian Filterd Image');
```

Output:





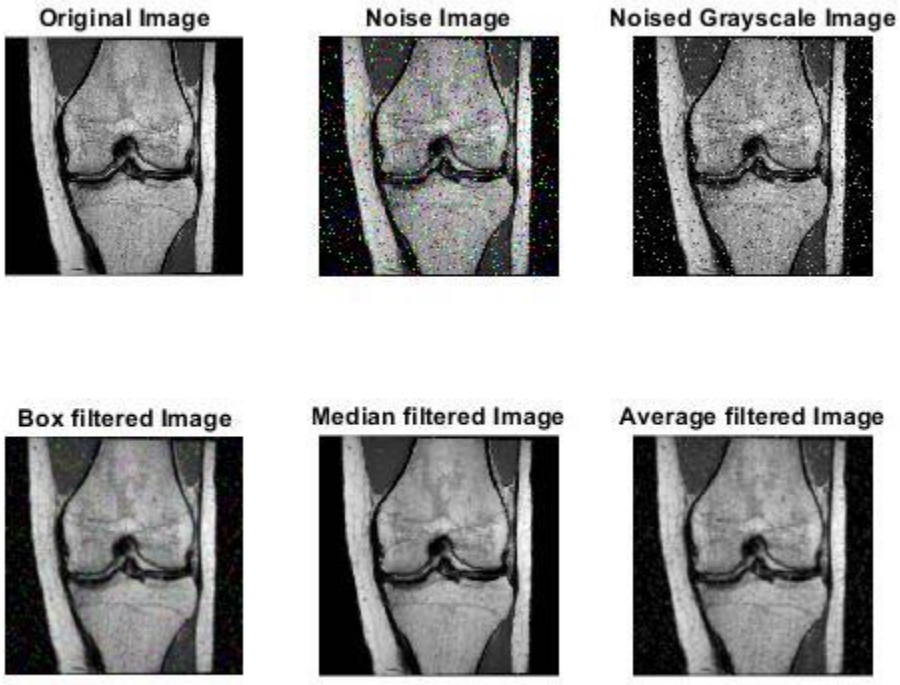
5. Apply any noise to the following image and restore it using:
    a) Box filtering
    b) Average filtering
    c) Median filtering

    Show input and output side by side. Also show the comparison between the 3 techniques. Mention which method works better than others.

Code:

```
%Qus 5
img4 = imread('MRI_2.jpg');
guss2 = imnoise(img4,'salt & pepper');
box = imboxfilt(guss2);
gray1 = rgb2gray(guss2);
med = medfilt2(gray1,[3,3]);
h = fspecial('average',3);
avg = imfilter(gray1,h);
subplot(2,3,1);imshow(img4);title('Original Image');
subplot(2,3,2);imshow(guss2);title('Noise Image');
subplot(2,3,3);imshow(gray1);title('Noised Grayscale Image');
subplot(2,3,4);imshow(box);title('Box filtered Image');
subplot(2,3,5);imshow(med);title('Median filtered Image');
subplot(2,3,6);imshow(avg);title('Average filtered Image');
```

Output:



6. Adjust the contrast of the following image.

Code:

```
% Qus 6
I5 = imread('i-5.png');
imshow(I5);title('Original Image');
G3 = rgb2gray(I5);
adj1 = imadjust(G3);
figure,imshow(adj1);title('Contrast Adjusted Image');
```

Output:

7. Brighten the following image

Code

```
% Qus 7
I6 = imread('i-6.png');
%subplot(2,1,1)
figure,imshow(I6);title('Original Image');
%subplot(2,1,2)
b1 = I6+75;
figure,imshow(b1);title('Brighten Image');
```

Output

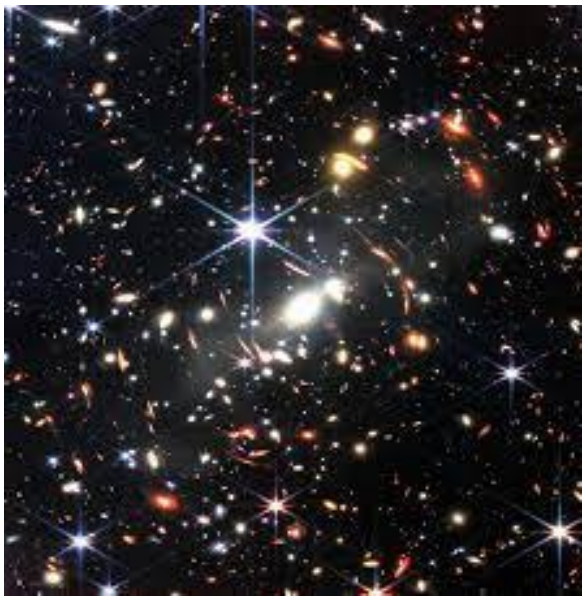
8. Quantize the Grayscale image by 8 levels.

Code:

```
% Qus - 8
I7 = imread('i-7.jpg');
figure,imshow(I7);title('Original Image');
thresh = multithresh(I7,7);
valuemax = [thresh max(I7(:))];
[quant8,index] = imquantize(I7,thresh,valuemax);
figure,imshow(quant8);title('Quantized by 8 levels');
```

Output:

9. Using the following image, solve questions a - f.



a) Read and show the image.
b) Show the matrix form of the image.
c) Show the pixel information by hovering the cursor on the image.
d) Find the value of the pixel (10, 78).
e) Show the size of the image.
f) Show the all the information of the image.

Codes:

```
% Ques - 9
% a)    Read and show the image.
I1 = imread('i-1.jpg');
s = imshow(I1);
%%
% b)    Show the matrix form of the image.
disp(I1);
%%
% c)    Show the pixel information by hovering the cursor on the image.
impixelinfo();
%%
% d)    Find the value of the pixel (10, 78).
r_v = I1(10,78,1);
g_v = I1(10,78,2);
b_v = I1(10,78,3);
sprintf('value of red, green, blue is %d %d %d',r_v,g_v,b_v)
%%
% e)    Show the size of the image.
size(I1)
%%
% f)    Show the all the information of the image.
imfinfo('i-1.jpg')
imageinfo(s)
```

Output:

a)

B)

24	25	5	14	40	25	62	60	13
17	23	12	10	23	20	55	54	18
13	16	16	9	6	9	19	27	24
17	10	13	9	5	0	6	16	19
15	10	8	6	12	4	19	22	8
10	13	7	0	10	8	27	17	0
7	10	2	0	2	3	12	4	6
6	4	0	3	0	0	0	0	16
7	0	6	5	0	7	10	8	11
8	10	0	0	29	72	44	6	0
7	0	0	39	115	137	65	0	0
4	0	37	108	132	86	29	14	3
4	1	46	85	54	4	4	28	0
4	0	12	15	7	0	8	7	0
5	10	2	15	47	24	0	0	0
12	7	0	51	119	53	0	11	15
36	28	8	42	68	29	2	4	14
94	74	20	0	5	0	0	3	0
105	92	40	2	0	6	20	21	0
30	30	16	5	3	5	16	17	24
2	2	4	6	4	0	10	16	23
4	7	5	1	0	0	0	5	4
4	8	9	1	0	0	0	0	0

c)

d)

```
ans =

    'value of red, green, blue is 21 22 24'
```

e)

```
ans =

   295   289     3
```

f)

```
          Filename: 'E:\438\labl\i-1.jpg'
       FileModDate: '01-Feb-2023 06:44:50'
          FileSize: 33971
            Format: 'jpg'
     FormatVersion: ''
             Width: 289
            Height: 295
          BitDepth: 24
         ColorType: 'truecolor'
   FormatSignature: ''
   NumberOfSamples: 3
      CodingMethod: 'Huffman'
     CodingProcess: 'Sequential'
           Comment: {}
```

10. Using the following images, solve questions a - i.

RGB Image

Grayscale Image

Indexed Image

a) Read and show all three types of images (RGB, Grayscale, and Indexed).
b) Turn the RGB image to Grayscale image.
c) Turn the Indexed image to Grayscale image.
d) Turn the Indexed image to RGB image.
e) Convert the Grayscale image to a Binary image.
f) Show the inverted form of that Binary image.
g) Show the histogram of the Grayscale image.

h) Invert the RGB image.
i) Blur the RGB image.

Codes:

```
% Ques - 10
% a)    Read and show all three types of images (RGB, Grayscale, and Indexed).
I2 = imread('i-2.jpg');
I3 = imread('i-3.jpg');
[I4,I4map] = imread('i-4.png');

figure, imshow(I2)
figure,imshow(I3)
figure,imshow(I4,I4map)
%%
% b)    Turn the RGB image to Grayscale image.
G1 = rgb2gray(I2)
figure,imshow(G1)
%%
% c)    Turn the Indexed image to Grayscale image.
G2 = ind2gray(I4,I4map);
figure,imshow(G2)
```

```
% d)    Turn the Indexed image to RGB image.
rgb1 = ind2rgb(I4,I4map);
figure,imshow(rgb1)
%%
% e)    Convert the Grayscale image to a Binary image.
bw1 = imbinarize(G2);
figure,imshow(bw1)
%%
% f)    Show the inverted form of that Binary image.
figure,imshow(~bw1)
%%
% g)    Show the histogram of the Grayscale image.
figure,imhist(G2);
%%
% h)    Invert the RGB image.
inv = imcomplement(rgb1);
figure,imshow(inv)
%%
% i)    Blur the RGB image.
ww=7;
kernel = ones(ww)/ww^2;
blur1 = imfilter(rgb1,kernel);
figure,imshow(blur1)
```

Output:

a)

b)

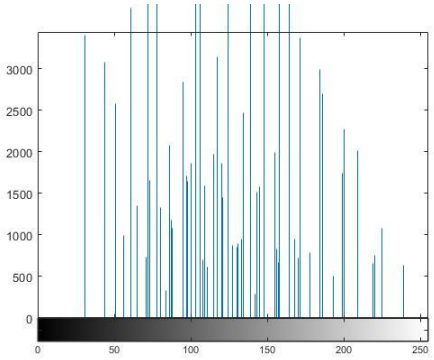
c)

d)

e)

f)

g)



h)

i)