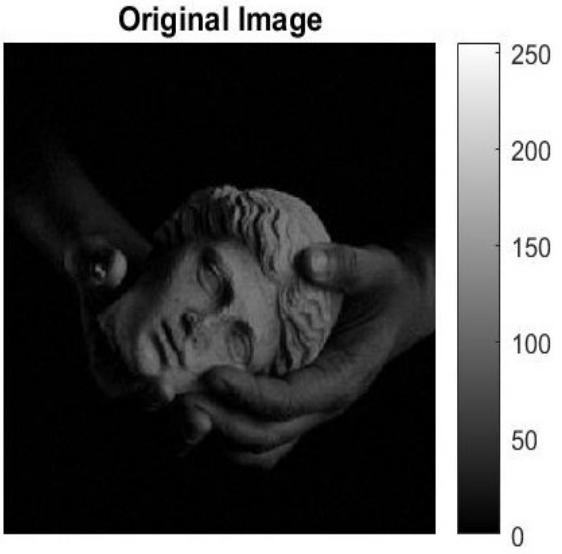
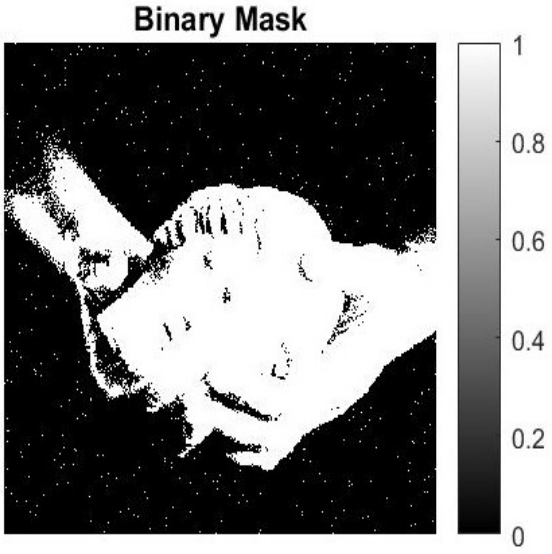
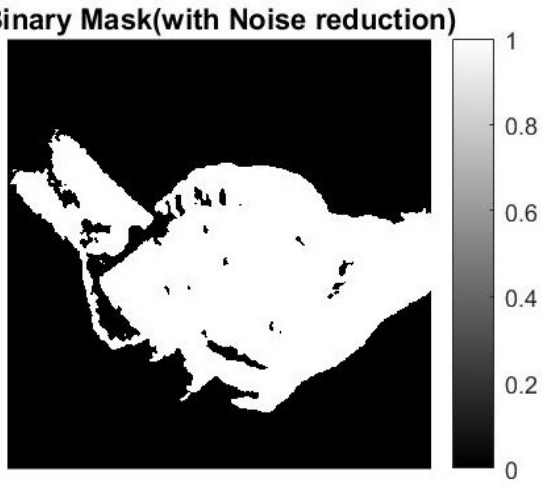
REPORT

Q2. A) Foreground Mask

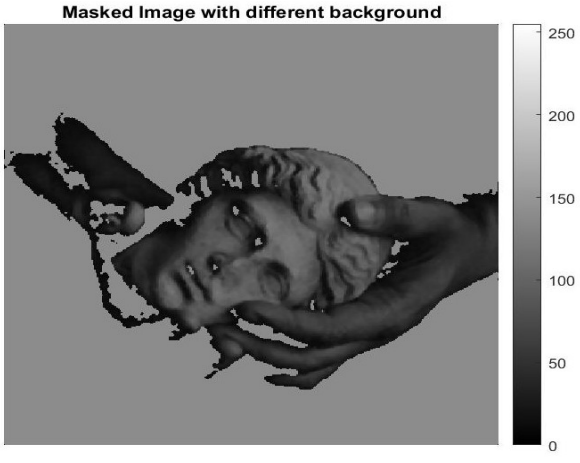




Binary Mask for 'statue.png'. Threshold intensity = 8



Threshold below 14 makes the binary mask image noisy. Hence, median filter has been applied twice for noise reduction



Here, the background has intensity 255 for the first image and 140 for the second.

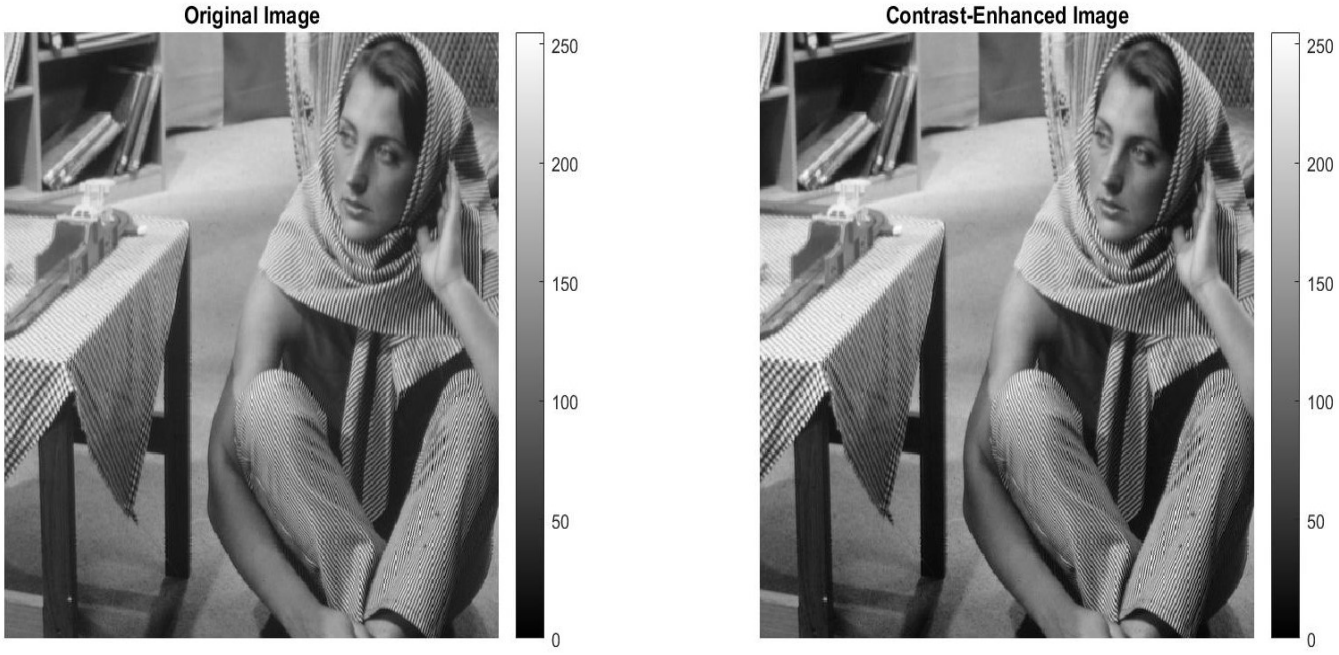
Q2. B) Linear Contrast Stretching

Pseudocode for the Linear Contrast Stretching Function to 'stretch' the range of intensity values it contains to span [0,255]:

$$out_i = \frac{(max_{new} - min_{new}) * (inp_i - min_{old})}{(max_{old} - min_{old})} + min_{new}$$

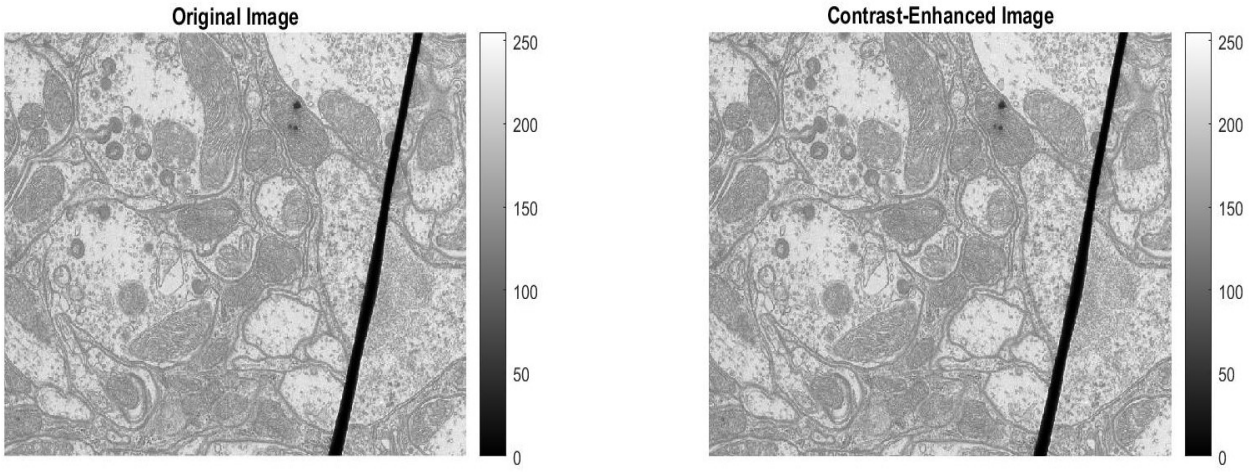
where,
out_i = *output intensity value*
inp_i = *input intensity value*
max_{old} = *maximum of all the intensity values of the channel*
min_{old} = *minimum of all the intensity values of the channel*
max_{new} = *maximum of the resulting intensity value (here max[0,255] = 255)*
min_{new} = *minimum of the resulting intensity value (here 0)*

Image 1- 'Barbara.png'



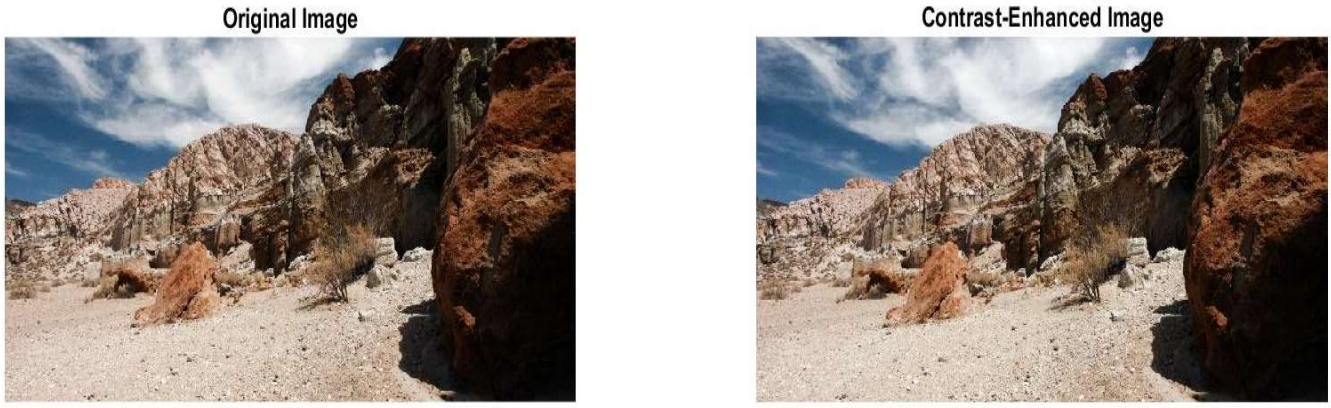
Here, Linear Contrast Stretching has improved the contrast a bit, by stretching the range from [12,246] to [0,255]

Image 2- 'TEM.png'



There is no difference since the image is already in the intensity range [0,255]

Image 3 – 'canyon.png'



There is no change since min and max value of actual and desired intensity range is almost same

Min_{old} intensity value across all channels = 0

Max_{old} intensity value: Channel 1=255, channel 2= 253, channel 3= 252.
Hence, there is no effect of Linear Contrast Stretching

Image 5: 'church.png'

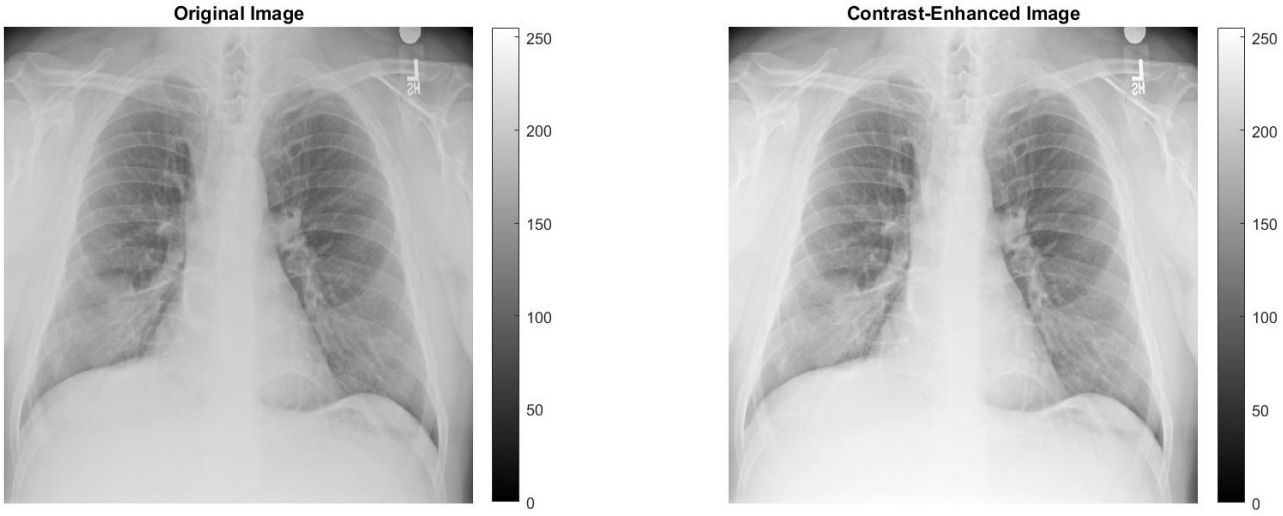
There is absolutely no change after applying Linear Contrast Stretching here because the range of actual intensity values is exactly same as the range of desired intensity values

$\mathbf{Max_{old}}$ intensity value across all channels = 255

$\mathbf{Min_{old}}$ intensity value across all channels = 0

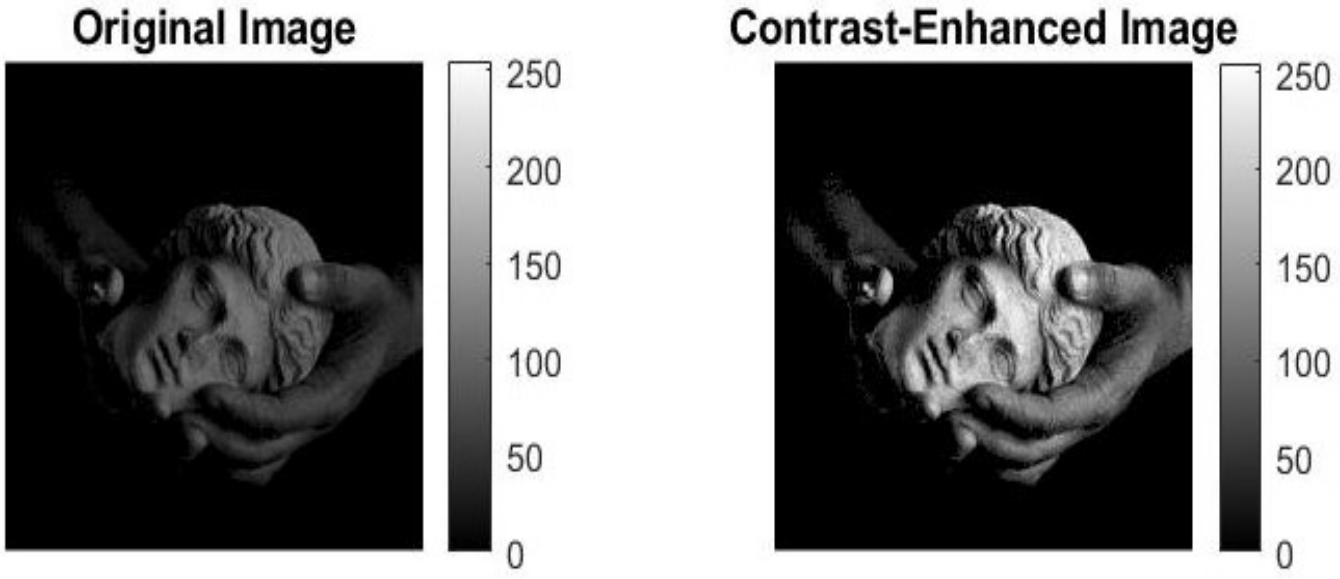
Hence, it can be concluded that Linear Contrast Stretching is not (as) effective in such cases where actual and desired intensity range is equal (nearly same).

Image 6: 'chestXray.png'



Here, Linear Contrast Stretching has improved the contrast a bit, by stretching the range from [6,226] to [0,255]

Image 7: Masked image 'statue.png'



Linear Contrast Stretching has improved the contrast. Note that intensities in range of 150-200 have become closer to 255

Q2. C) Histogram Equalization

Histogram Equalization has enhanced the contrast of images by giving uniformity over the entire range of intensity values.

Image 1: 'barbara.png'

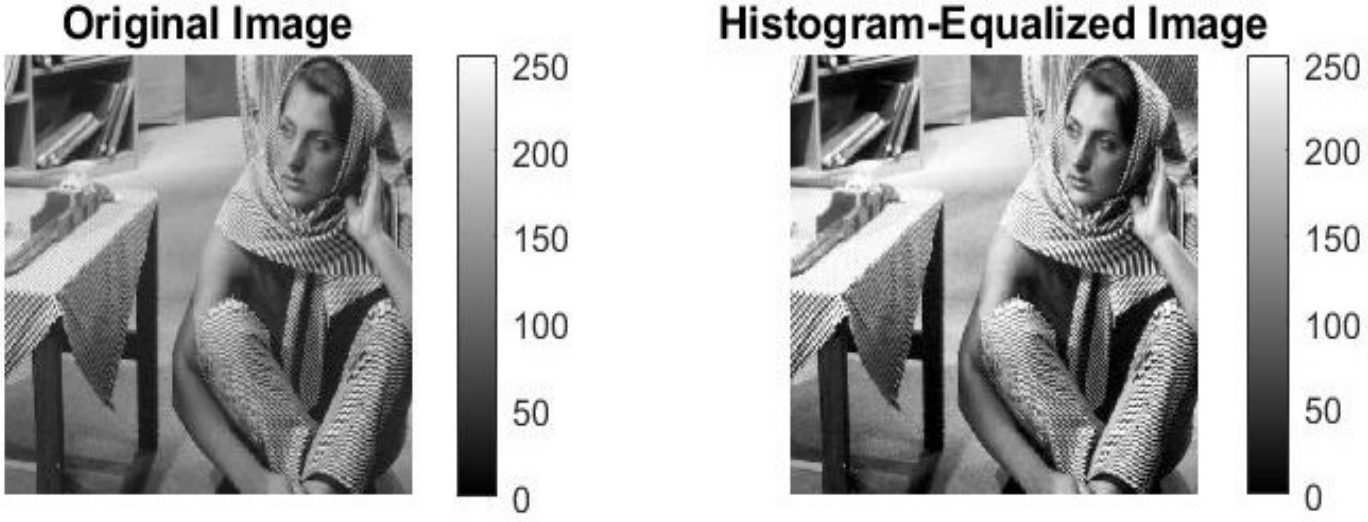


Image 2: 'TEM.png'

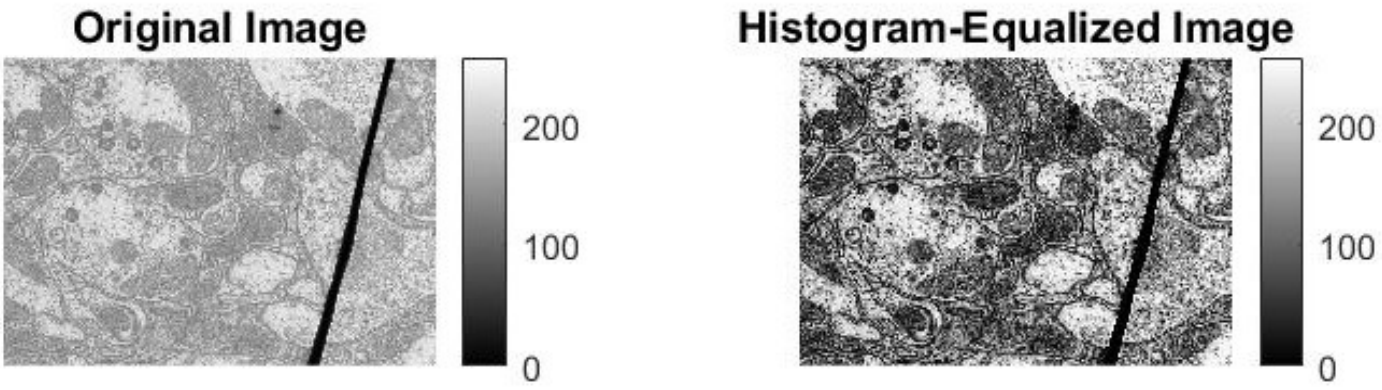


Image 3: 'canyon.png'

Image 5: 'church.png'

Histogram Equalization is definitely preferred over Linear contrast stretching here because Histogram Equalization has uniformly distributed the intensities which were not correctly distributed earlier. This has made the hidden details visible which were not visible in the earlier image because of low intensities over all the channels.

Image 6: 'chestXray.png'

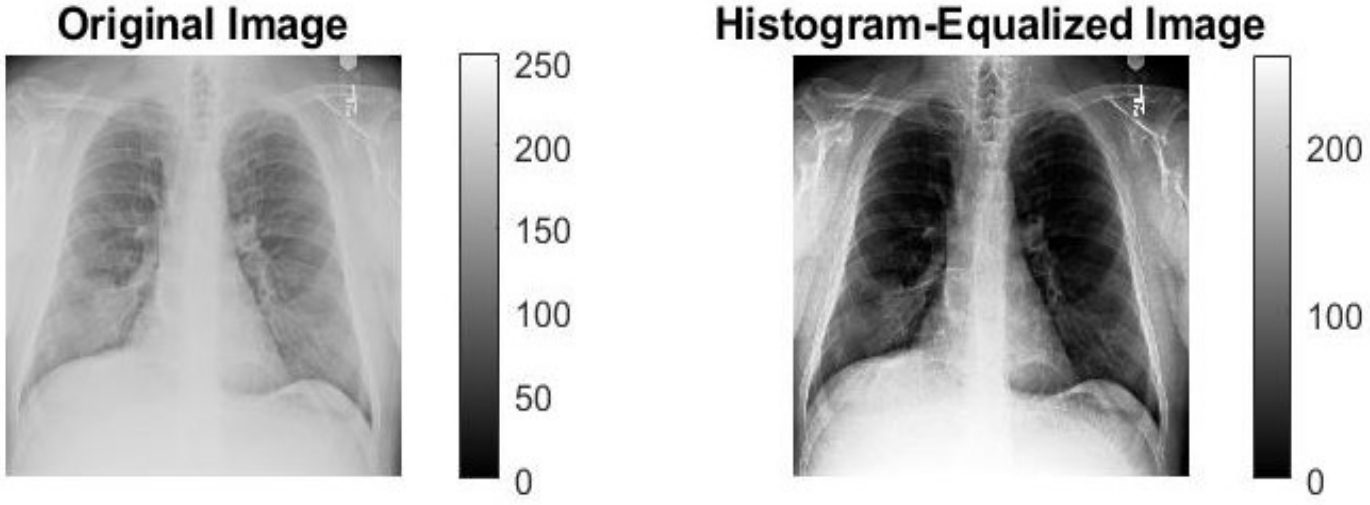
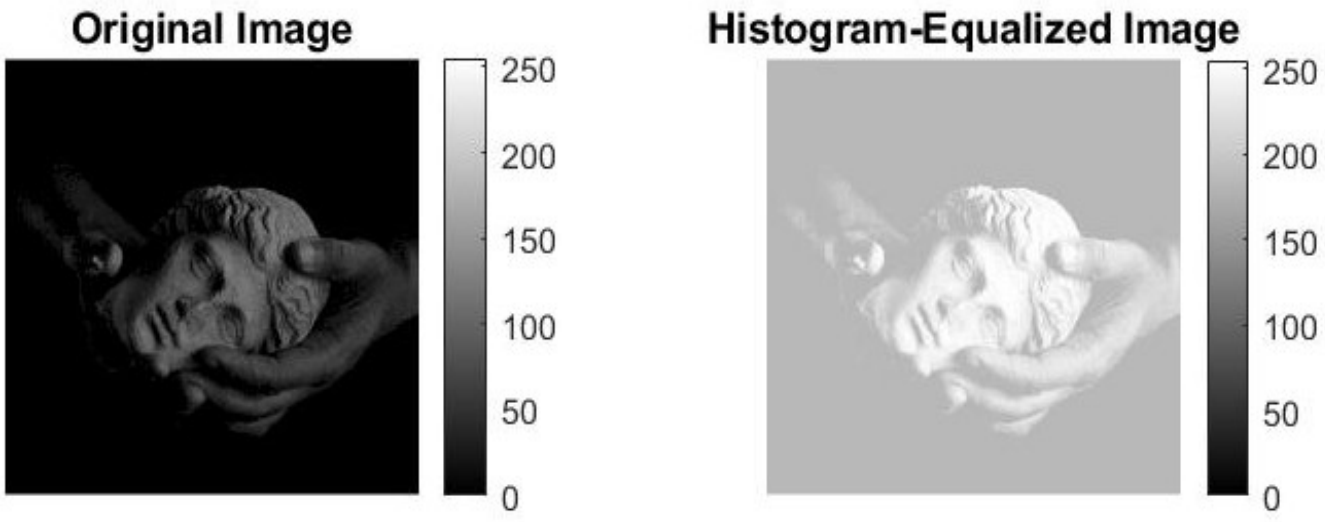


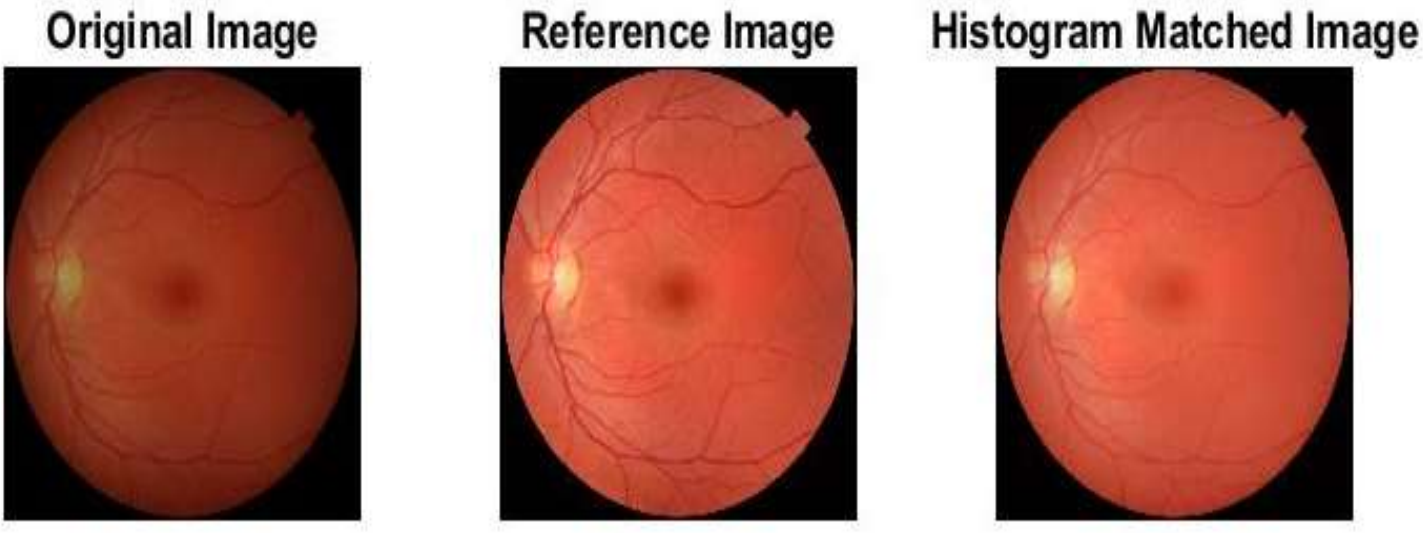
Image 7: Masked image'statue.png'



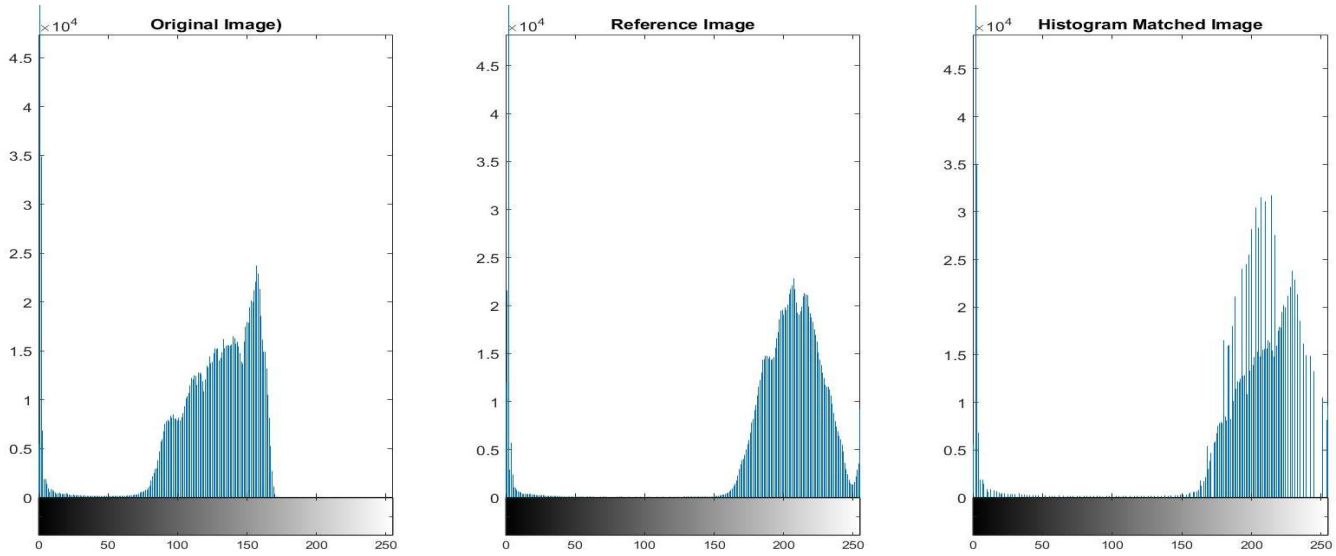
Although, Histogram Equalization works good in most cases, it tends to dissolve the black and white intensities together, to achieve even distribution.

Q2. D) Histogram Matching

Image4: 'retina.png'

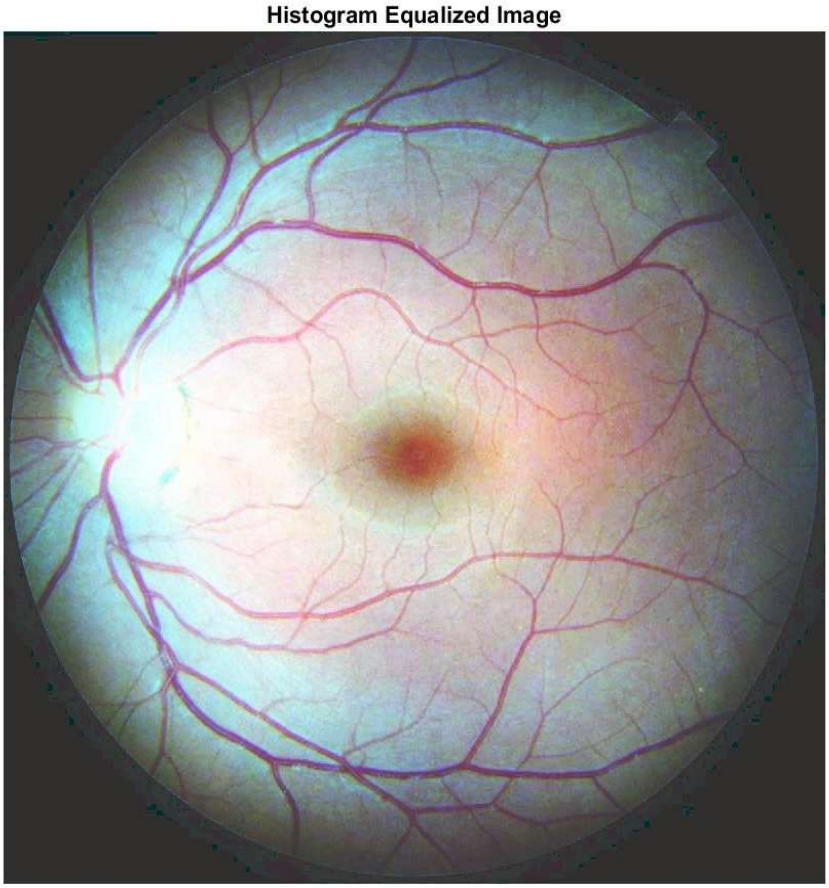


Histogram matching- Although the contrast has enhanced when histogram of original image has been matched to the reference image, the new image is not exact replica of the reference image (veins are more clearly visible in reference image). Hence, it can be concluded that Histogram matching method doesn't match the histograms ***exactly*** *and there can be differences*

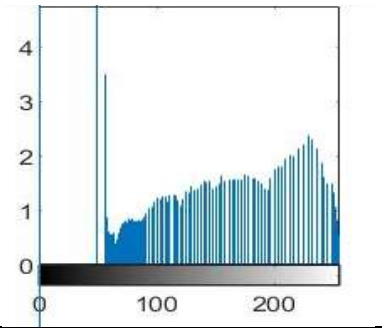
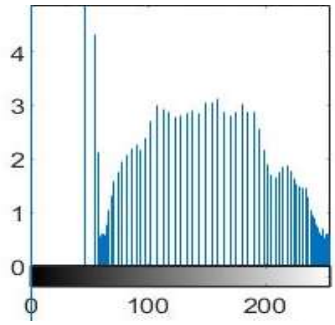
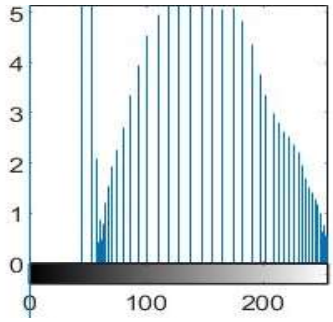
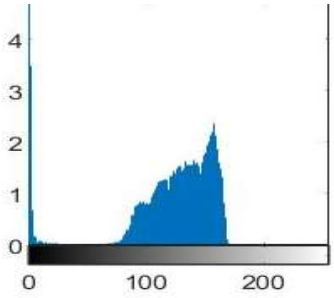
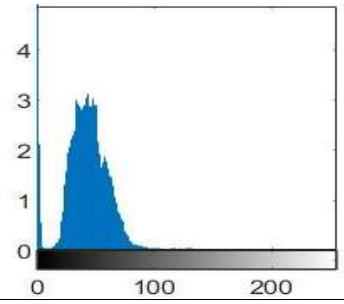
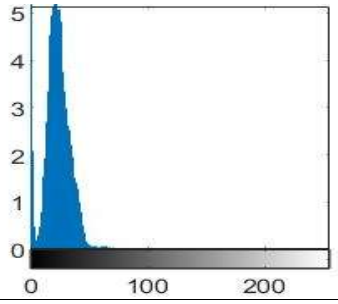


Histograms for channel 1(red)

Intensities have been concentrated towards the right because of histogram matching, but the shape is not exactly the same as that of reference image.



In the original image, red intensities were more spread, and blue and green intensities were sparsely distributed. But Histogram Equalization achieved uniformity in distribution of all the intensities, which resulted in the above Histogram Equalized image.

Histogram Equalized Image(Red)	***Histogram Equalized Image(Green)***	***Histogram Equalized Image(Blue)***
		
Original Image(Red)	***Original Image(Green)***	***Original Image(Blue)***
		

Histograms of Original and histogram Equalized image across all channels

(e) Contrast-Limited Adaptive Histogram Equalization (CLAHE)

Table of Contents

Reading images and colormap

```
myNumOfColors = 255;
myColorScale = [ [0:1/(myNumOfColors-1):1]',[0:1/
(myNumOfColors-1):1]' , [0:1/(myNumOfColors-1):1]' ];
im1 = imread('../data/barbara.png');
im2 = imread('../data/TEM.png');
im3 = imread('../data/canyon.png');
im4 = imread('../data/retina.png');
im5 = imread('../data/church.png');
im6 = imread('../data/chestXray.png');
im7 = imread('../data/statue.png');
```

```
im_ref= imread('../data/retinaRef.png');
```

extracting results using fine tuned parameters

Window size = 50 and histogram threshold = 0.1

```
tic;
clahed_img1a = myCLAHE(im1,50,0.1);
clahed_img2a = myCLAHE(im2,50,0.1);
clahed_img3a = myCLAHE(im3,50,0.1);
clahed_img6a = myCLAHE(im6,50,0.1);
toc;
```

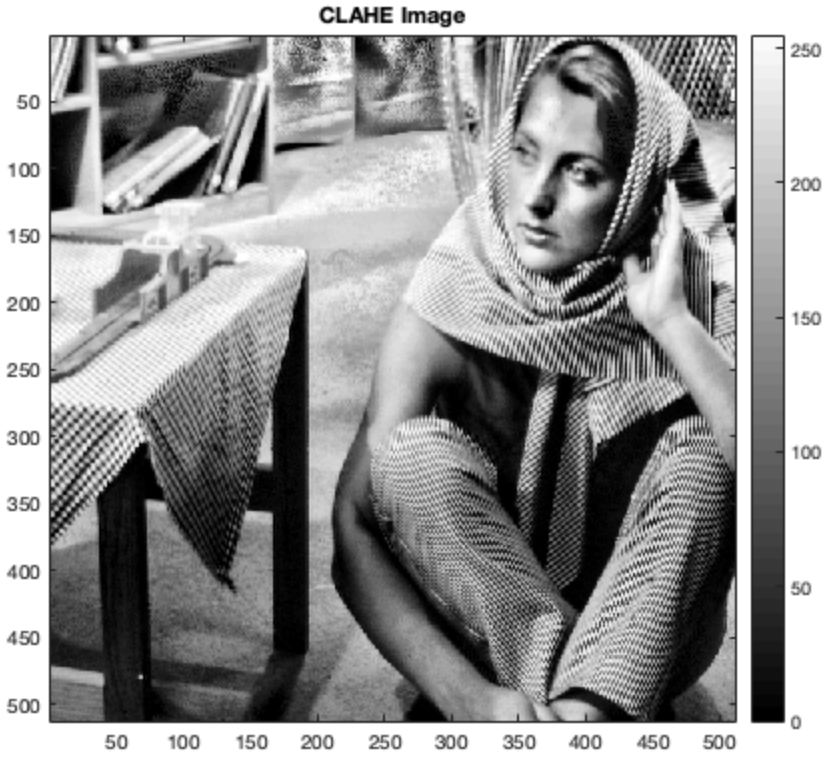
Elapsed time is 168.372220 seconds.

The original image1 (barbara.png)

```
imagesc(single(im1),[0,255]), title('Original
 Image'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

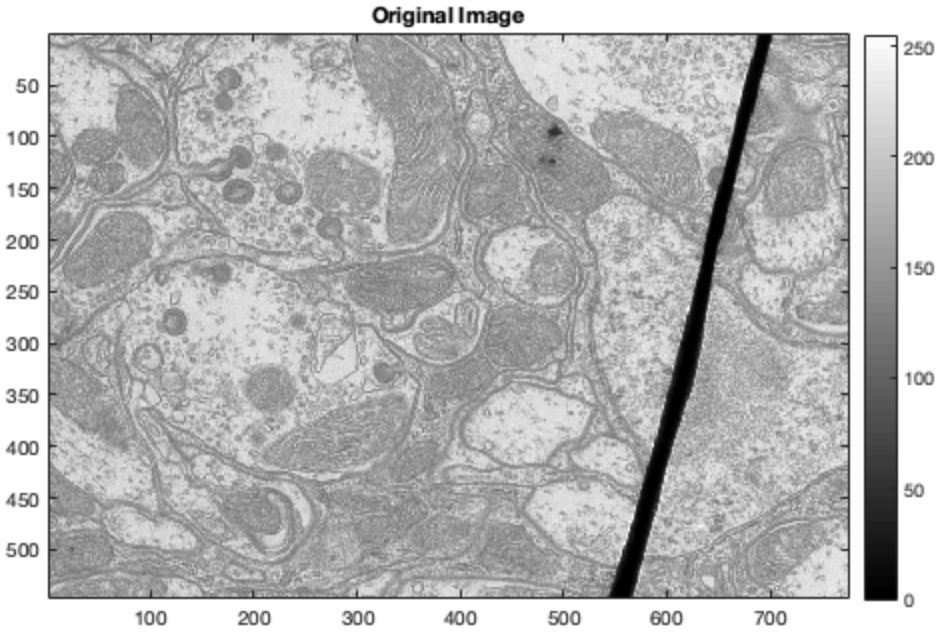
The Clahed image1 (barbara.png)

```
imagesc(single(clahed_img1a),[0,255]), title('CLAHE Image'),
 colormap(myColorScale),daspect ([1 1 1]); axis tight; colorbar;
```



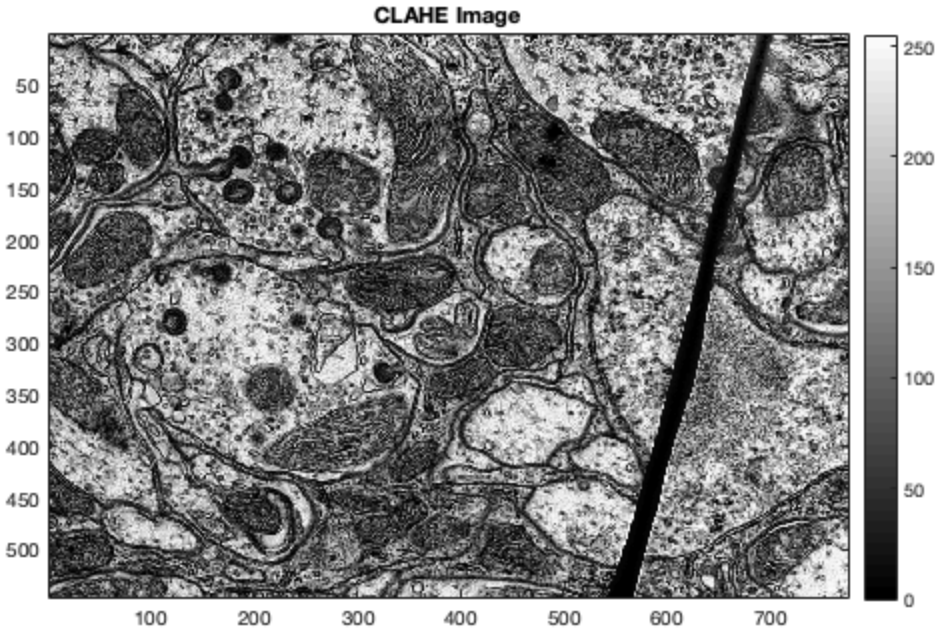
The original image2 (TEM.png)

```
imagesc(single(im2),[0,255]), title('Original
 Image'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```



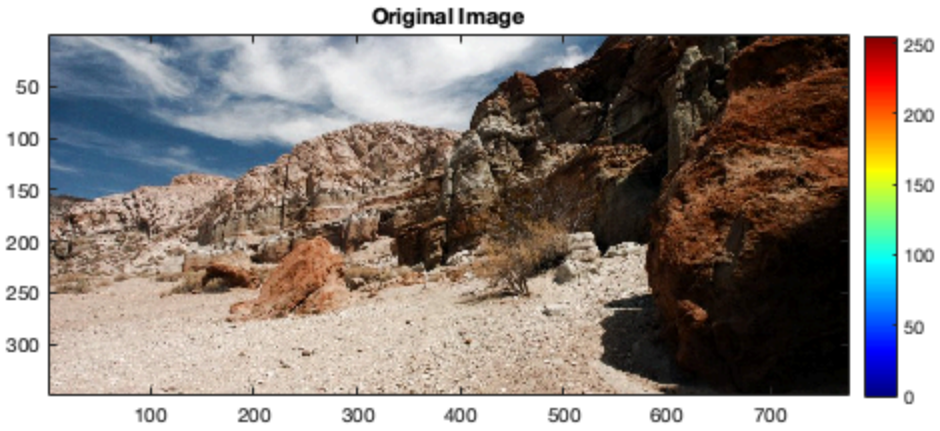
The Clahed image2 (TEM.png)

```
imagesc(single(clahed_img2a),[0,255]), title('CLAHE Image'),
 colormap(myColorScale),daspect ([1 1 1]); axis tight; colorbar;
```



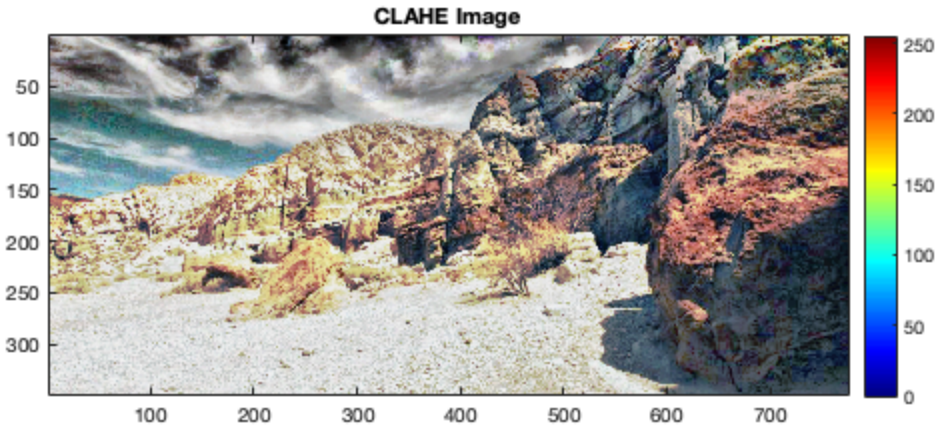
The original image3 (canyon.png)

```
imagesc(mat2gray(im3),[0,255]), title('Original
 Image'),colormap(myColorScale),colormap jet,daspect ([1 1 1]);
 axis tight; colorbar;
```



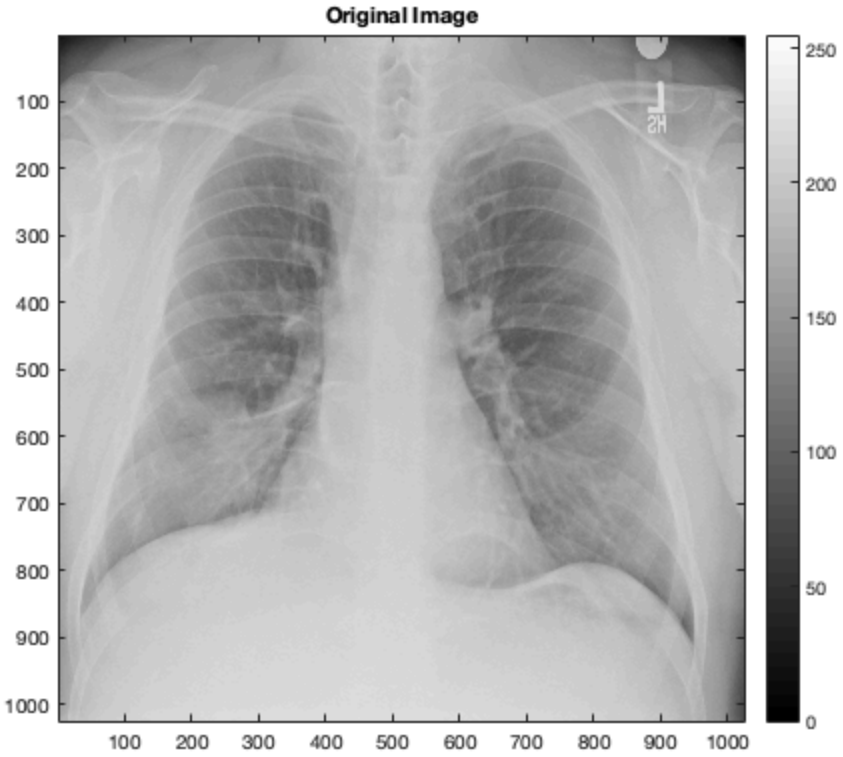
The Clahed image3 (canyon.png)

```
imagesc(mat2gray(clahed_img3a),[0,255]), title('CLAHE Image'),
 colormap(myColorScale),colormap jet,daspect ([1 1 1]); axis tight;
 colorbar;
```



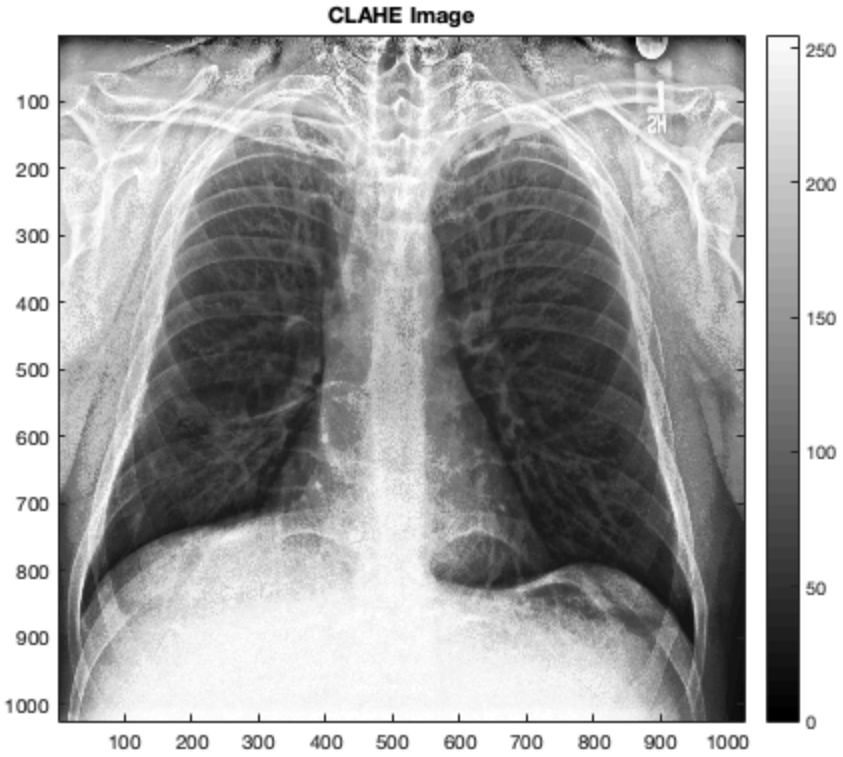
The original image6 (chestXray.png)

```
imagesc(single(im6),[0,255]), title('Original
 Image'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```



The Clahed image6 (chestXray.png)

```
imagesc(single(clahed_img6a),[0,255]), title('CLAHE Image'),
 colormap(myColorScale),daspect ([1 1 1]); axis tight; colorbar;
```



Extracting results after varying Window Size parameter

Window Size = {10,100} histogram threshold = 0.1

```
tic;
progress = waitbar(0,"Running Clahe Algorithm ...");
clahed_img1b1 = myCLAHE(im1,10,0.1); waitbar(1/12);
clahed_img1b2 = myCLAHE(im1,100,0.1); waitbar(3/12);
clahed_img2b1 = myCLAHE(im2,10,0.1); waitbar(4/12);
clahed_img2b2 = myCLAHE(im2,100,0.1); waitbar(6/12);
clahed_img3b1 = myCLAHE(im3,10,0.1); waitbar(7/12);
clahed_img3b2 = myCLAHE(im3,100,0.1); waitbar(9/12);
clahed_img6b1 = myCLAHE(im6,10,0.1); waitbar(10/12);
clahed_img6b2 = myCLAHE(im6,100,0.1); waitbar(12/12);
delete(progress);
toc;
```

Elapsed time is 433.569351 seconds.

image1 with window size 100 (barbara.png)

```
imagesc(single(clahed_img1b1),[0,255]), title('10 window
 size'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

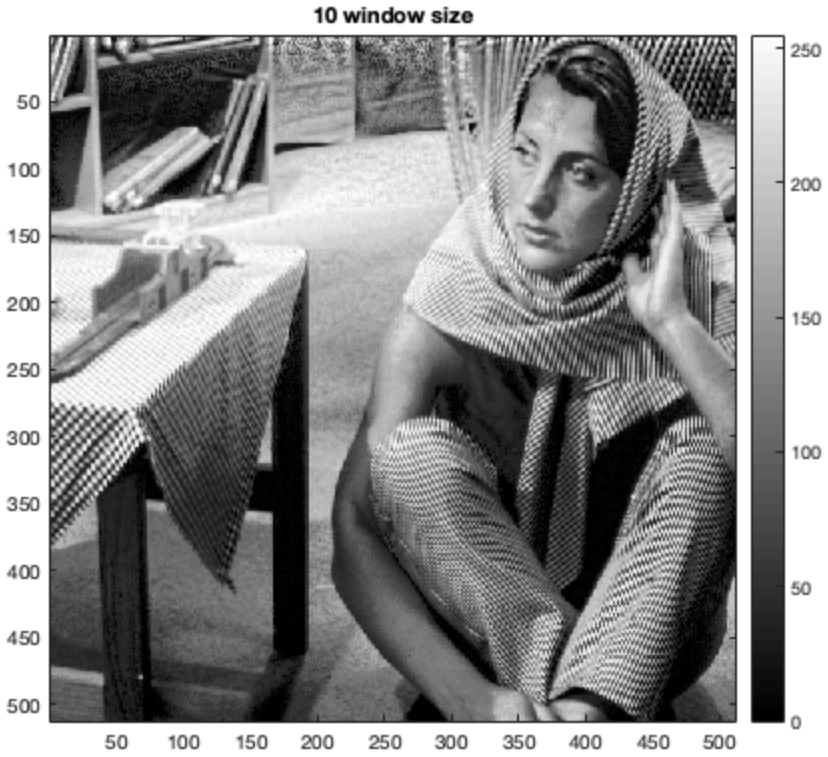


image1 with window size 100 (barbara.png)

```
imagesc(single(clahed_img1b2),[0,255]), title('100 window
 size'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

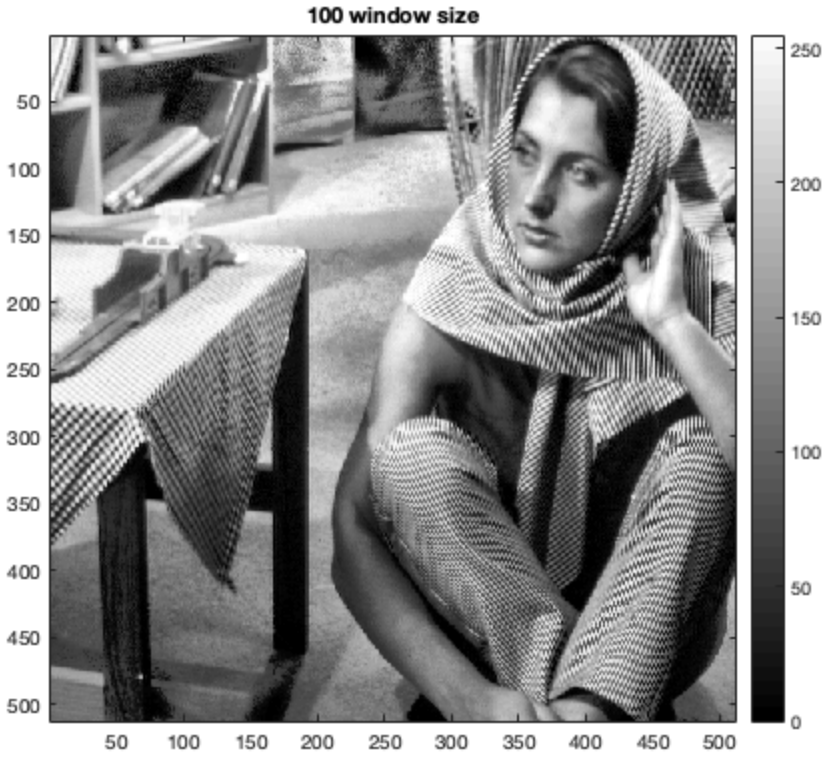


image2 with window size 10 (TEM.png)

```
imagesc(single(clahed_img2b1),[0,255]), title('10 window
 size'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

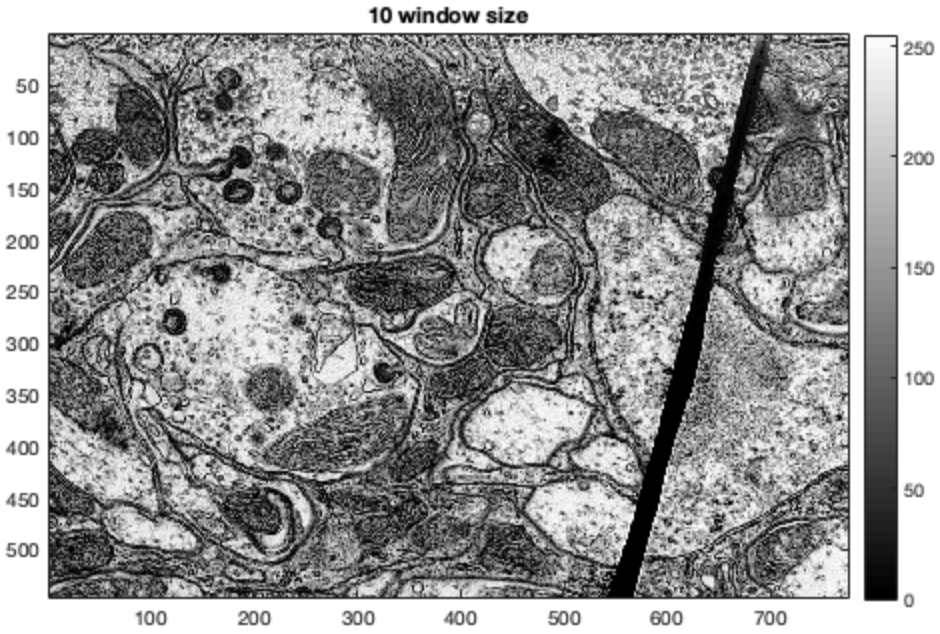


image2 with window size 100 (TEM.png)

```
imagesc(single(clahed_img2b2),[0,255]), title('100 window
 size'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

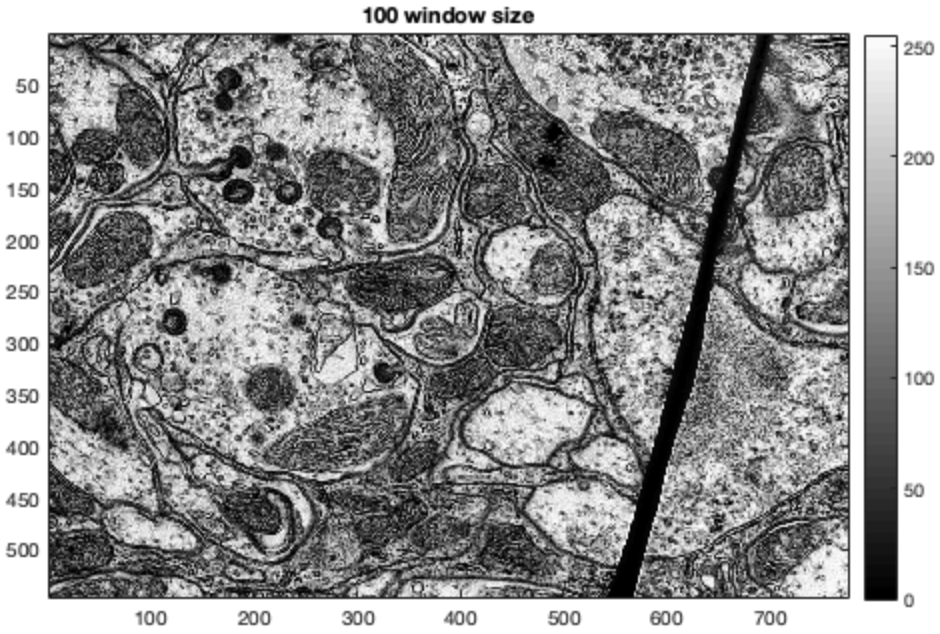


image3 with window size 10 (canyon.png)

```
imagesc(mat2gray(clahed_img3b1),[0,255]), title('10 window
 size'),colormap(myColorScale),colormap jet,daspect ([1 1 1]);
 axis tight; colorbar;
```

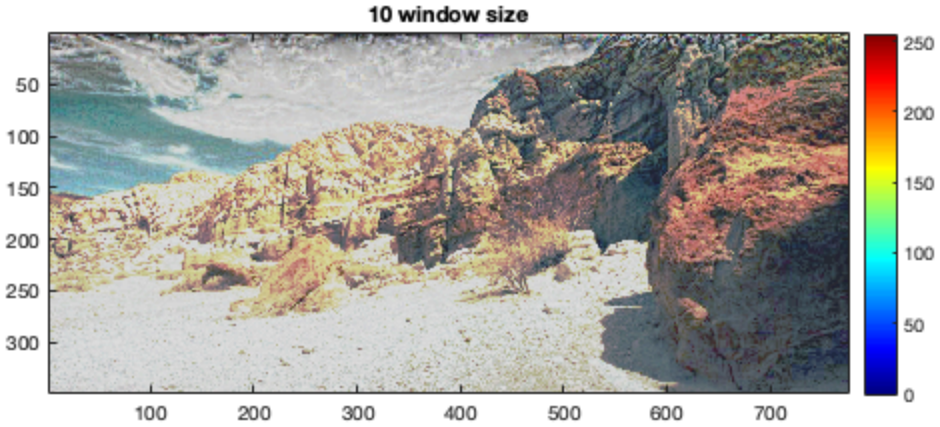


image3 with window size 100 (canyon.png)

```
imagesc(mat2gray(clahed_img3b2),[0,255]), title('100 window
 size'),colormap(myColorScale),colormap jet,daspect ([1 1 1]);
 axis tight; colorbar;
```

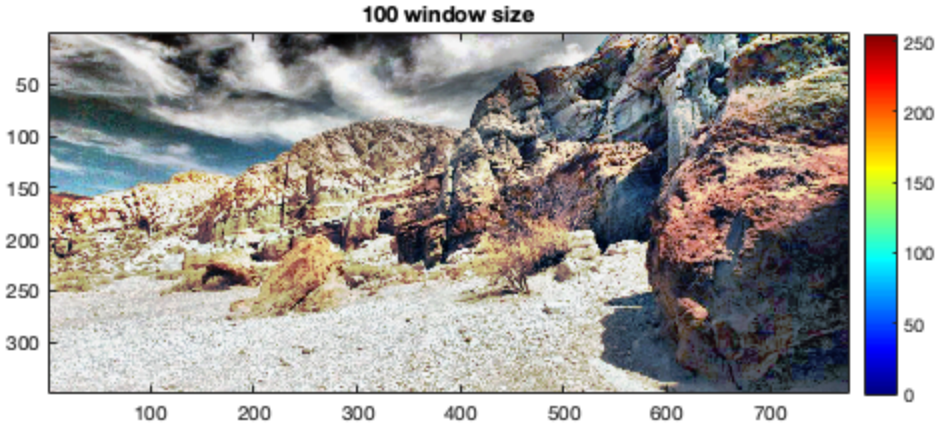


image6 with window size 10 (chestXray.png)

```
imagesc(single(clahed_img6b1),[0,255]), title('10 window
 size'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

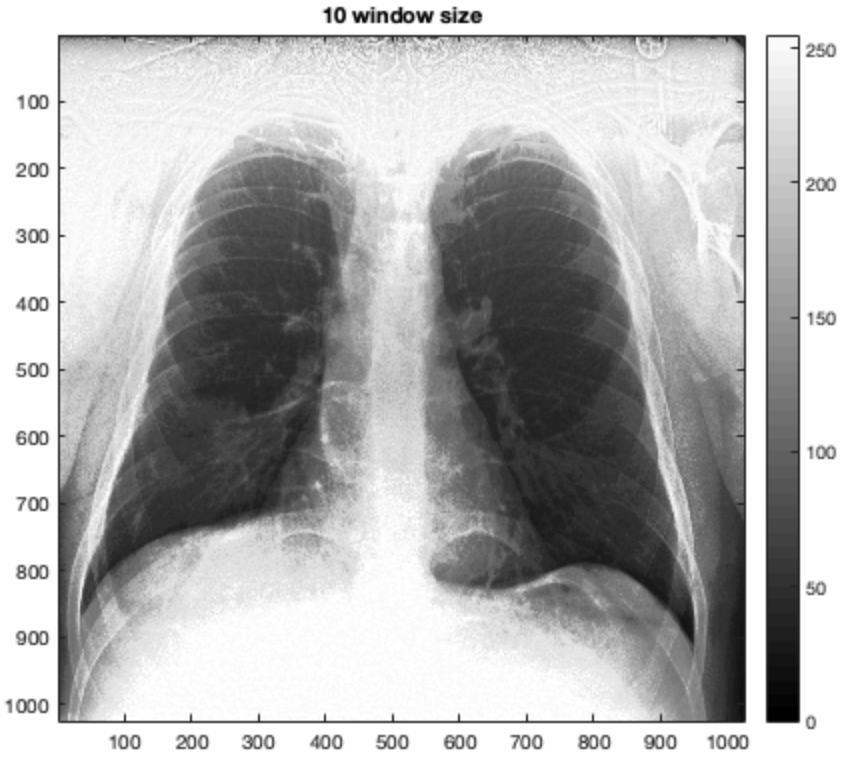
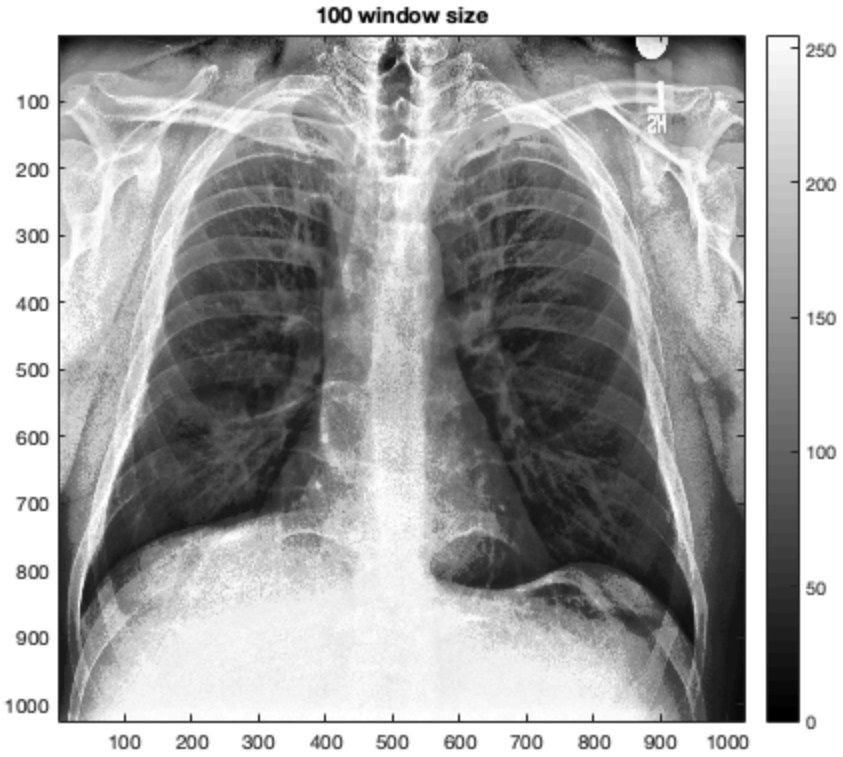


image6 with window size 10 (chestXray.png)

```
imagesc(single(clahed_img6b2),[0,255]), title('100 window
 size'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```



Observation for variation of window size parameter

We can observe that as the window size is reduced we are more affected by the changes in the neighbourhood of a given pixel, the noise generated for a small window size here is not limited by the used threshold of 0.1 which lead to overall blurry and less detailed images.

Extracting results for histogram threshold variation

Window size = 50, histogram threshold = 0.05

```
tic;
clahed_img1c = myCLAHE(im1,50,0.05);
clahed_img2c = myCLAHE(im2,50,0.05);
clahed_img3c = myCLAHE(im3,50,0.05);
clahed_img6c = myCLAHE(im6,50,0.05);
toc;

Elapsed time is 214.776700 seconds.
```

image1 with histogram threshold 0.1 (barbara.png)

```
imagesc(single(clahed_img1a),[0,255]), title('0.1 histogram
 threshold'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

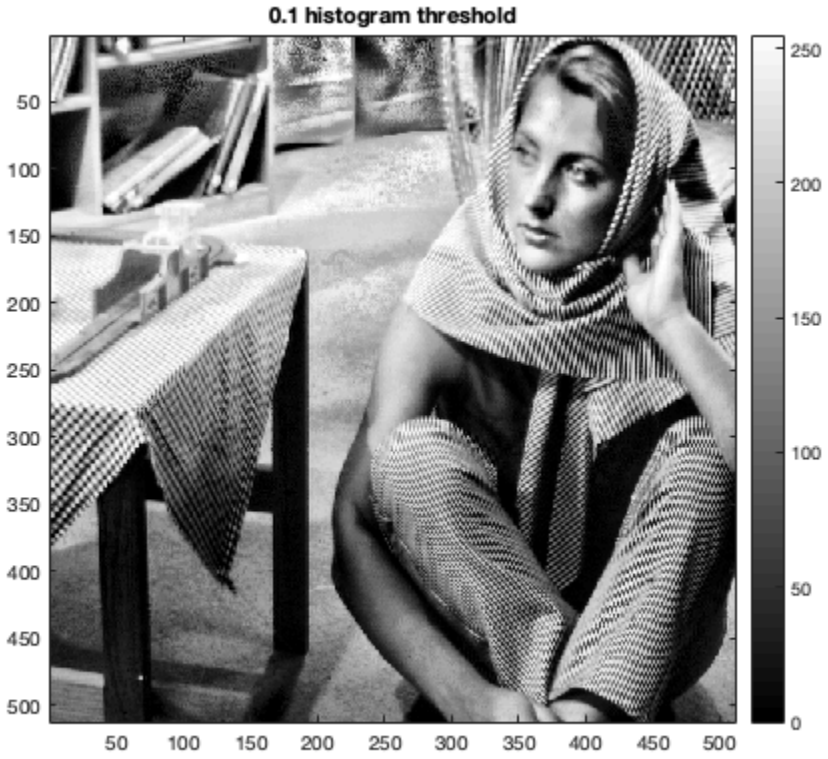


image1 with histogram threshold 0.05 (barbara.png)

```
imagesc(single(clahed_img1c),[0,255]), title('0.05 histogram
 threshold'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

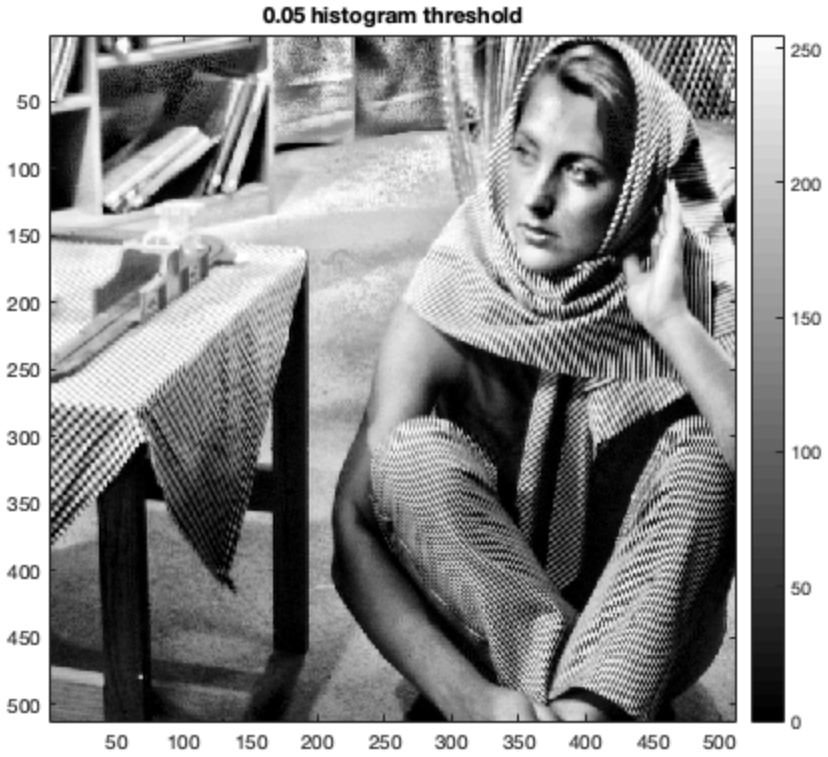


image2 with histogram threshold 0.1 (TEM.png)

```
imagesc(single(clahed_img2a),[0,255]), title('0.1 histogram
 threshold'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

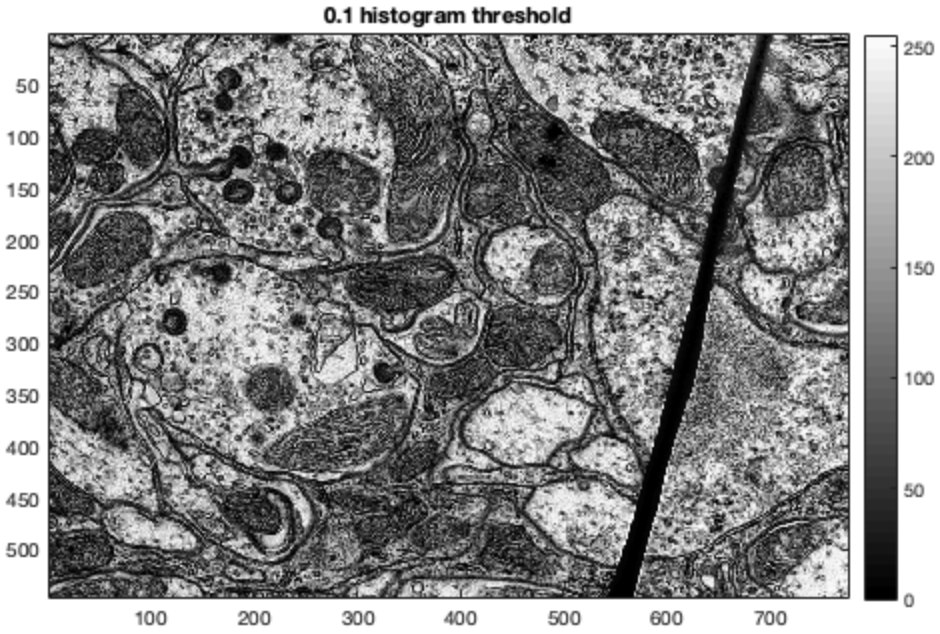


image2 with histogram threshold 0.05 (TEM.png)

```
imagesc(single(clahed_img2c),[0,255]), title('0.05 histogram
 threshold'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

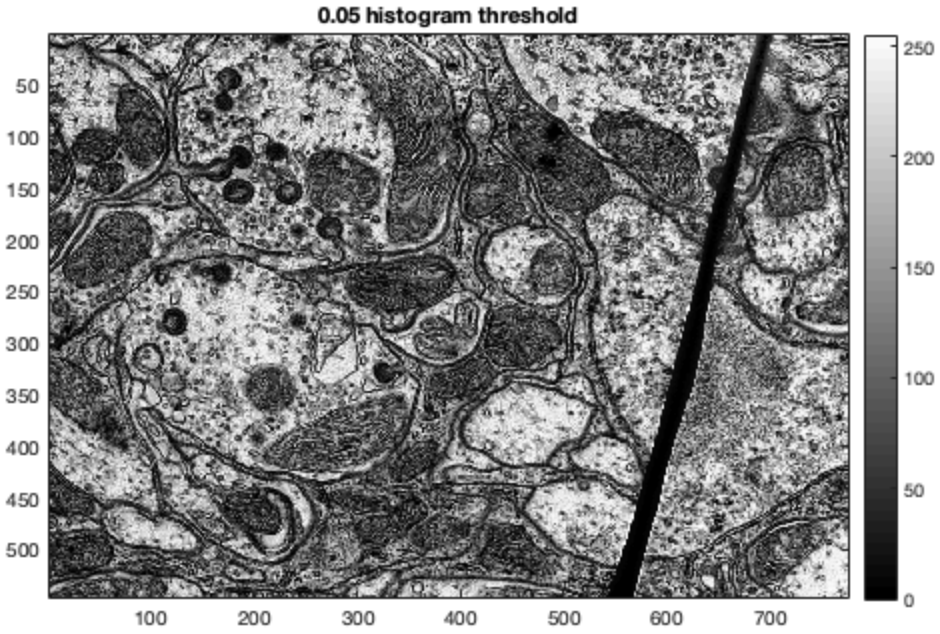


image3 with histogram threshold 0.1 (canyon.png)

```
imagesc(mat2gray(clahed_img3a),[0,255]), title('0.1 histogram
 threshold'),colormap(myColorScale),colormap jet,daspect ([1 1 1]);
 axis tight; colorbar;
```

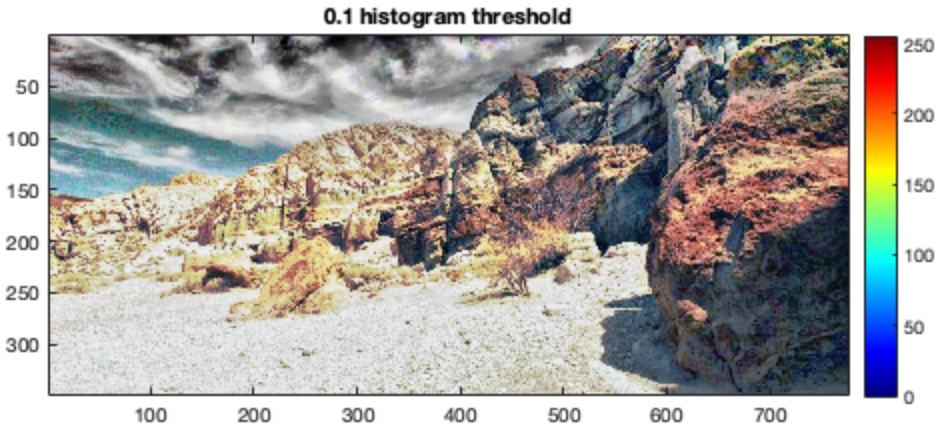


image3 with histogram threshold 0.05 (canyon.png)

```
imagesc(mat2gray(clahed_img3c),[0,255]), title('0.05 histogram
 threshold'),colormap(myColorScale),colormap jet,daspect ([1 1 1]);
 axis tight; colorbar;
```

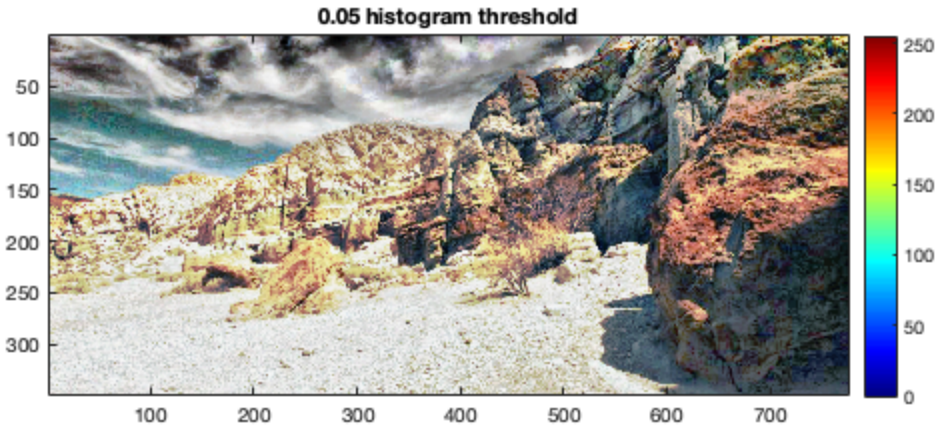


image6 with histogram threshold 0.1 (chestXray.png)

```
imagesc(single(clahed_img6a),[0,255]), title('0.1 histogram
 threshold'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```

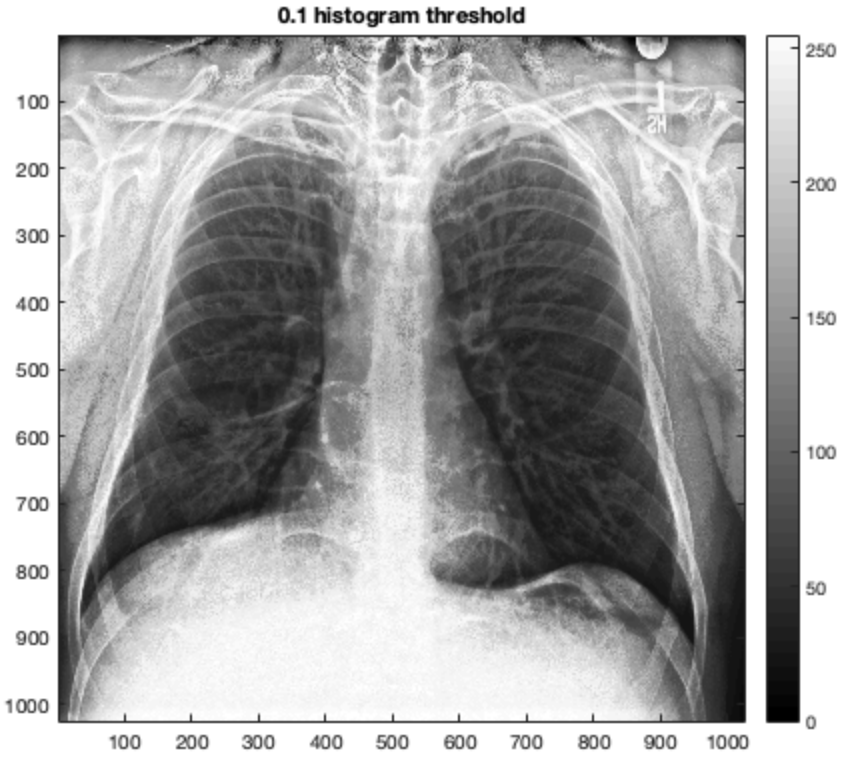
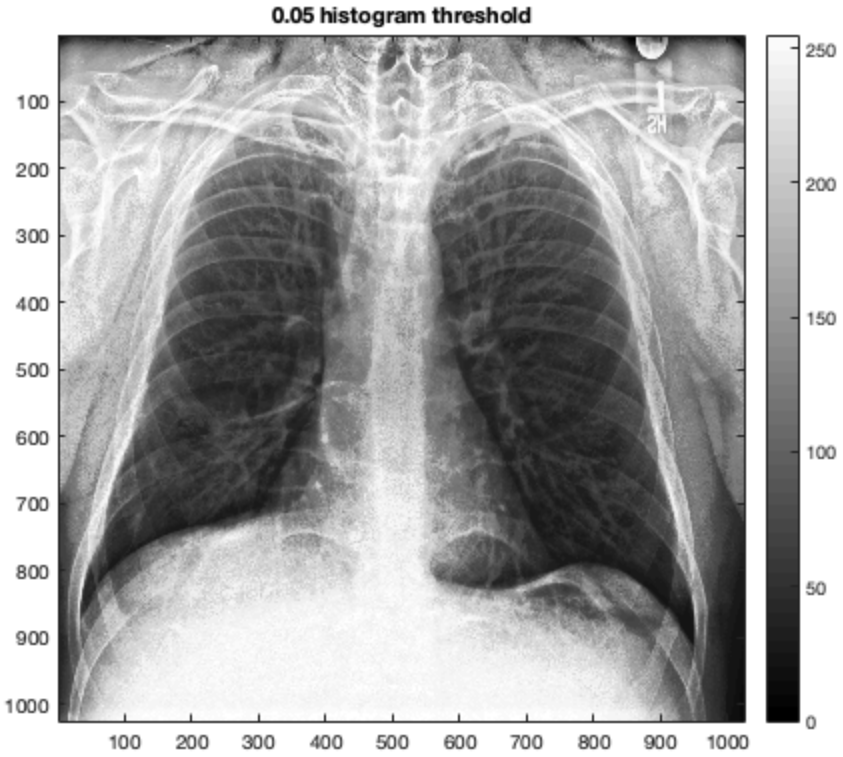


image6 with histogram threshold 0.05 (chestXray.png)

```
imagesc(single(clahed_img6c),[0,255]), title('0.05 histogram
 threshold'),colormap(myColorScale),daspect ([1 1 1]); axis tight;
 colorbar;
```



Observations for variation of histogram threshold

it is harder to see significant changes in this variation but we can still observe in the image1 (barbara.png) the shadows are slightly darker and sharp. since as the histogram threshold gets smaller it leads to more distributions of intensities across all bins of histogram

THE END

Published with MATLAB® R2020a