ASSIGNMENT-03

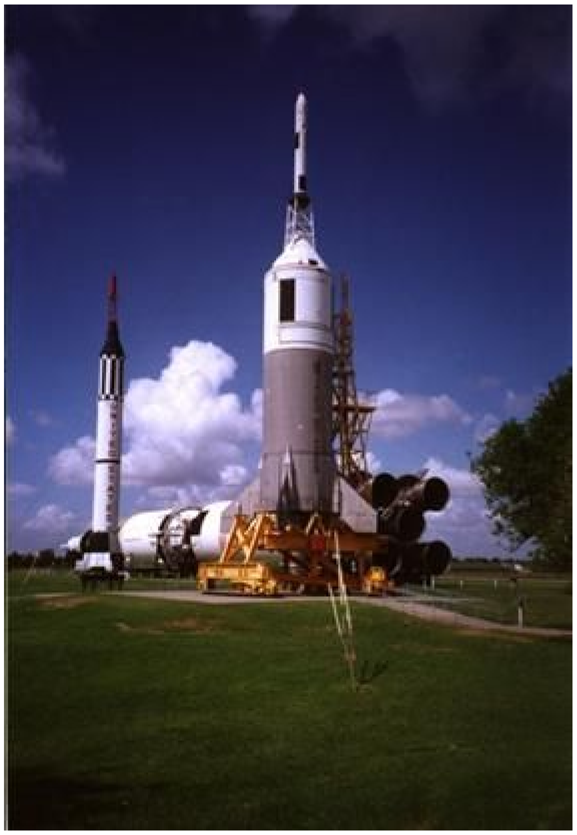
Course Title: Image Processing
Course ID: CSC 420
Submitted By: Abu Bakar Siddik Nayem.
ID: 1510190.
Section: 01.

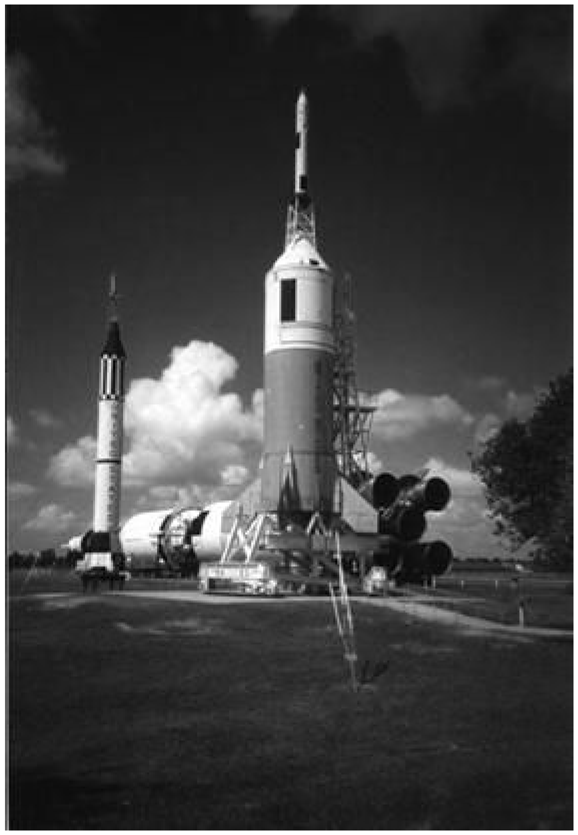
Date of Submission: 13th October, 2018.
Submitted To: Md. Ashraful Amin, PhD.

A. Monochromatic image display for color images for color uses frequency analysis in paintings.

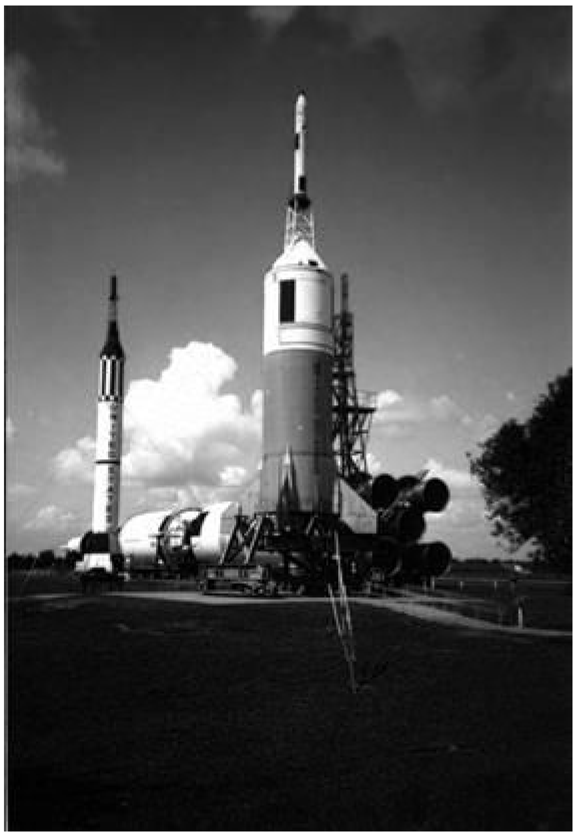
Ans: Input image is a 3 channel rgb image. By separating individual channels (red, green & blue) we can get 3 monochromatic images for the input rgb image.

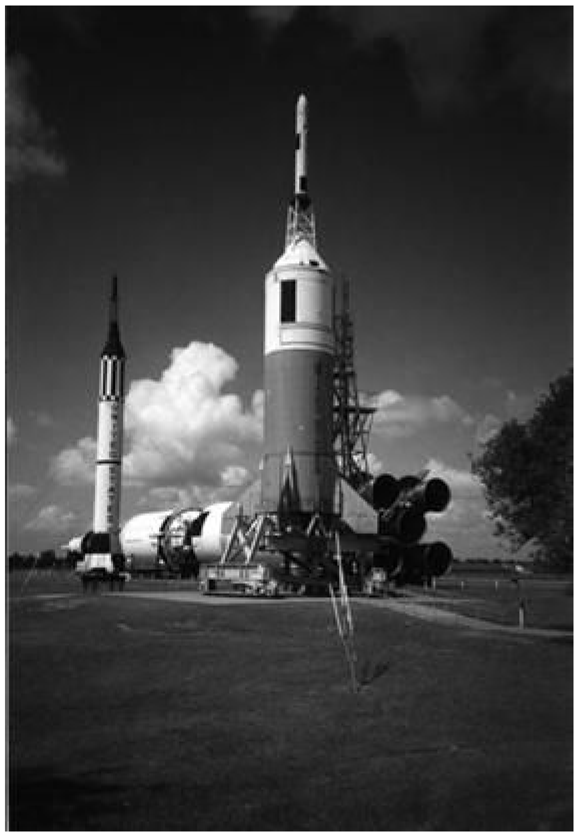
Input : RGB image file. **Output : Red channel.**





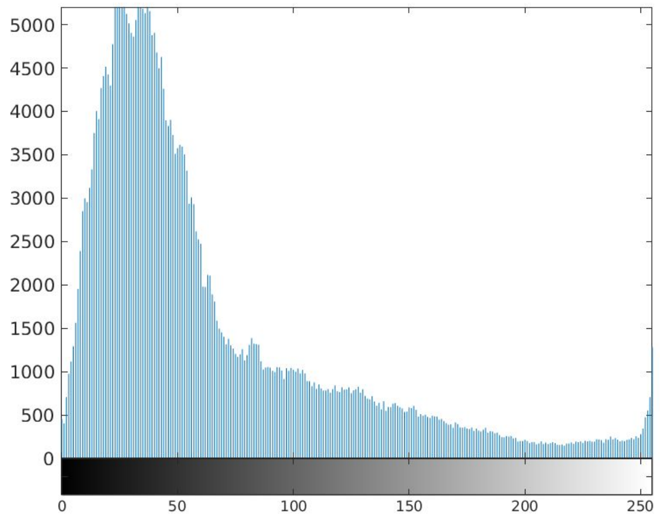
Output: Green channel. **Output: Blue channel.**



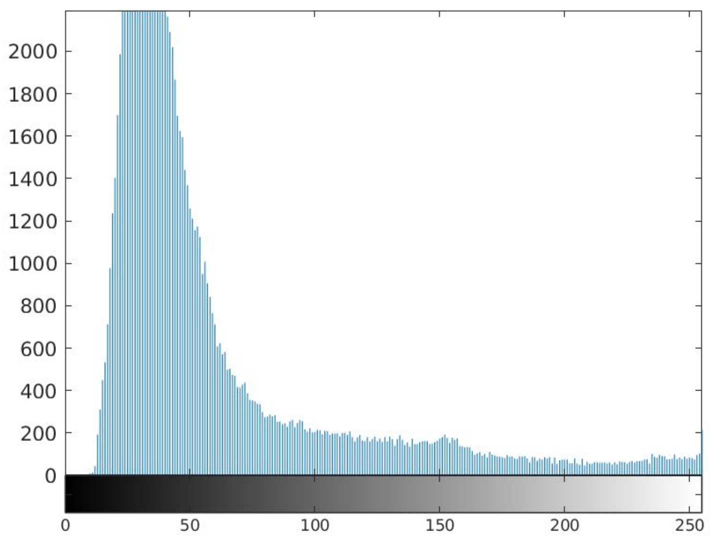


<u>Histograms:</u>

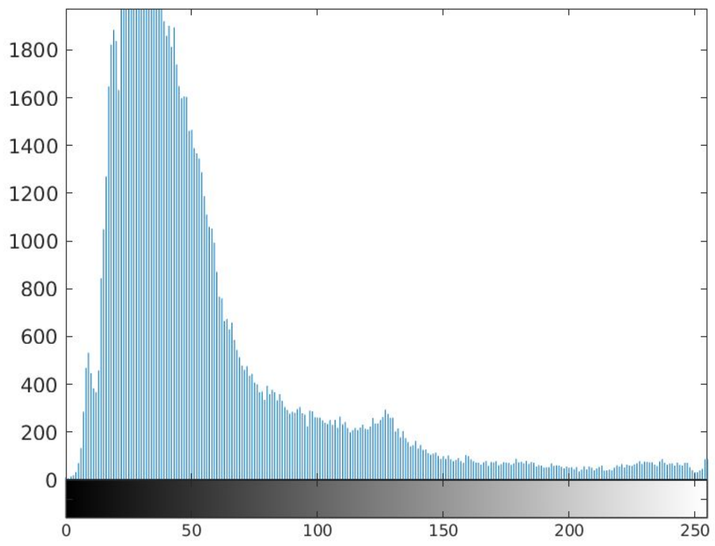
Histogram of the input RGB image:



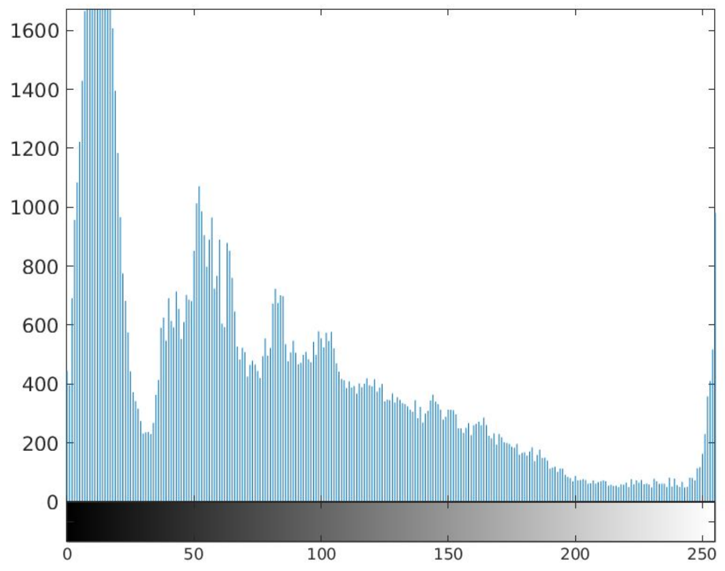
Histogram of Red channel:



Histogram of Green channel:



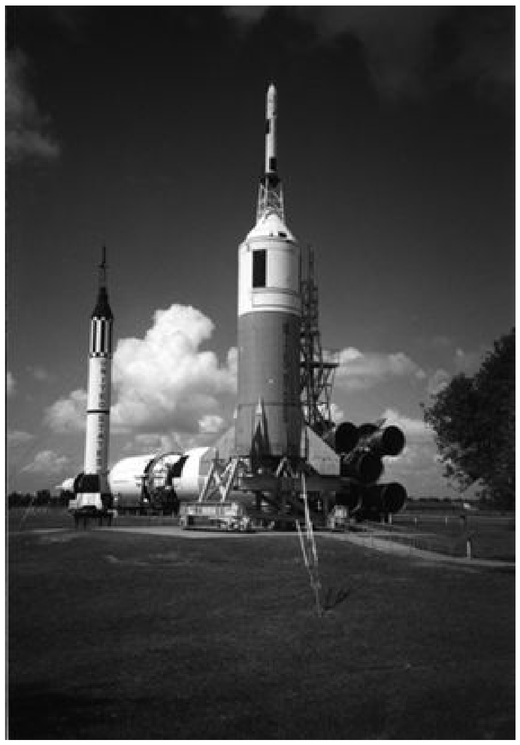
Histogram of Blue channel:



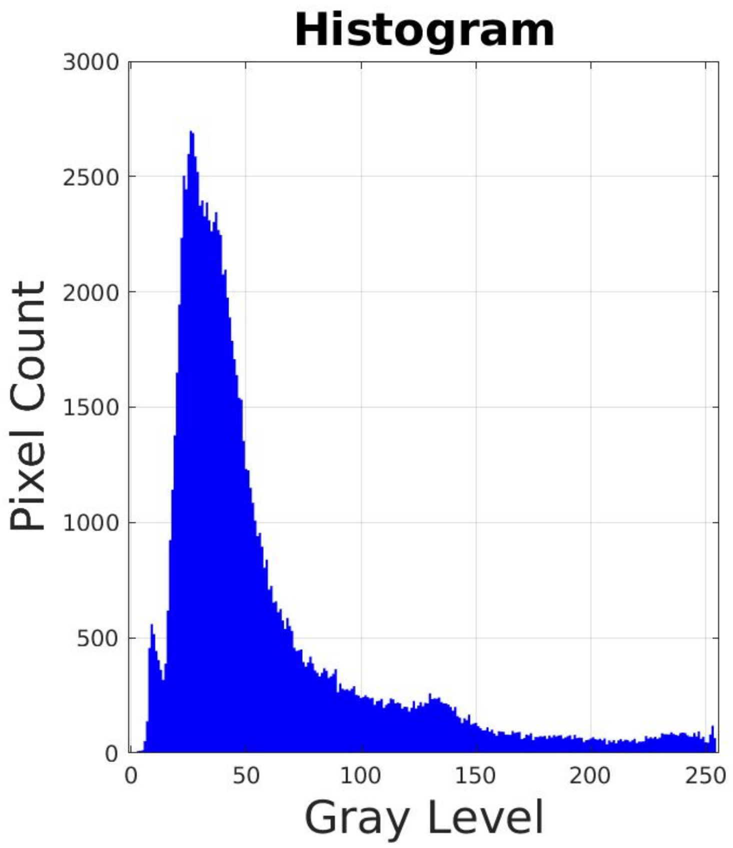
B. Gray image histogram calculation function & display function.

Ans:

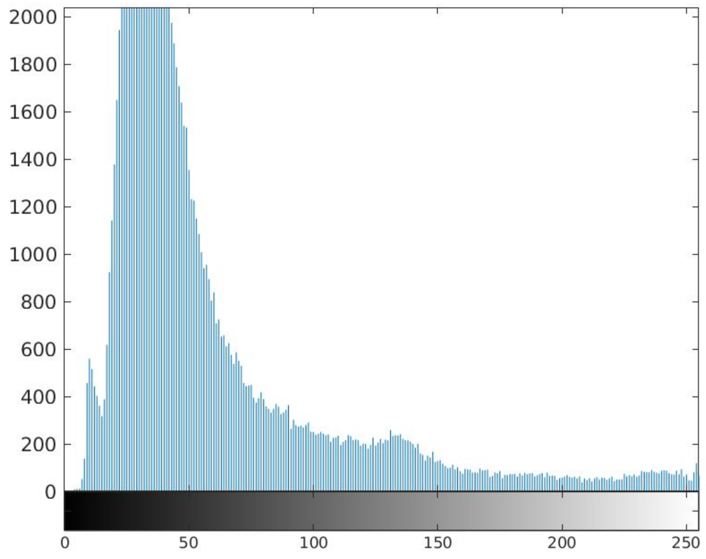
Input Image:



My Histogram:



Matlab imhist():

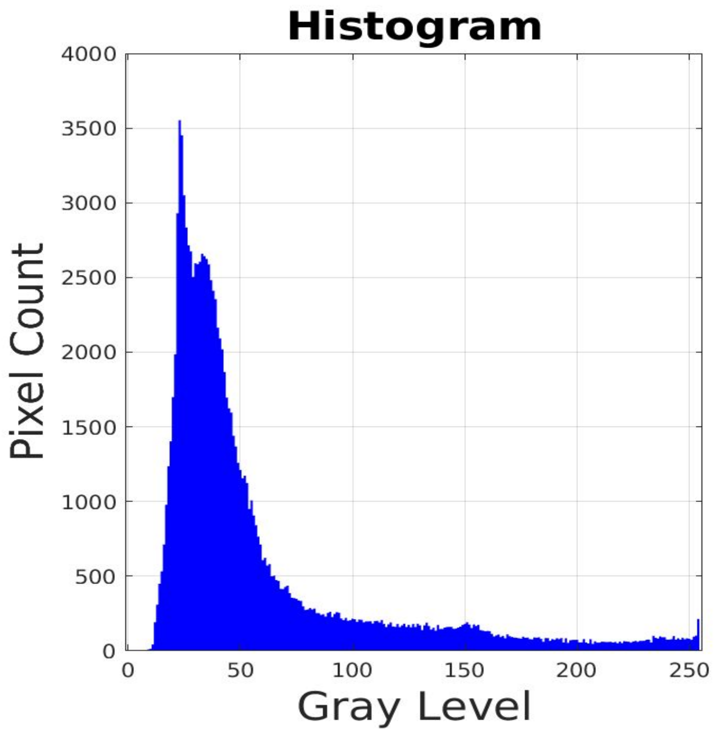


C. Color image histogram calculation function & display function.

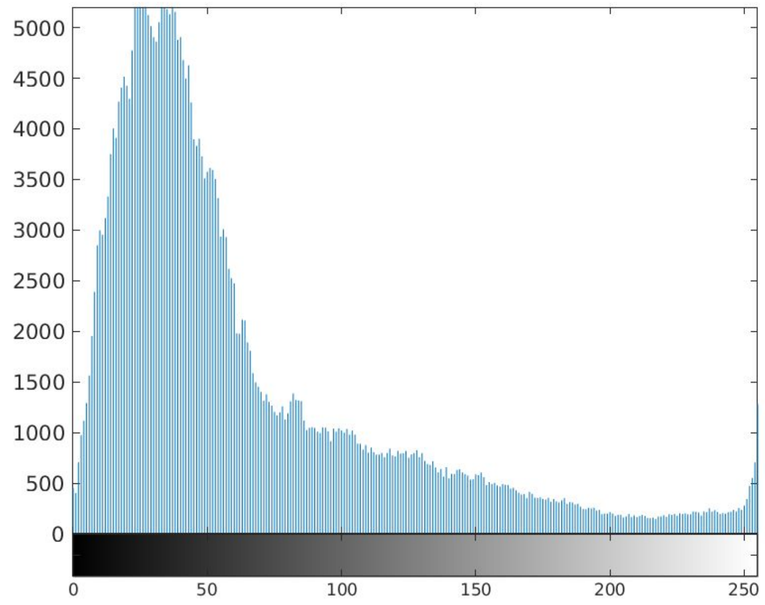
Ans:

Input Image:

My Histogram:

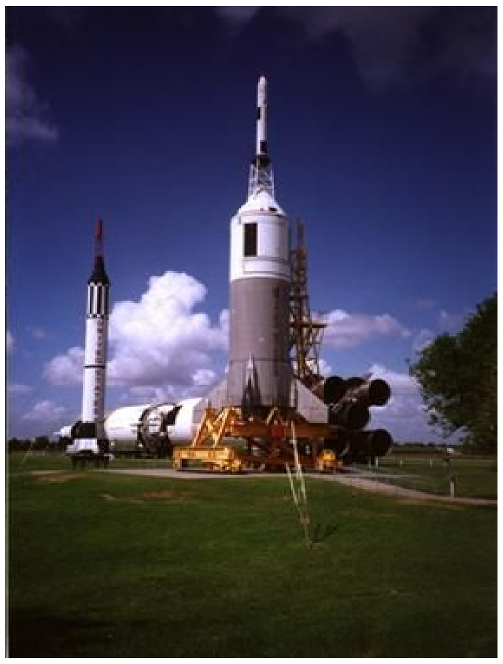


Matlab imhist():

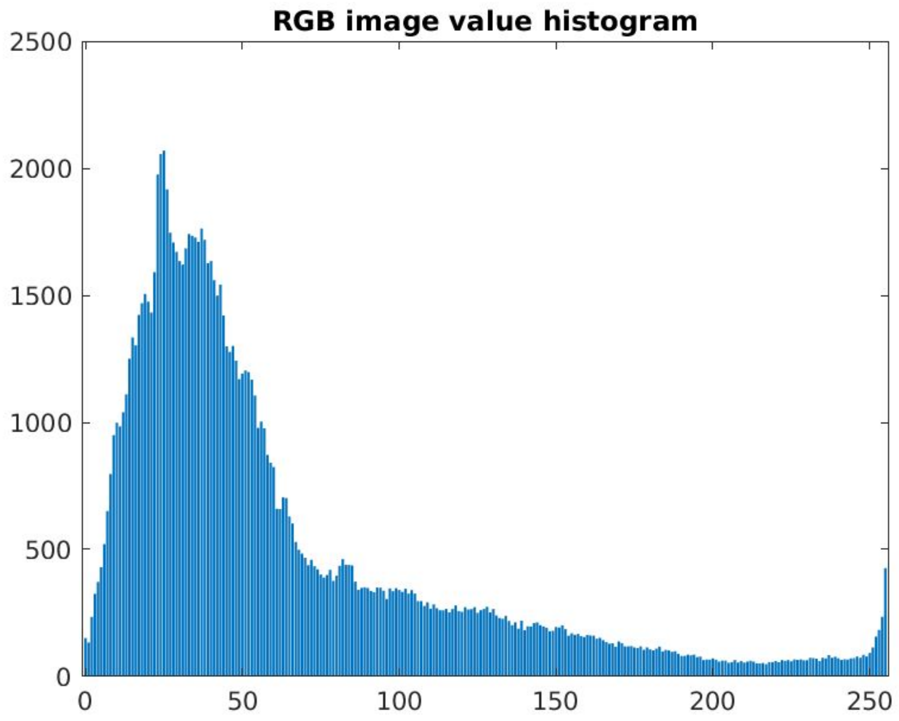


D. Color image value and luminosity histogram calculation function. Display function.
Ans:

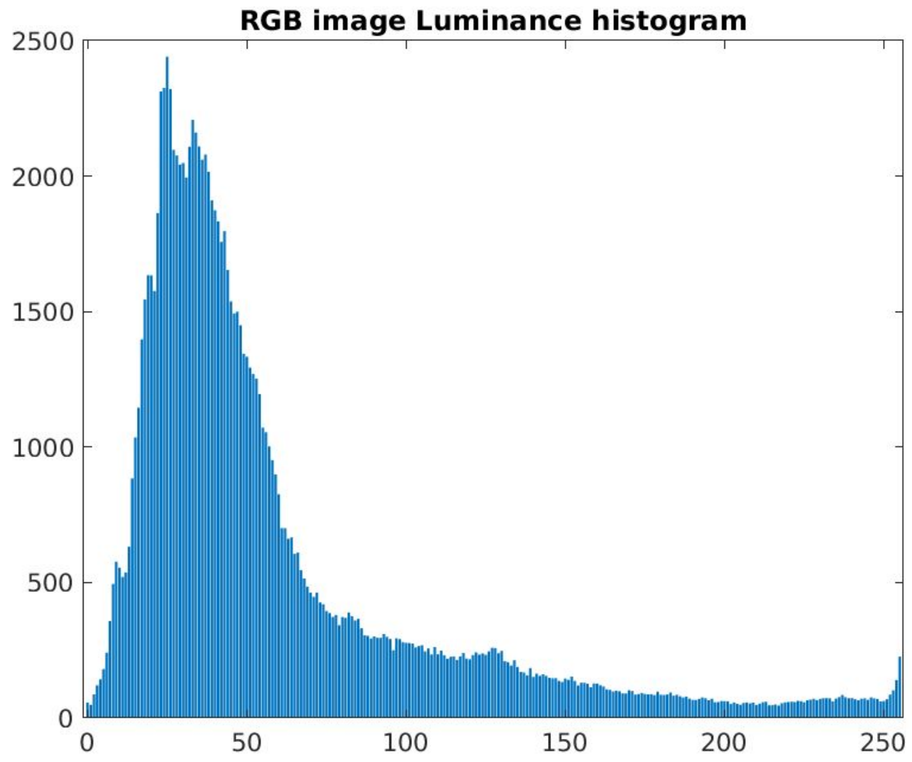
Input Image:



Value histogram:

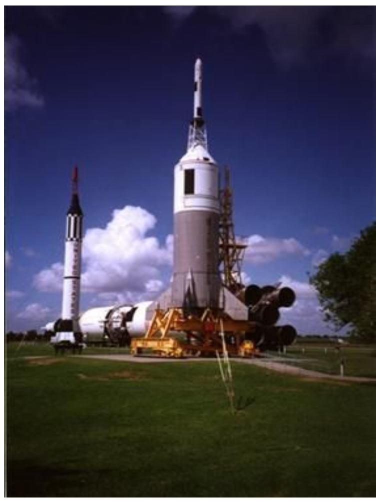


Luminosity histogram:

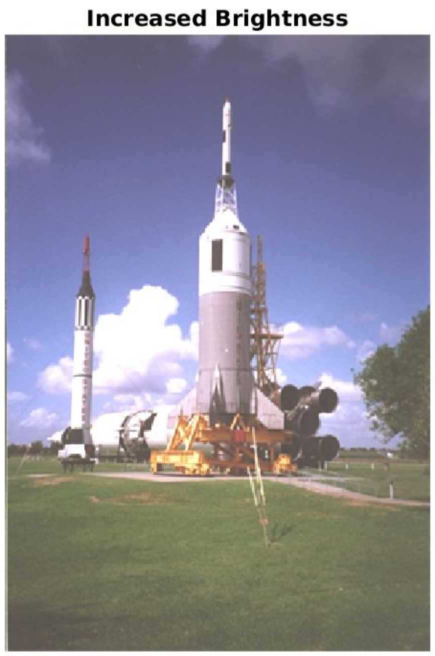


E. Increase brightness function.
Ans:

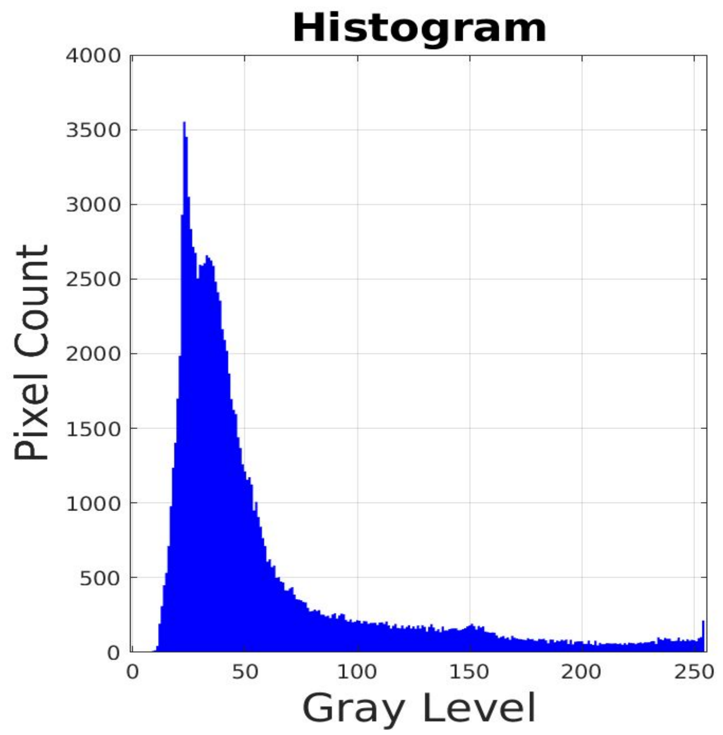
Input Image:



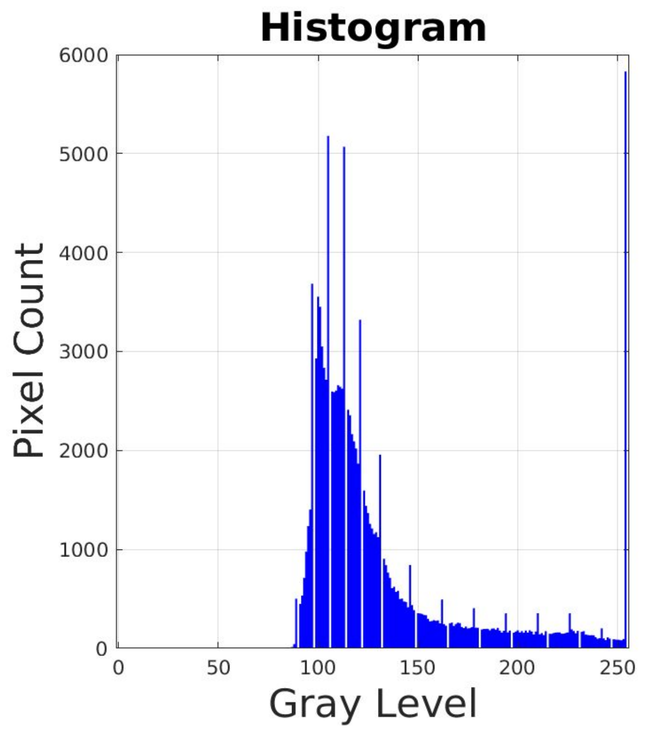
Increased Brightness:



Original image histogram:



Histogram after brightness increased:

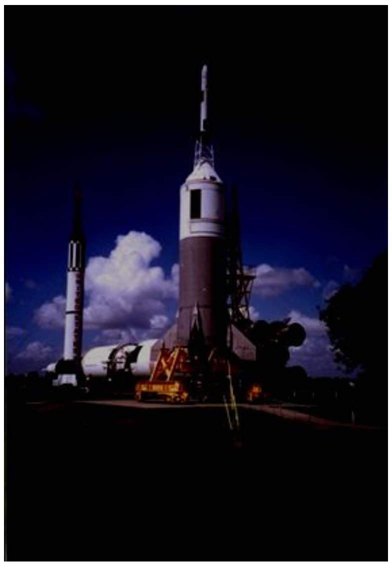


F. Decrease brightness function.

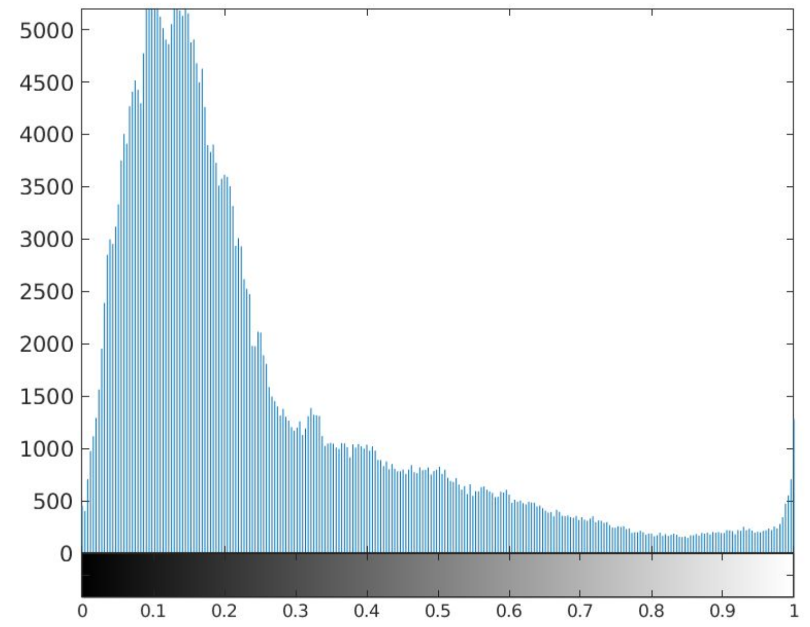
Ans:

Input Image:

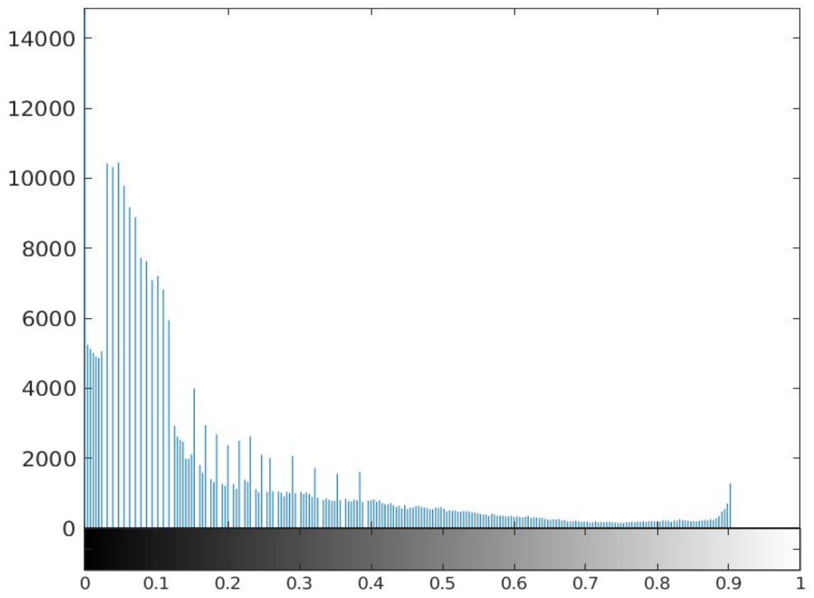
Decreased Brightness:



Original image histogram:



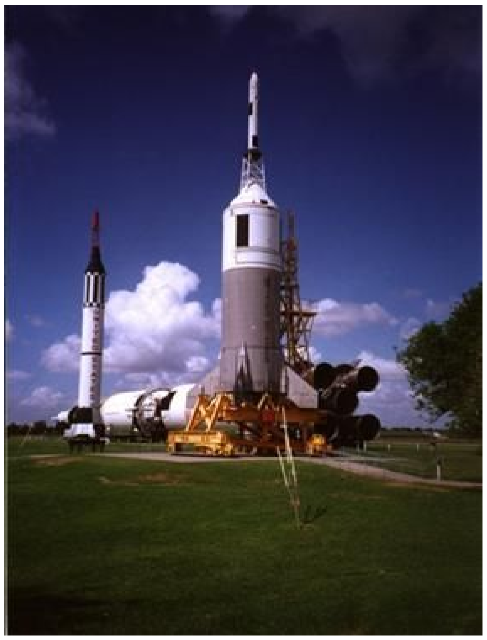
Histogram after brightness Decreased:



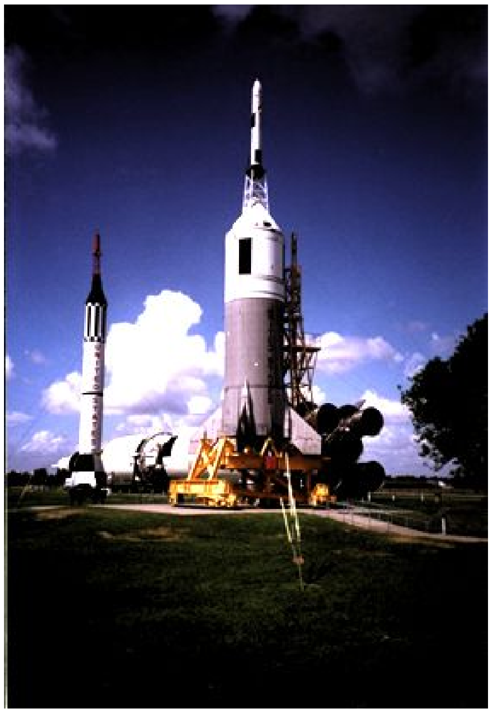
G. Increase contrast function.

<u>Ans:</u>

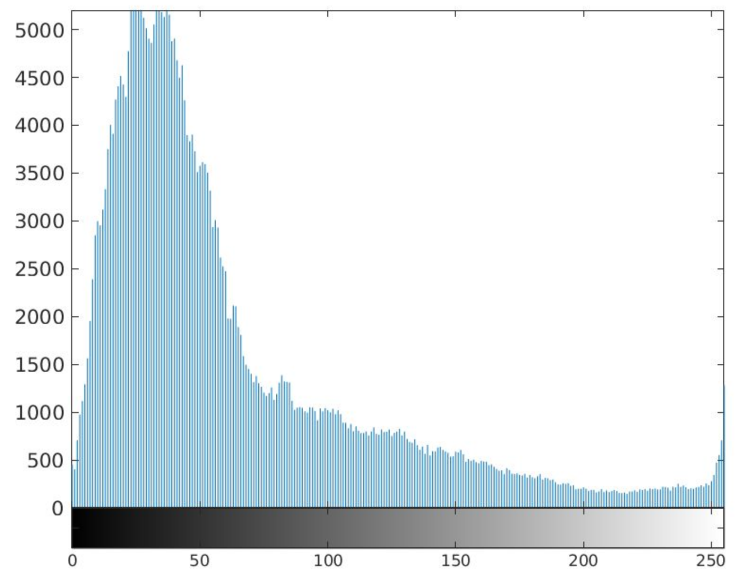
<u>Input Image:</u>



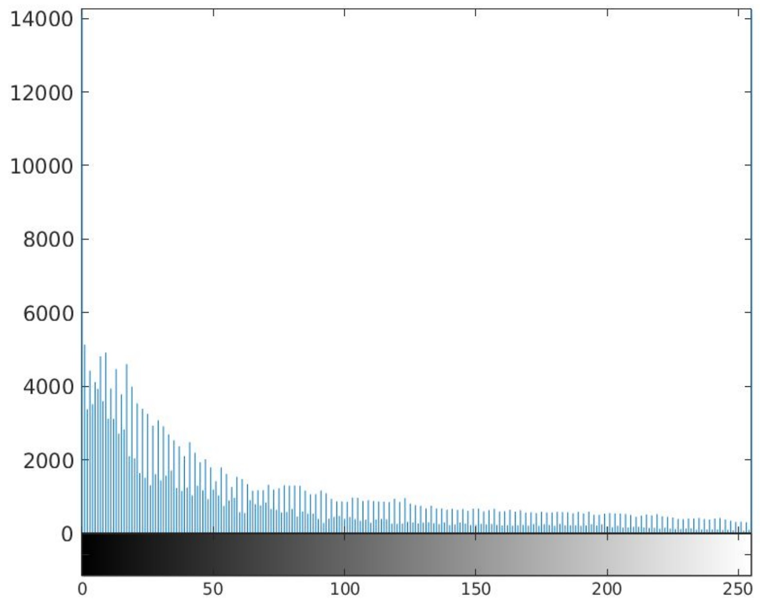
<u>Increased contrast:</u>



Original image histogram:

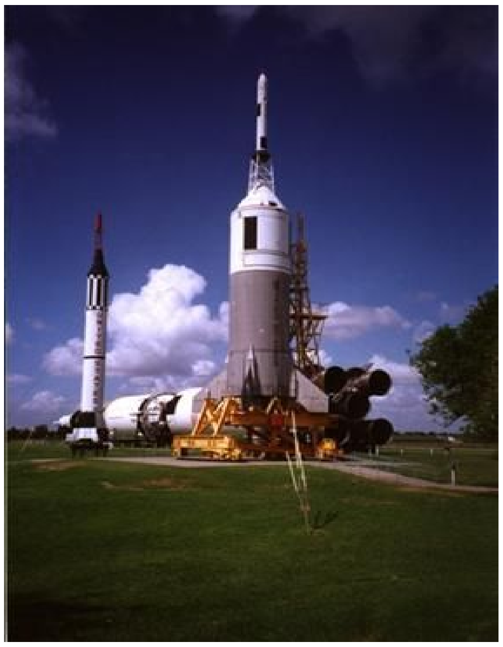


Histogram after Contrast increased:

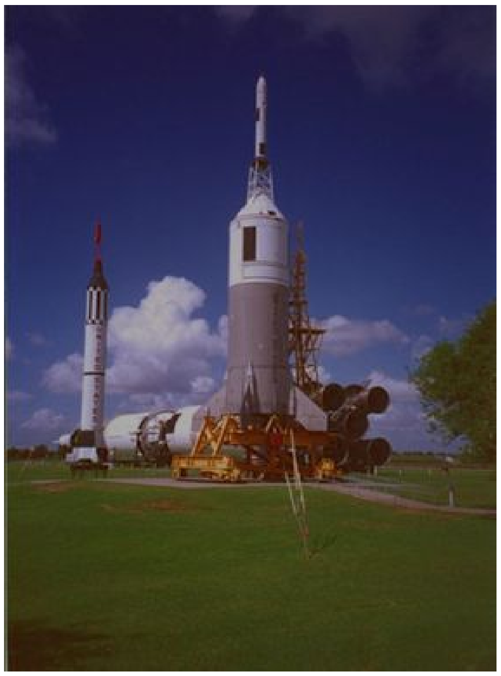


H. Decrease contrast function.
Ans:

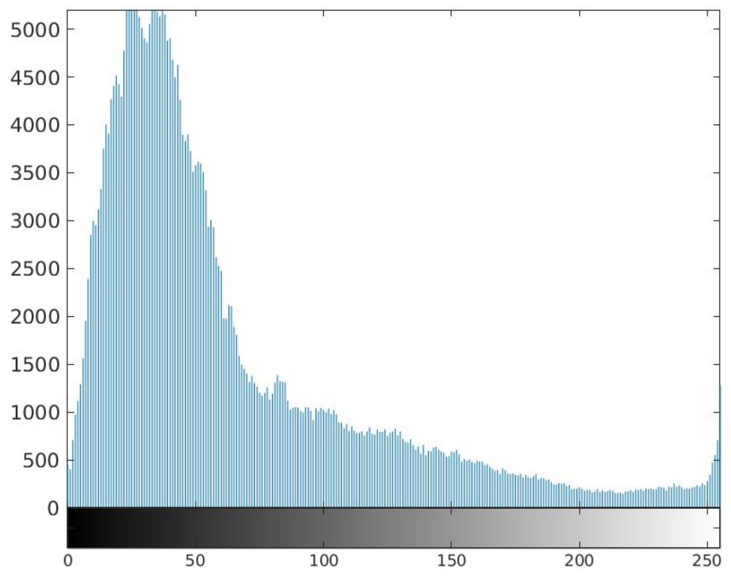
Input Image:



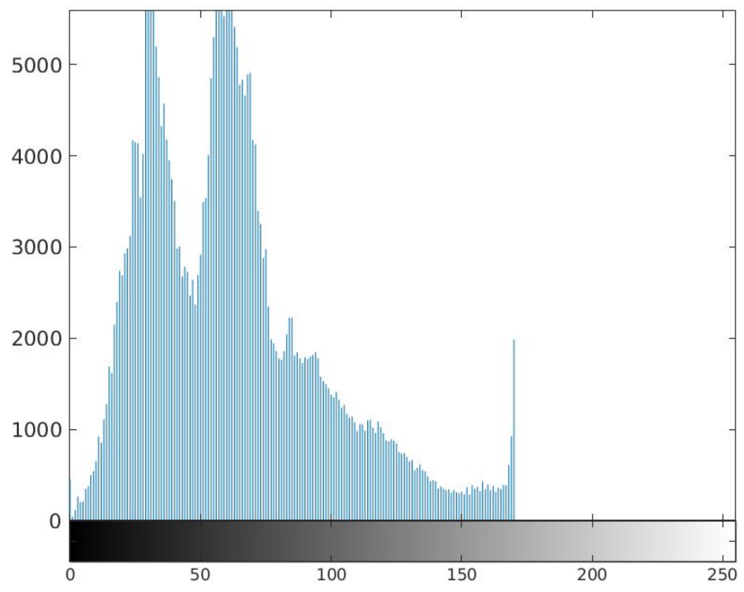
Decreased contrast:



Original image histogram:

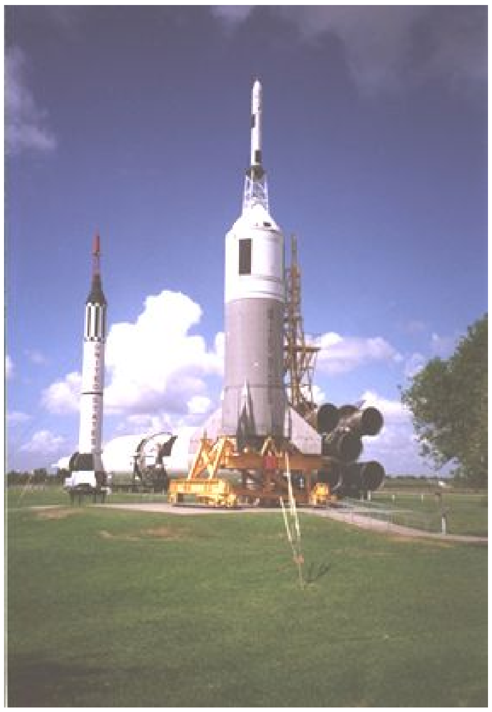


Histogram after Contrast decreased:

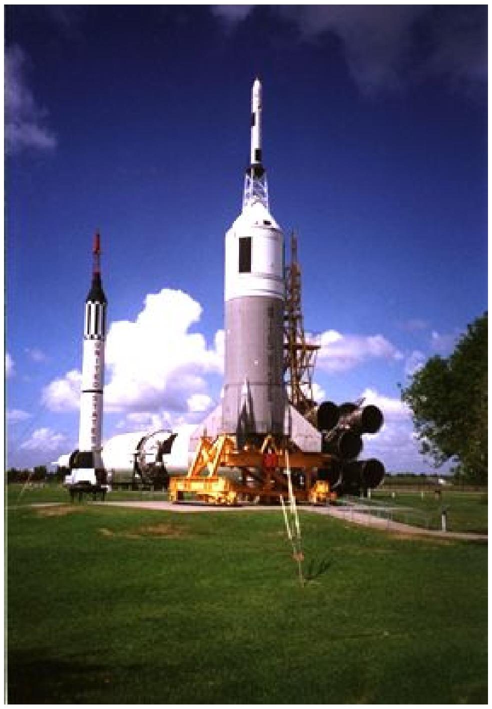


I. Stretch contrast function.
Ans:

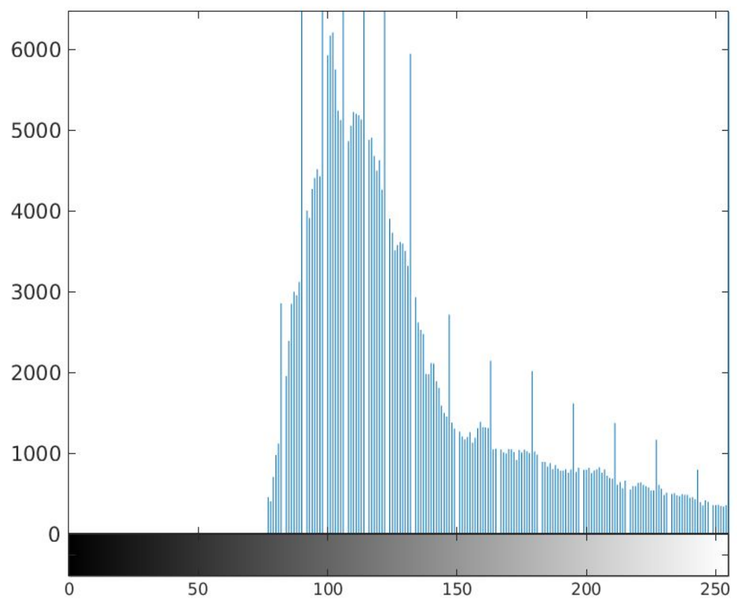
Input Image:



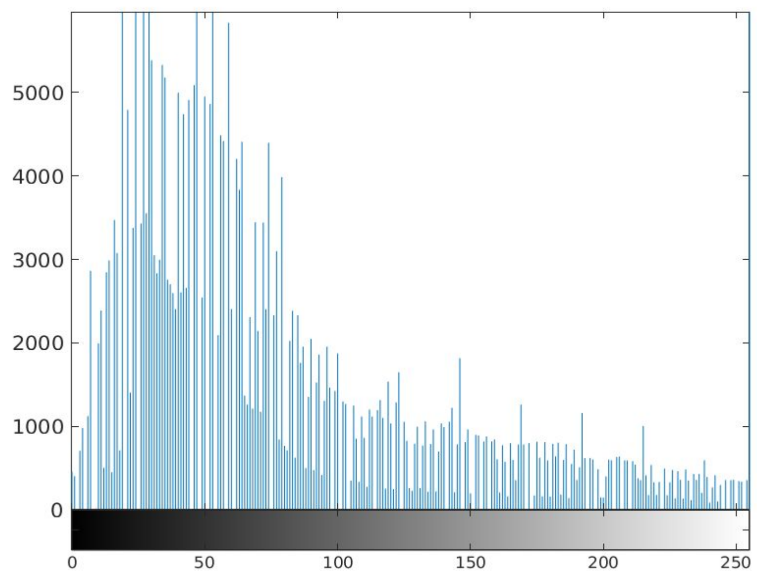
Stretched contrast image:



Histogram of input image:



Histogram of stretched contrast image:



k. Increase gamma function.

Ans:

Input image:

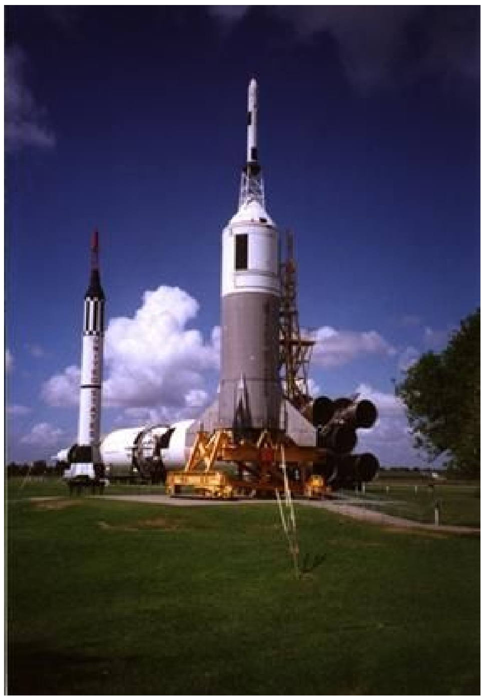
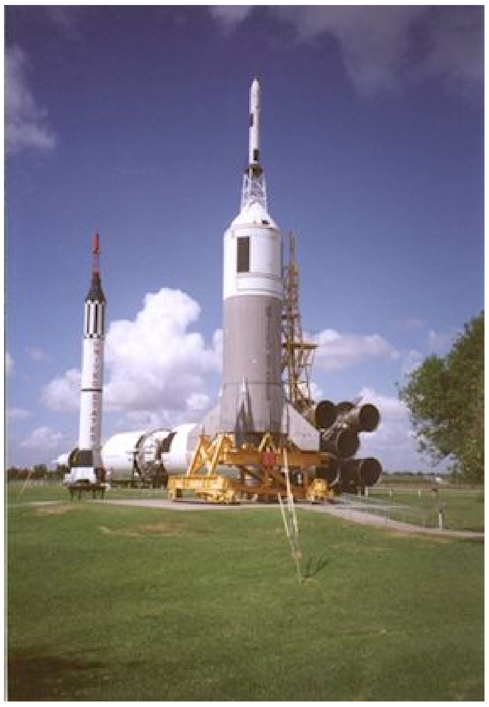
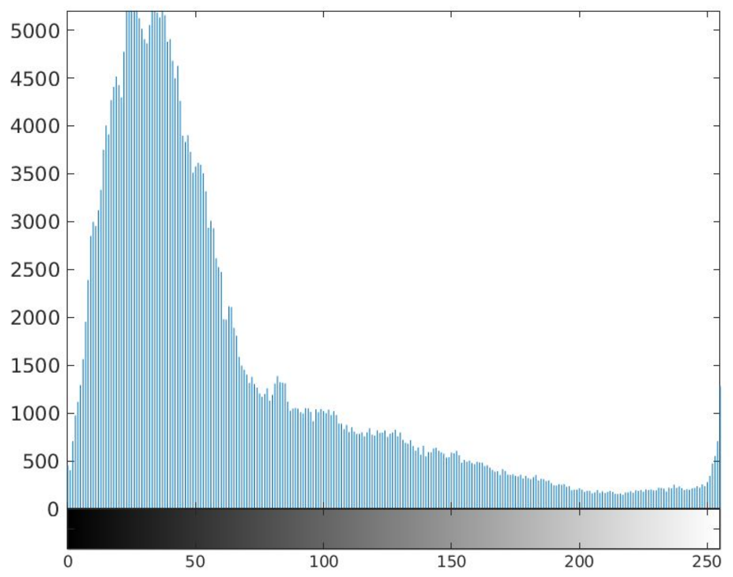


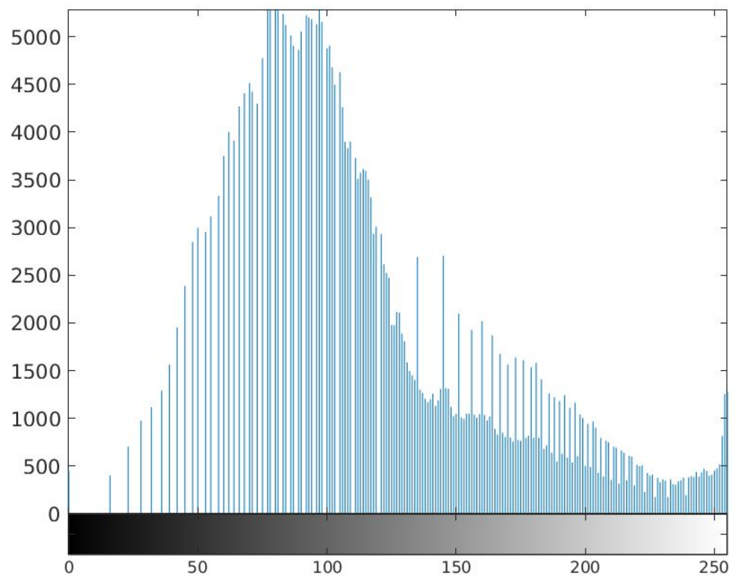
Image with increased gamma:



Histogram of original image:



Histogram of the image with increased gamma:



L. Decrease gamma function.

<u>Ans:</u>

Input image:

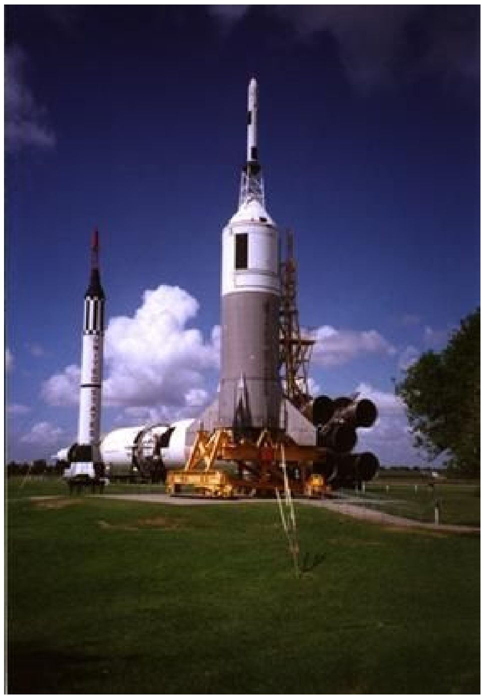
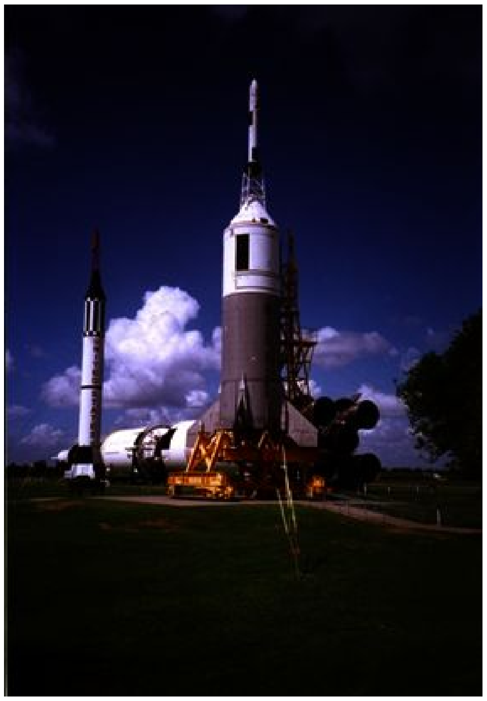
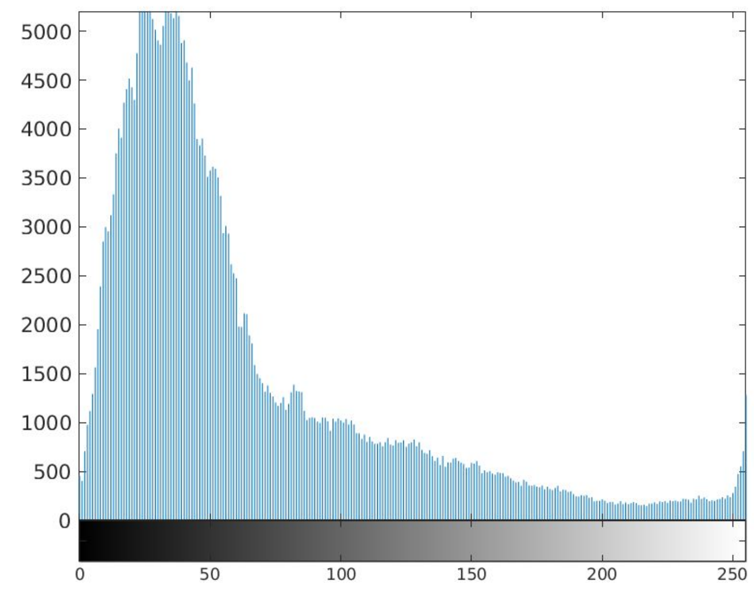


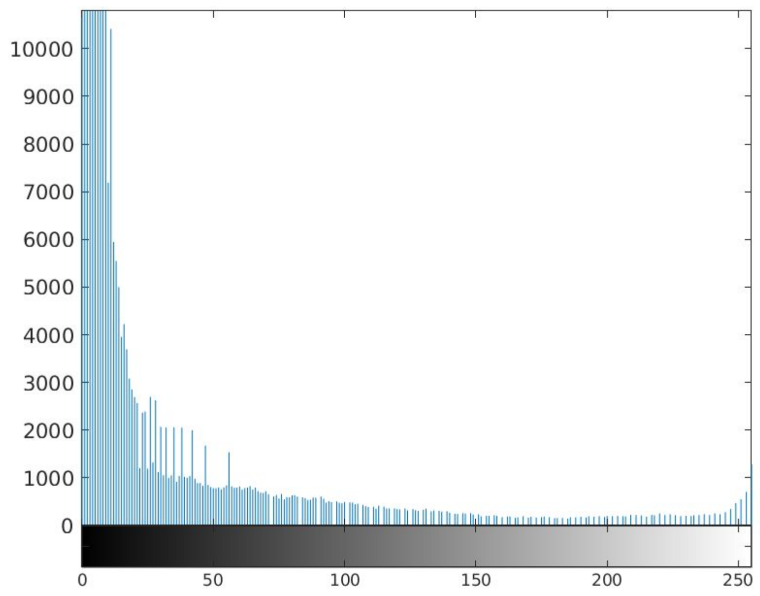
Image with decreased gamma:



Histogram of original image:



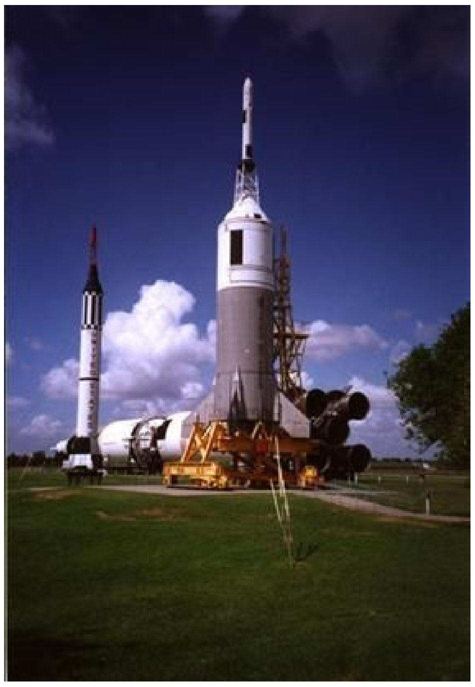
Histogram of the image with decreased gamma:



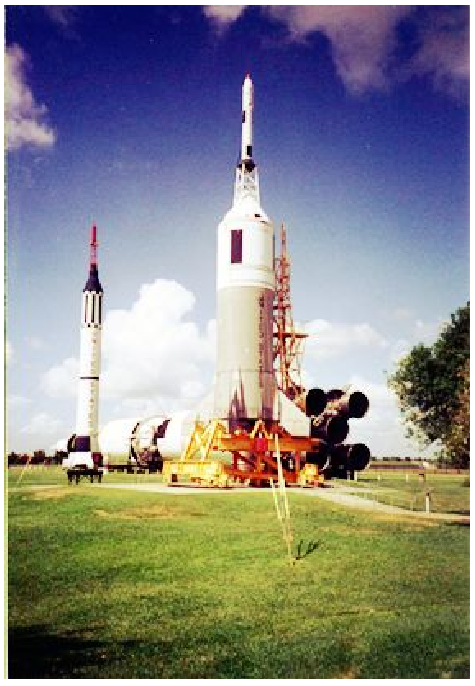
M. Histogram equalization function.

Ans:

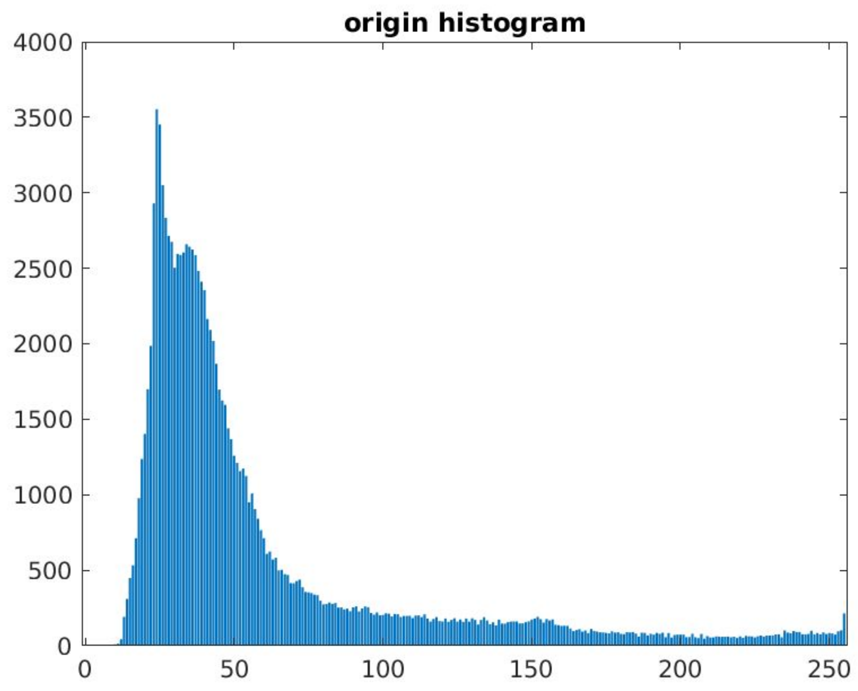
Input image:



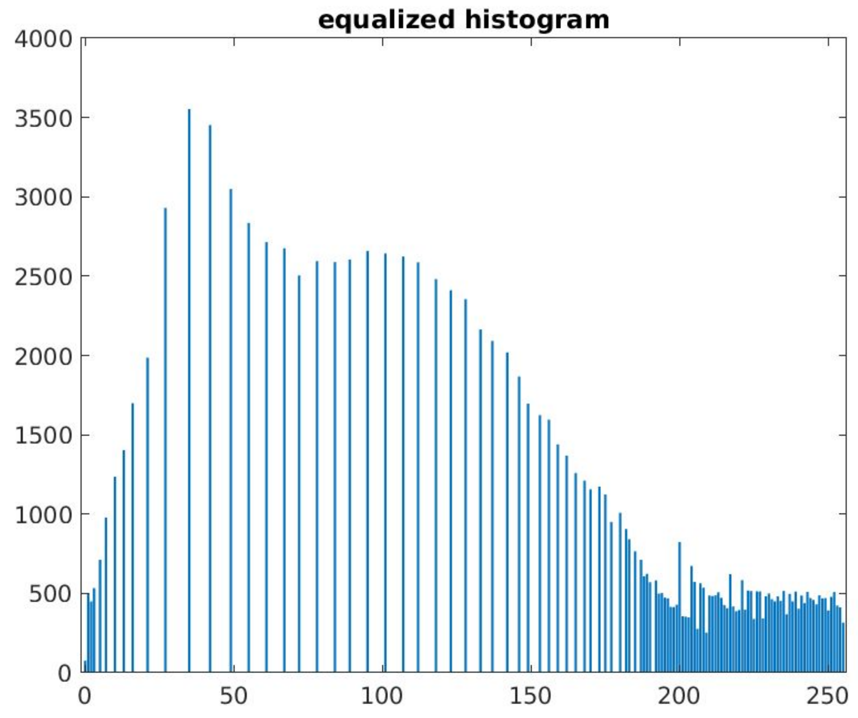
Histogram equalized image:



Input Image histogram:



Histogram after histogram equalization applied:



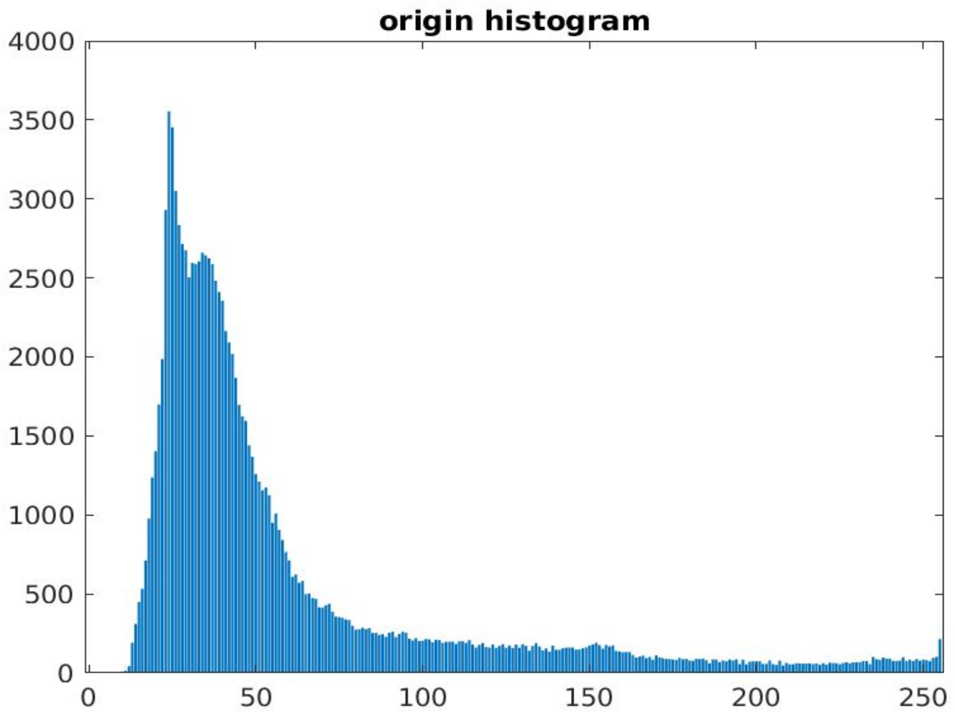
N. Histogram mapping function.

<u>Ans:</u>

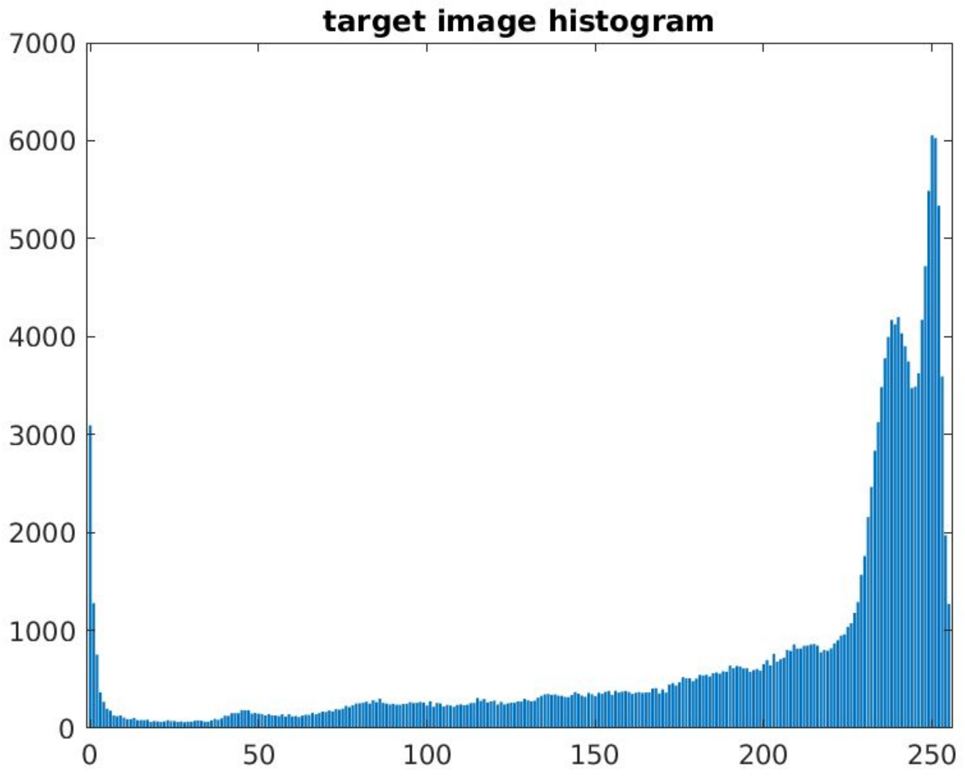
Input image:

Target image:

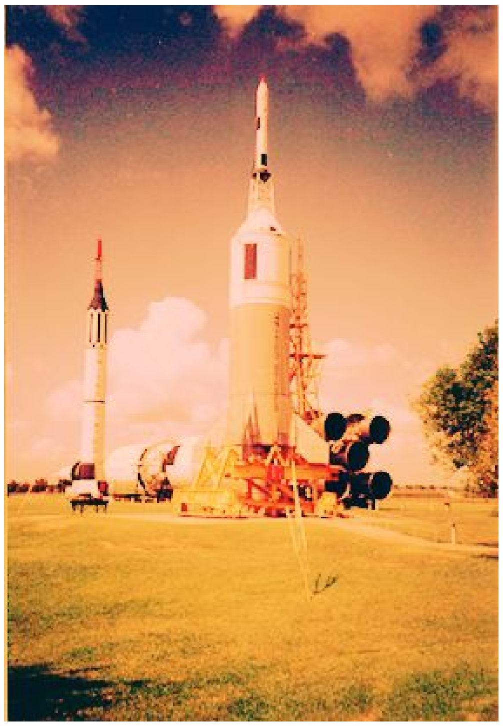
Histogram of the input image:



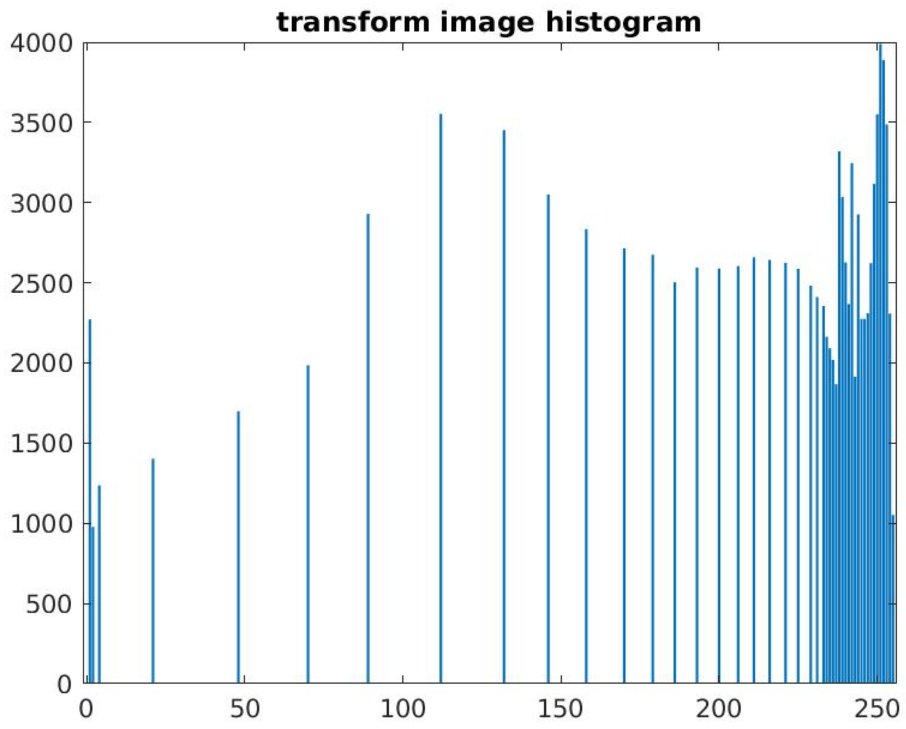
Histogram of the target image:



Mapped image:



Histogram of the mapped image:

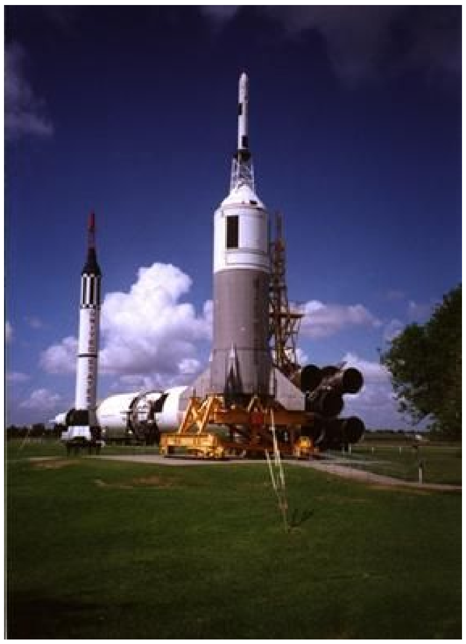


O. Report on my analysis of increasing and decreasing brightness of an image using a given image. Why such patterns in histogram appears?

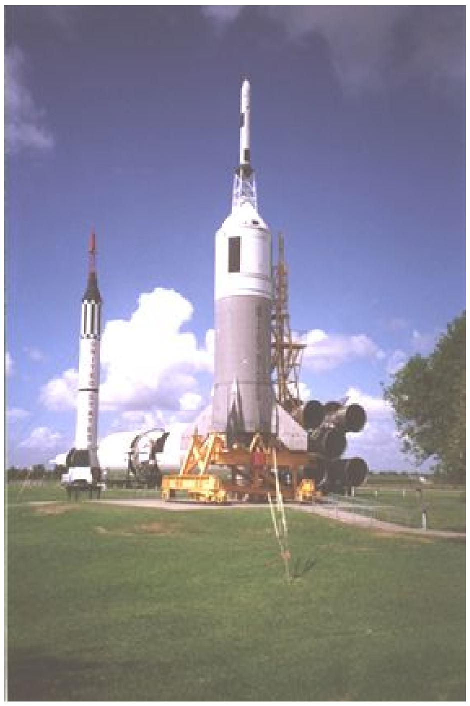
<u>**Ans:**</u>

<u>**Increase brightness function:**</u>

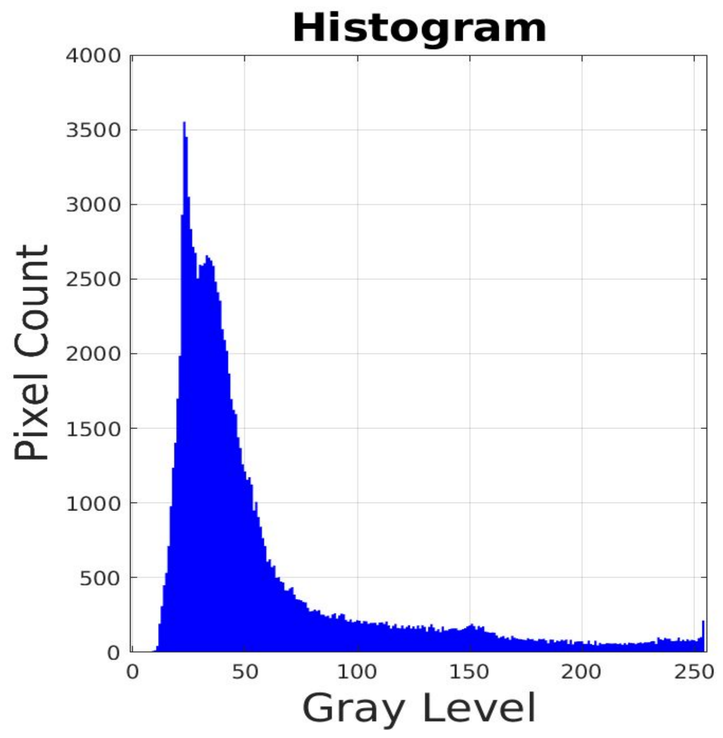
<u>Input Image:</u>



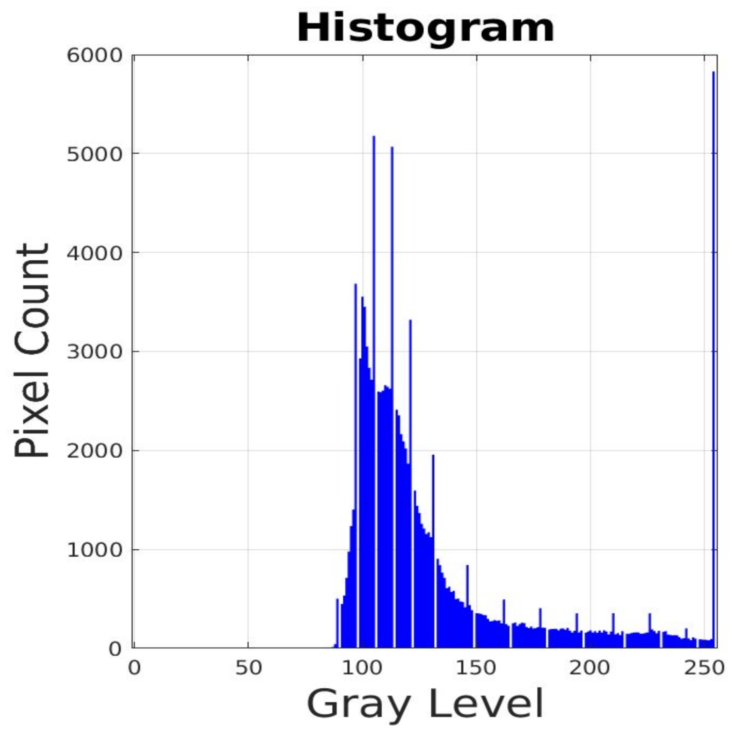
<u>Increased Brightness:</u>



Original image histogram:

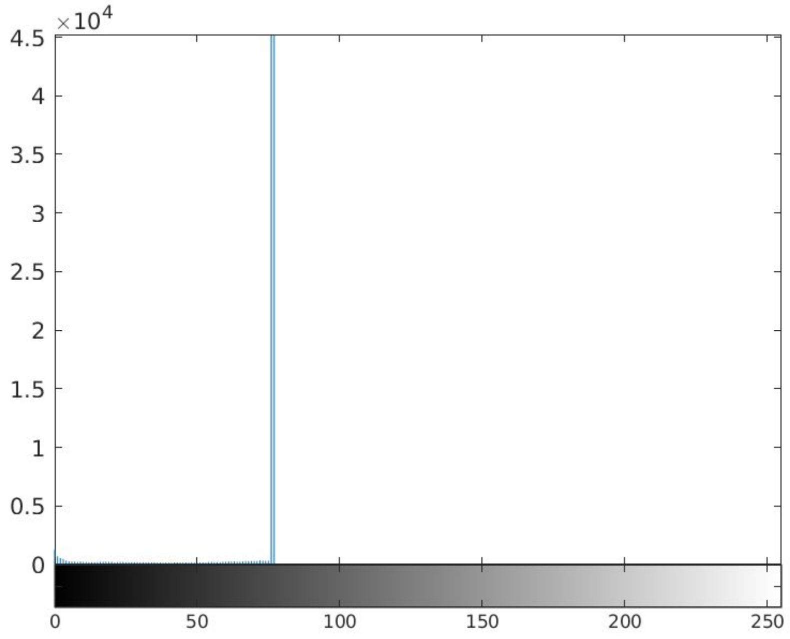


Histogram after brightness increased:



Difference between the input image and the brighter image:

Histogram of the difference image:



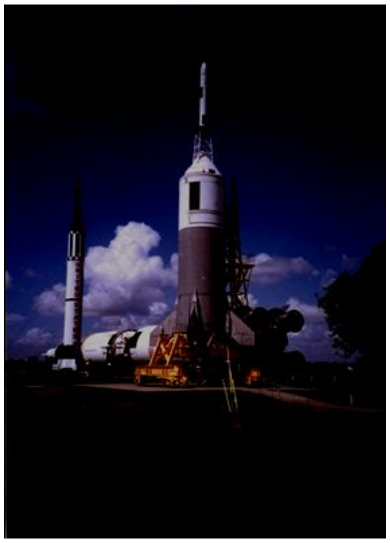
We increased the brightness of the image by adding constant number to the image matrix. Hence the whole histogram shifts to the brighter side by a constant ratio. The pixels that were already bright before brightening, they become 255 (the highest value).

If we subtract the difference image from the brightened image, we get close to the original image. Only the pixels whose values went upto 255 are quite not same as the original image even after transformed back because those values lost their integrity while brightening by going upto the highest limit(255).

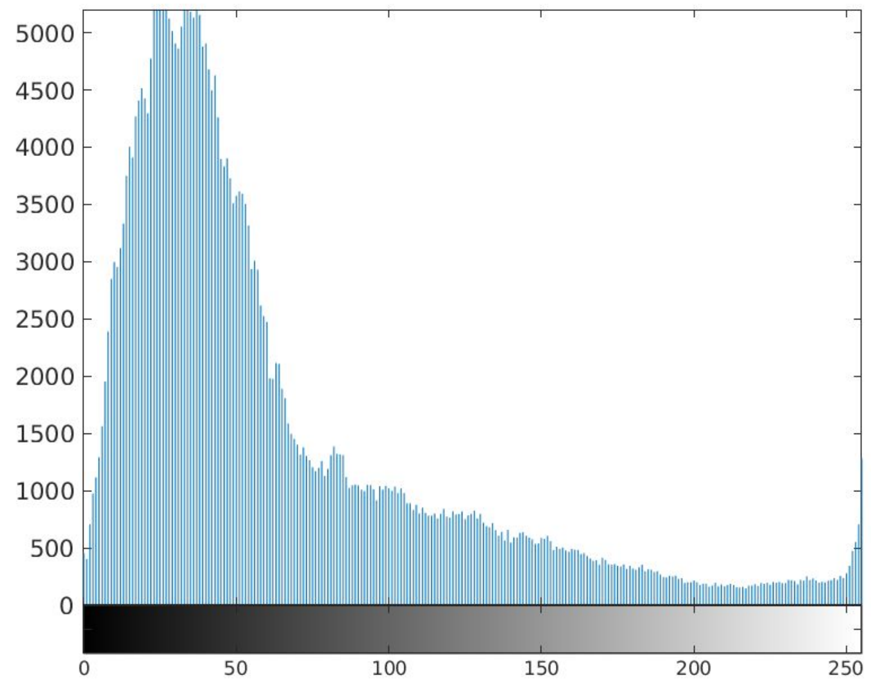
Decrease brightness function:

Input Image:

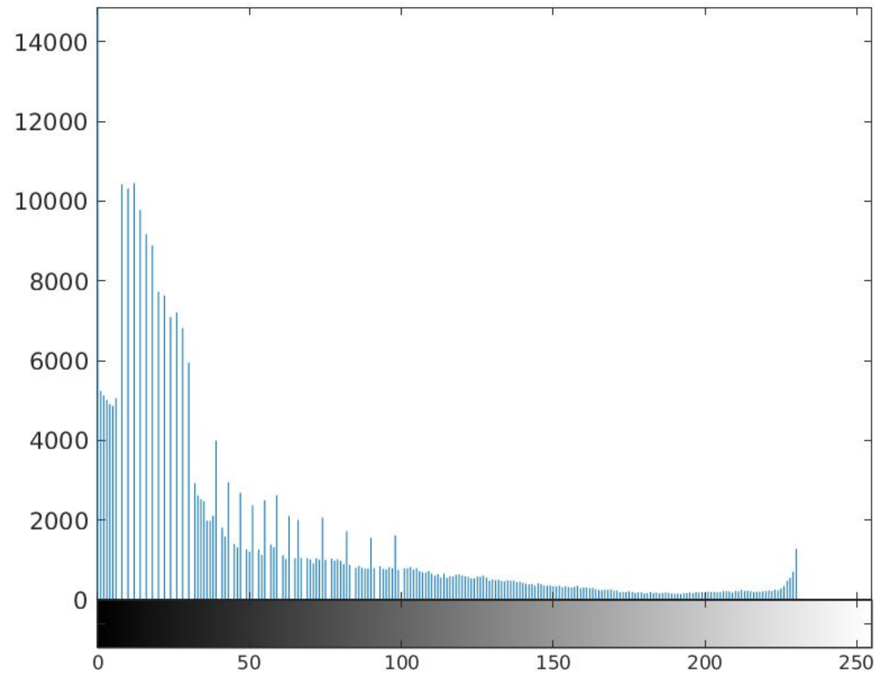
Decreased Brightness:



Original image histogram:

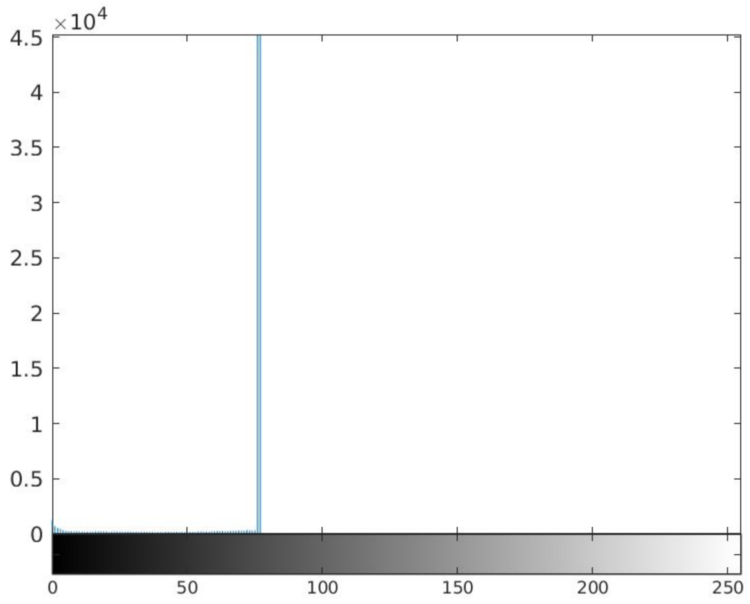


Histogram after brightness Decreased:



Difference between the input image and the darker image:

Histogram of the difference image:



We decreased the brightness of the image by subtracting constant number to the image matrix. Hence the whole histogram shifts to the darker side by a constant ratio. The pixels that were already darker before decreasing brightness, they become closer to 0(the lowest value).

If we add the difference image with the darker image, we get close to the original image. Only the pixels whose values went down to 0 are quite not same as the original image even after transformed back because those values lost their integrity while darkening by going down to the lowest limit(0).

P. Report on my analysis of increasing and decreasing contrast of an image using the given image. Why such patterns in the histogram appears?

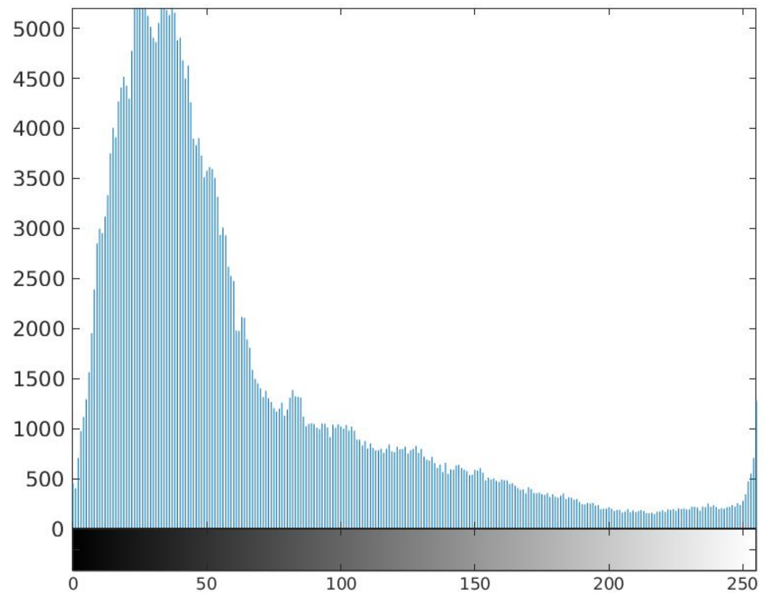
Ans:

Increase contrast function:

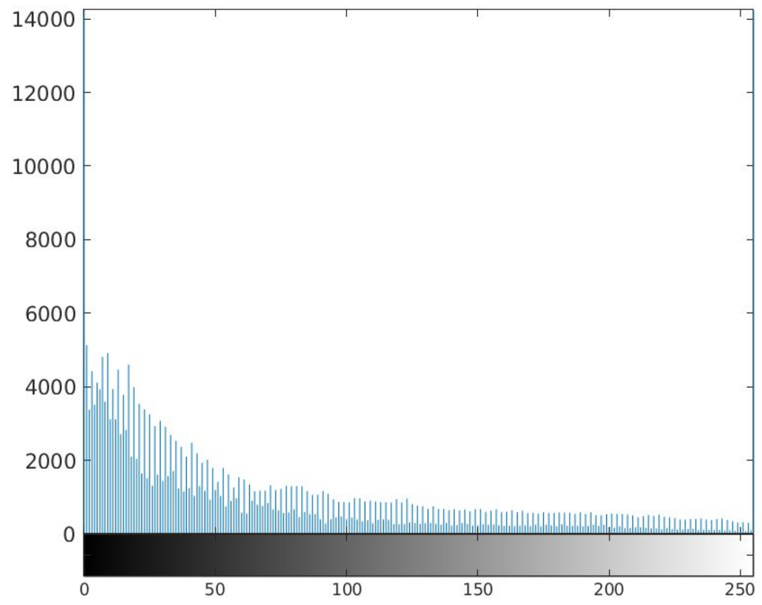
Input Image:

Increased contrast:

Original image histogram:



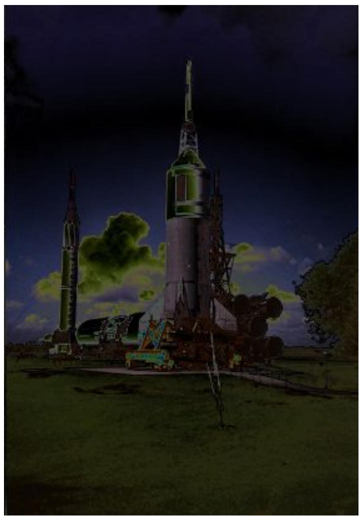
Histogram after Contrast increased:



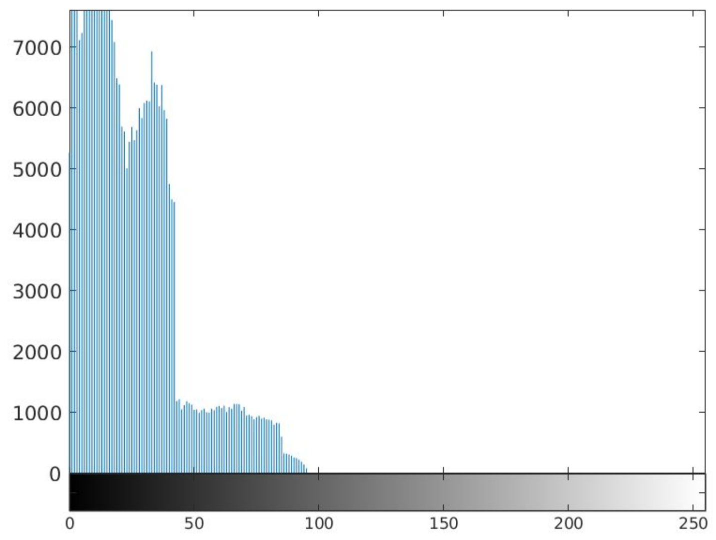
The contrast increment is done by first converting the RGB image to HSV image.
The mean of V(value) of HSV is then calculated. After that the value is changed by adding the mean V(Vm) and multiplying the (V-Vm) by a positive number larger than 1.
V = Vm + 2 * (V-Vm) .

So when the contrast is increased the darker pixels get more dark and the brighter pixels get more bright. Also pixels that are more in the middle of the histogram are spread to the left or right of the original position.

Difference between the original input vs the image with increased contrast:



Histogram of the difference image:



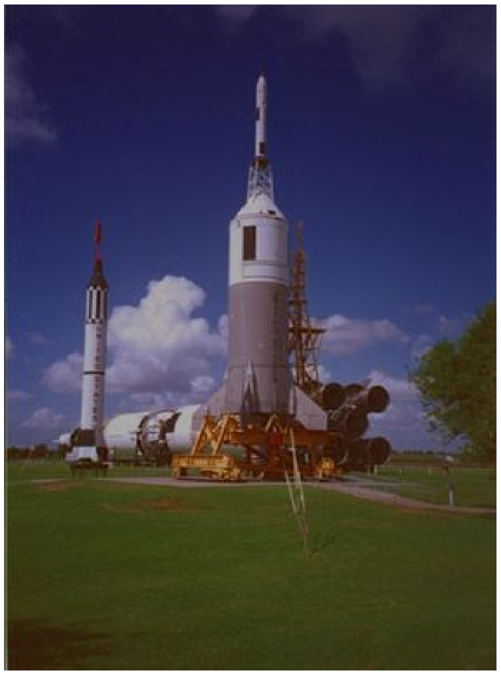
Subtracting the difference image from the increased contrast image gives a image closer to the original image.

But the darker pixels gets lost and becomes almost absolute black after converting back the image by subtracting the difference image from the high contrast image.

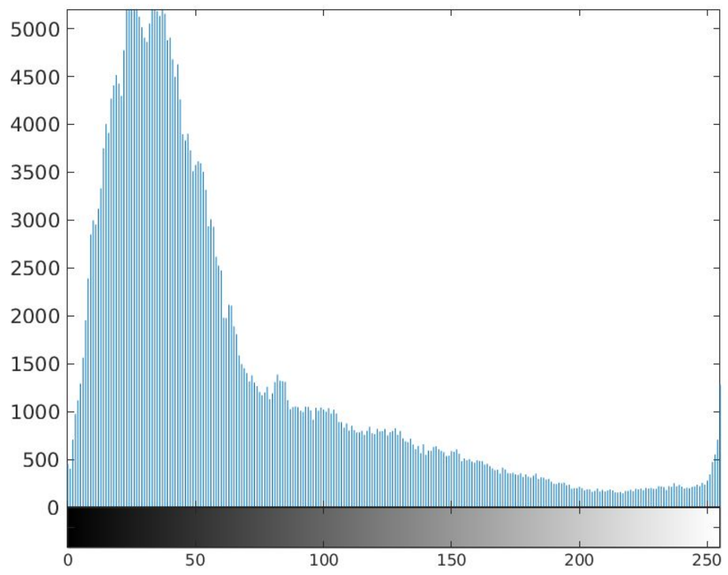
Decrease contrast function:

Input Image:

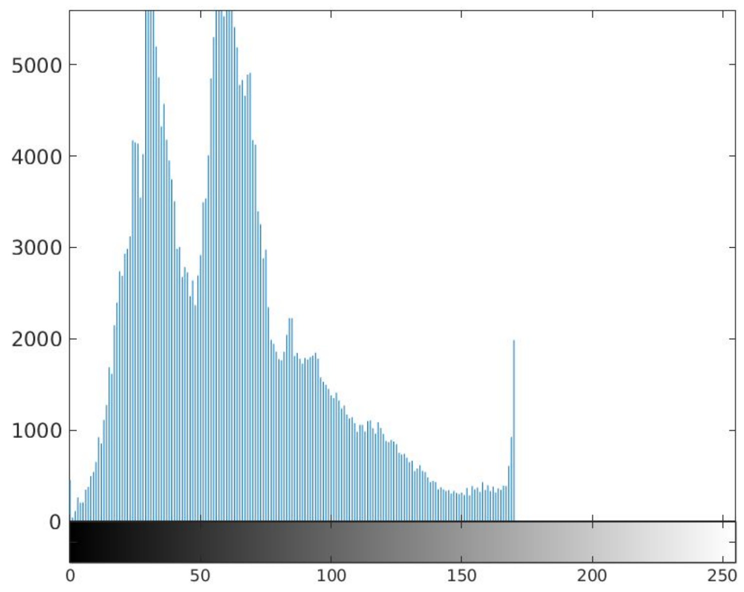
Decreased contrast:



Original image histogram:



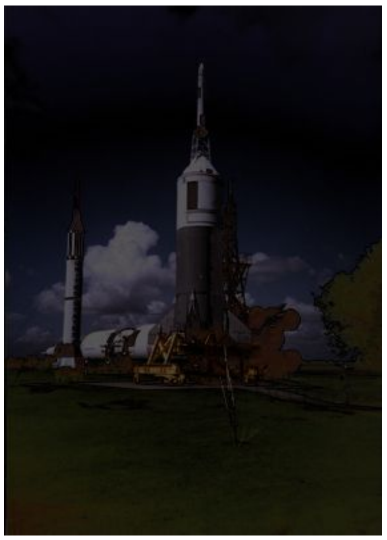
Histogram after Contrast decreased:



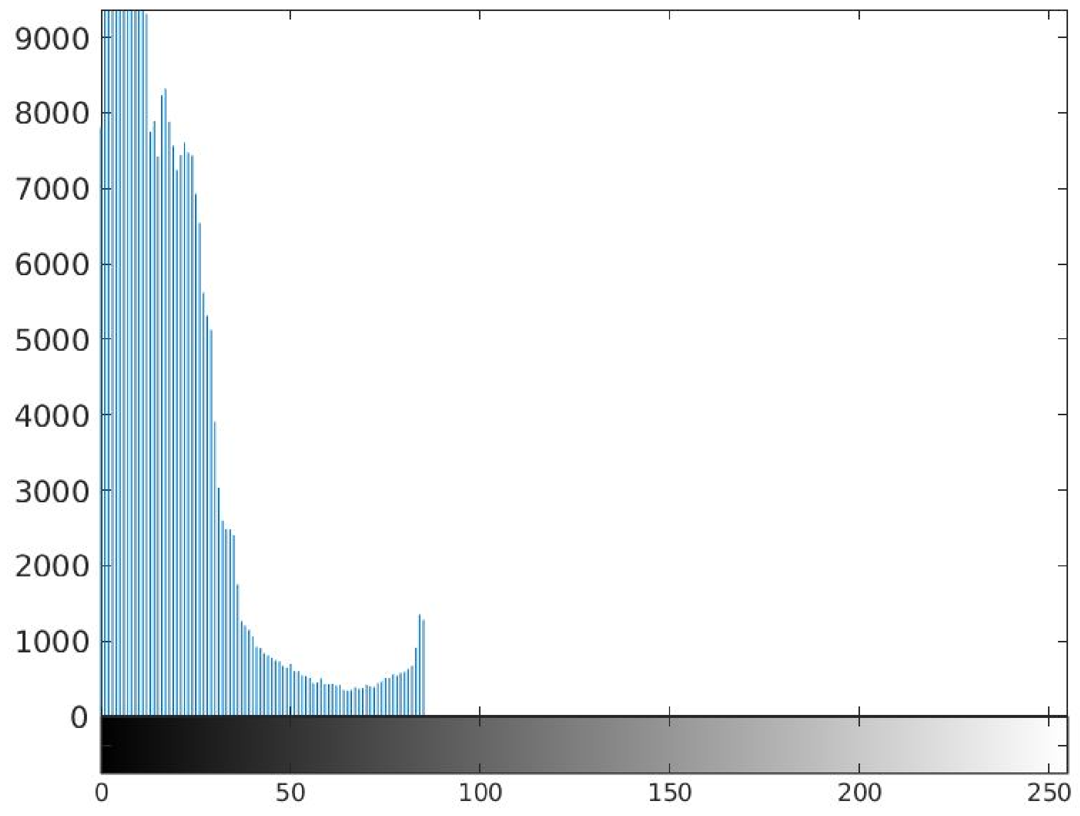
The contrast decrement is done by first converting the RGB image to HSV image.
The mean of V(value) of HSV is then calculated. After that the value is changed by adding the mean V(Vm) and multiplying the (V-Vm) by a positive number smaller than 1.
V = Vm + .5 * (V-Vm) .

So when the contrast is decreased the span of the histogram is decreased. The pixels closer to 0 or 255 are moved from their original position/value and towards the center of the histogram.

Difference between the original input vs the image with decreased contrast:



Histogram of the difference image:



Adding the difference image with the decreased contrast image gives a image closer to the original image.

But the brighter pixels looks washed out after converting back the image by adding the difference image with the low contrast image.

R. Report on my analysis of increasing and decreasing gamma of an image using the given image. Why such patterns in the histogram happens?

Ans:

Increase gamma function:

Input image:

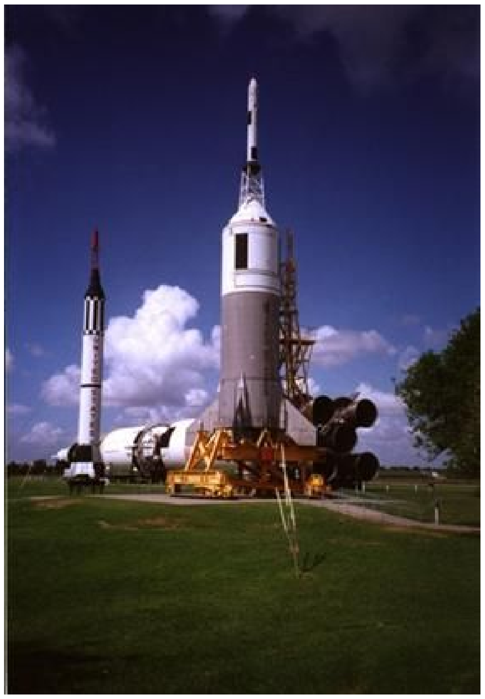
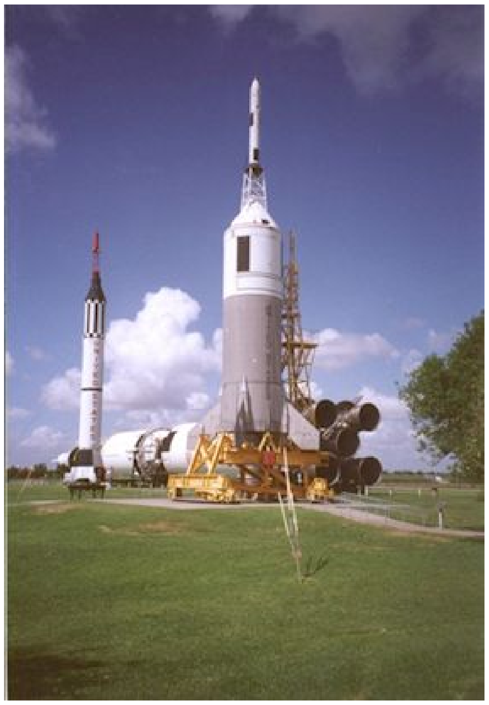
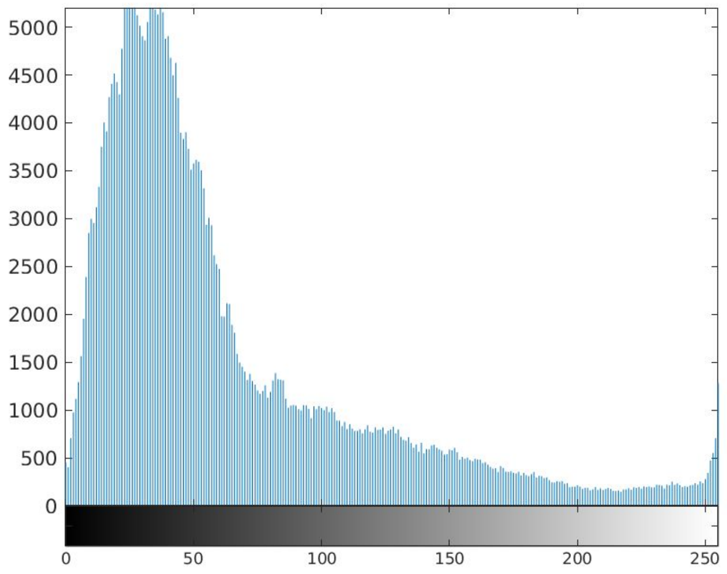


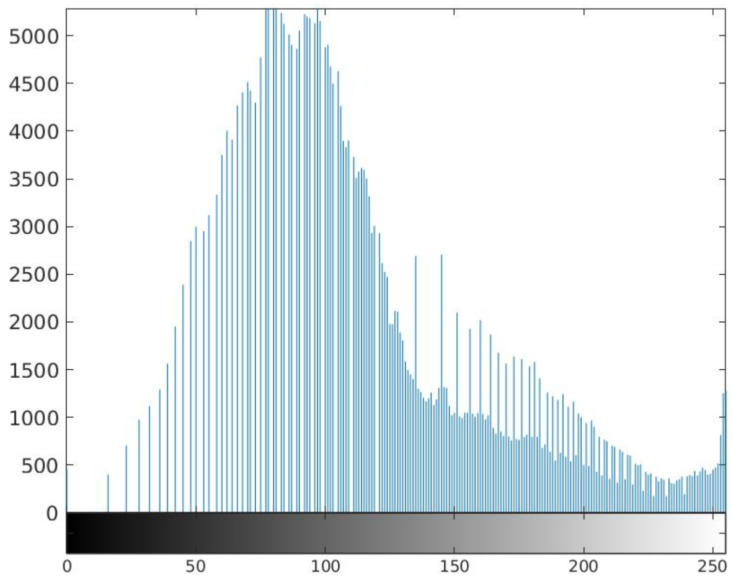
Image with increased gamma:



Histogram of original image:

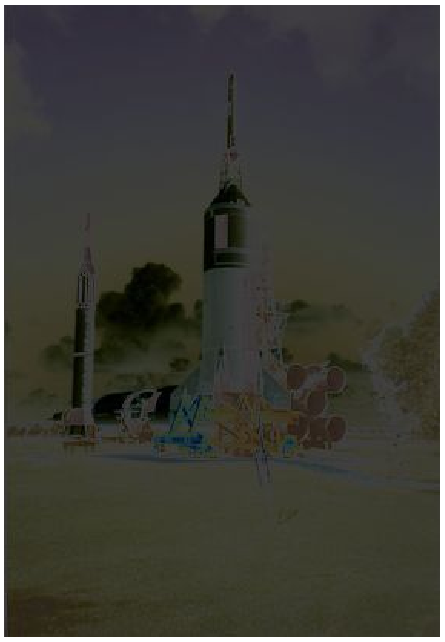


Histogram of the image with increased gamma:

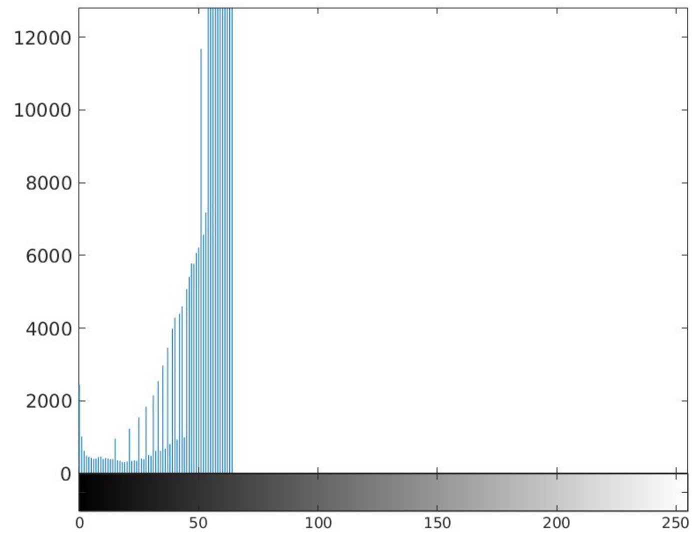


The histogram is shifted rightward (->255) when the gamma in increased by a factor of 2. To shift the histogram rightward, the density/frequency is loosened up in the left side of the histogram. Some upward and downward spike is observed in the histogram because multiple pixels end up having same value during gamma correction.

Difference image between the original and increased gamma :



Histogram of the difference image:



Decrease gamma function:

Ans:

Input image:

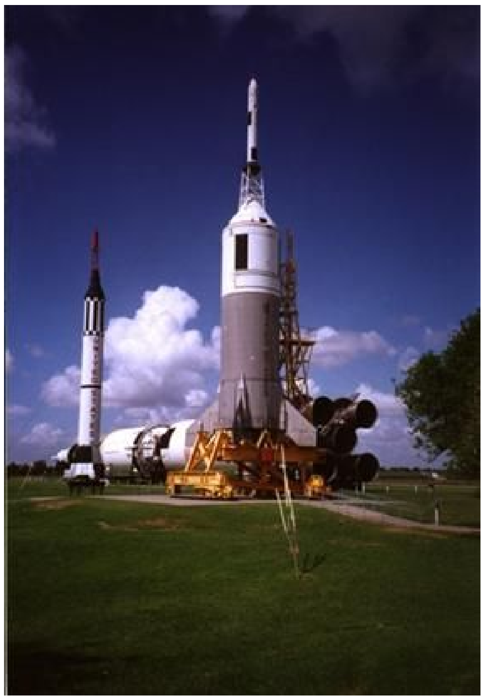
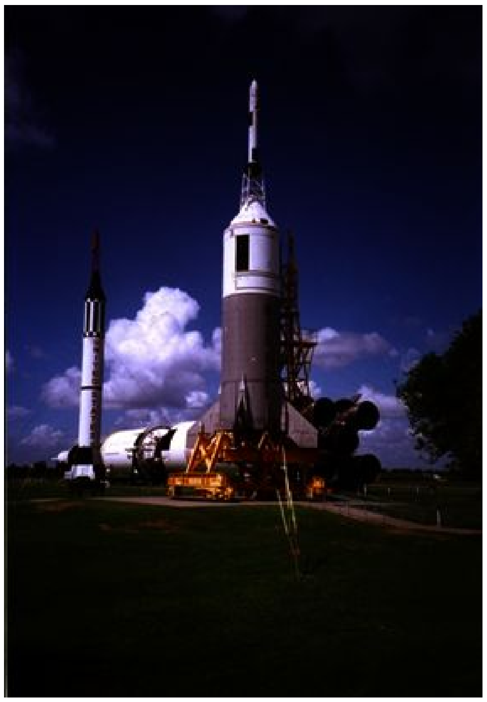
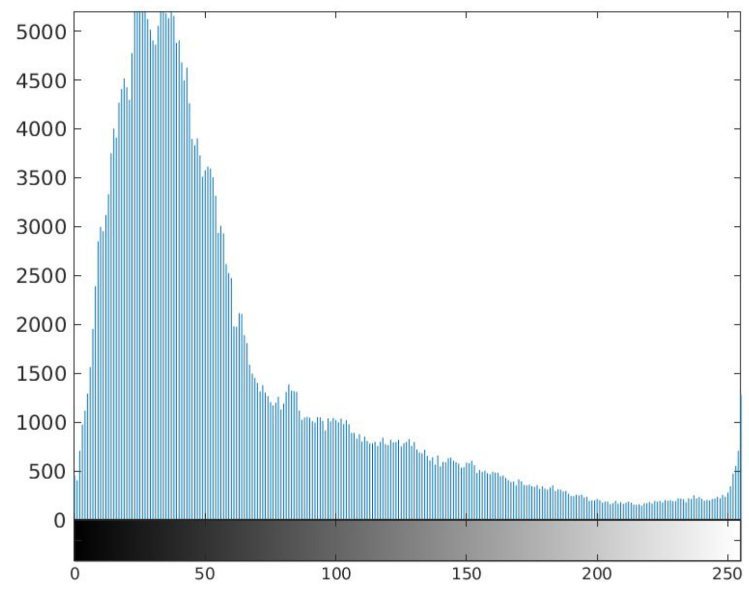


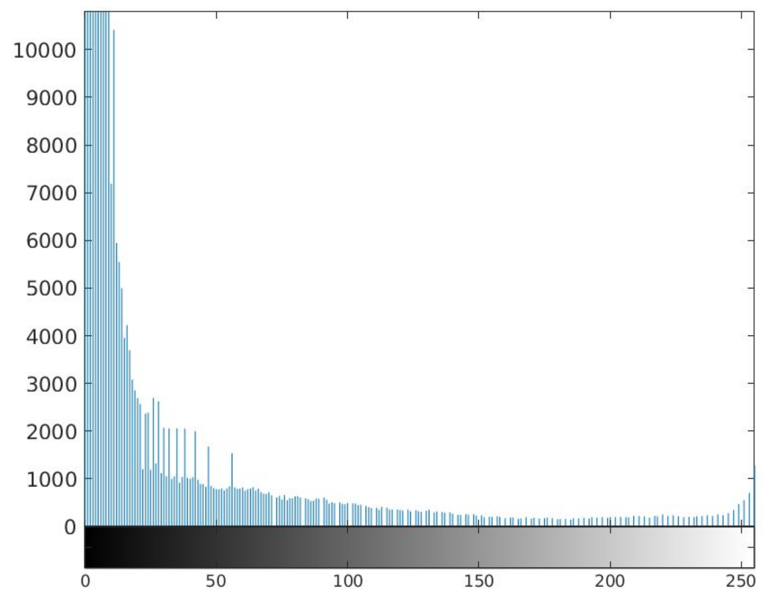
Image with decreased gamma:



Histogram of original image:

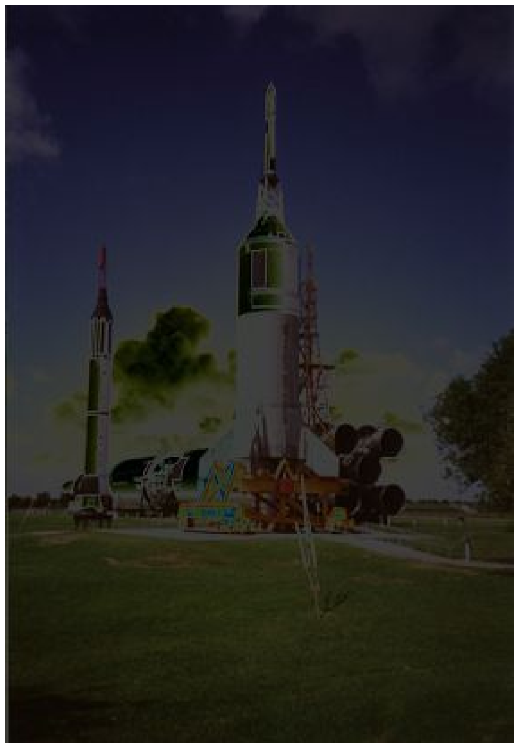


Histogram of the image with decreased gamma:

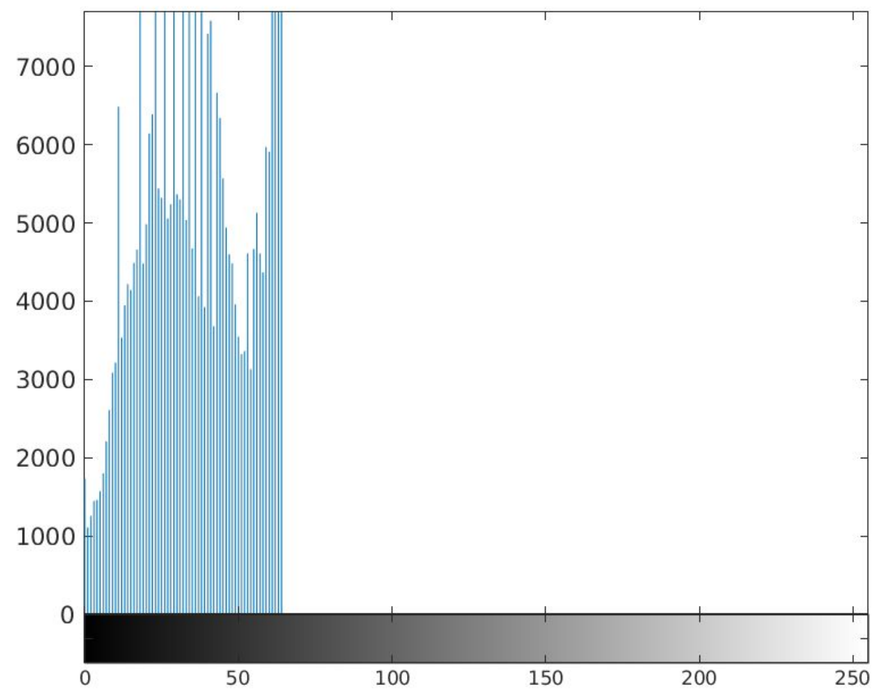


The histogram is shifted leftward (->0) when the gamma in decreased by half(1/2).
To shift the histogram leftward, the density/frequency is tightened up in the left side of the histogram. Some upward and downward spike is observed in the histogram because multiple pixels end up having same value during gamma correction.

Difference image between the original and decreased gamma :



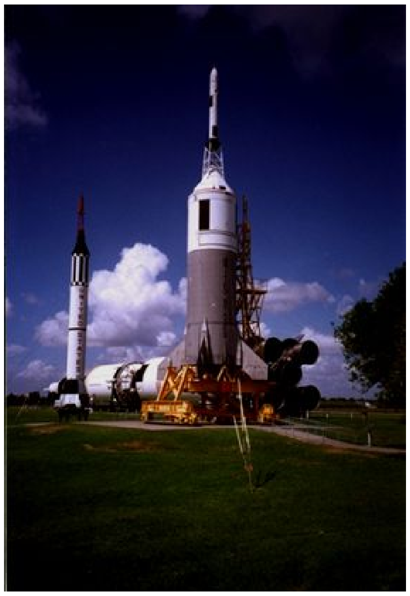
Histogram of the difference image:



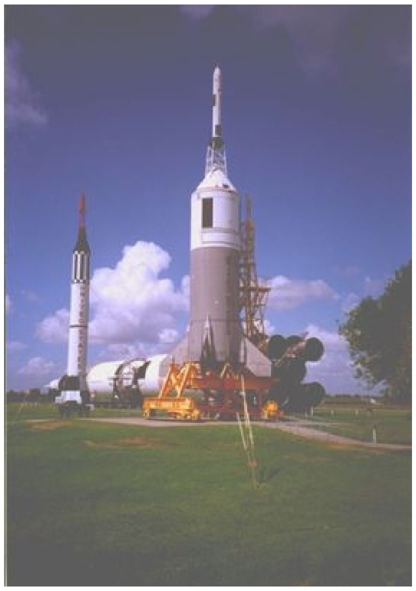
J. Automatically adjust Contrast and Brightness of an image.
Ans:

AutoBrighness

Input Image:



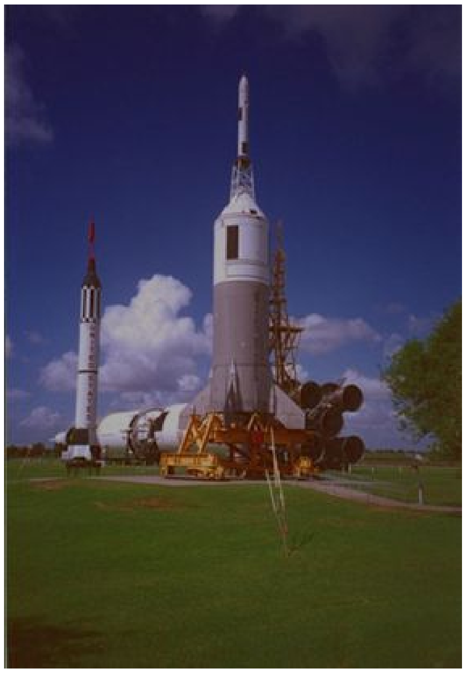
Output image with brightness auto adjusted:



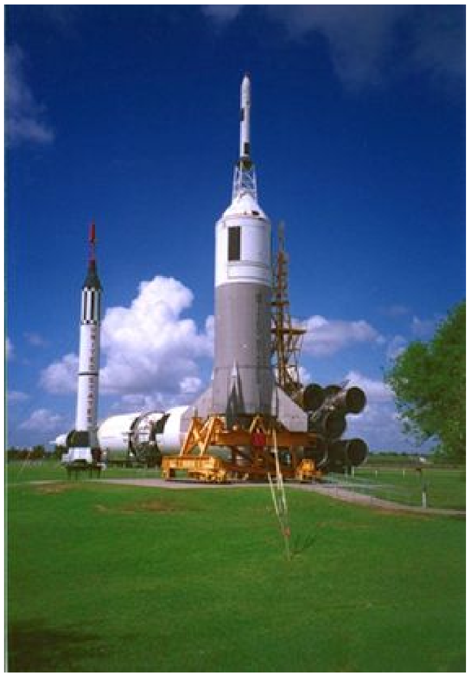
The quality(details) of the image is slightly lost due to the darker pixels becoming 255 when decreasing the brightness of the input image beforehand.

<u>AutoContrast:</u>

<u>Input image:</u>



<u>Output image with contrast auto adjusted:</u>



Some details are lost due to the lack of detail of the low contrast input image.

Appendix:

A)

```
%Monochromic image display for color images for color uses frequency analysis in paintings.

inputImage = imread('AtlasMercury.tif');
imshow(inputImage);

inputImage_r = inputImage(:,:,1);
inputImage_g = inputImage(:,:,2);
inputImage_b = inputImage(:,:,3);

figure;
imshow(inputImage_r);
imwrite(inputImage_r, 'band_r.jpg');
figure;
imshow(inputImage_g);
imwrite(inputImage_g, 'band_g.jpg');
figure;
imshow(inputImage_b);
imwrite(inputImage_b, 'band_b.jpg');
```

B)

```
inputImage = imread('Dec_bright.png');
imshow(inputImage);
% inputImage = rgb2gray(inputImage);
[counts, grayLevels] = MyHistogram(inputImage);

function [counts, grayLevels] = MyHistogram(inputImage)
[rows, columns, numberOfColorChannels] = size(inputImage);
counts = zeros(1, 256);
for col = 1 : columns
        for row = 1 : rows
                % Get the gray level.
                grayLevel = inputImage(row, col);
                % Add 1 because graylevel zero goes into index 1 and so on.
                counts(grayLevel) = counts(grayLevel) + 1;
        end
end
% Plot the histogram.
grayLevels = 0 : 255;
bar(grayLevels, counts, 'BarWidth', 1, 'FaceColor', 'b');
xlabel('Gray Level', 'FontSize', 20);
```

```
ylabel('Pixel Count', 'FontSize', 20);
title('Histogram', 'FontSize', 20);
grid on;
end
```

C)

```
I = imread('Dec_bright.png');
h_I = compute_hist(I);
x = 0 : 255;
figure;
image(I);
title('original image');
figure;
bar(x, h_I(:, 1));
title('original image histogram');

function h = compute_hist( img )
%compute histogram of an image
[m, n, c] = size(img);
h = zeros([256, c]);
        for i = 1 : 256
                h(i, :) = sum(sum(img == (i - 1)));
        end
end
```

D)

```
I = imread('AtlasMercury.tif');
h_I = compute_hist(I);
x = 0 : 255;
figure;
image(I);
title('origin image');
h = figure;
h_Val = (1/3)*(h_I(:, 1)+h_I(:, 2)+h_I(:, 3));
bar(x, h_Val(:, 1));
title('RGB image value histogram');
saveas(h,'hist_val.png')
k = figure;
h_Lum = (0.299*h_I(:, 1)+ 0.587*h_I(:, 2)+ 0.114*h_I(:, 3));
bar(x, h_Lum(:, 1));
title('RGB image Luminance histogram');
```

```
saveas(k,'hist_lum.png')
function h = compute_hist( img )
%compute histogram of an image

        [m, n, c] = size(img);
        h = zeros([256, c]);
        for i = 1 : 256
                h(i, :) = sum(sum(img == (i - 1)));
        end
end
```

E)

```
img = imread('AtlasMercury.tif');
img = im2double(img);
figure;
imshow(img);
title('Original Image');

IncreasedBrightnessImg = img + .3;

figure;
imshow(IncreasedBrightnessImg);
imwrite(IncreasedBrightnessImg,'Inc_bright.png');
title('Increased Brightness');
```

F)

```
img = imread('AtlasMercury.tif');
img = im2double(img);
figure;
imshow(img);
title('Original Image');


DecreasedBrightnessImg = img - .1;
imwrite(DecreasedBrightnessImg,'Dec_bright.png');
figure;
imshow(DecreasedBrightnessImg);
title('Decreased Brightness');
```

G)

```
rgbImage = imread('AtlasMercury.tif');
hsvImage = rgb2hsv(rgbImage);
hChannel = hsvImage(:, :, 1);
sChannel = hsvImage(:, :, 2);
vChannel = hsvImage(:, :, 3);
meanV = mean2(vChannel);
newV = meanV + 2 * (vChannel - meanV); % Increase contrast by factor of 2
newHSVImage = cat(3, hChannel, sChannel, newV);
newRGBImage = hsv2rgb(newHSVImage);
imshow(rgbImage);
figure;
imshow(newRGBImage);
imwrite(newRGBImage,'contrast_inc.png');
```

H)

```
rgbImage = imread('AtlasMercury.tif');
hsvImage = rgb2hsv(rgbImage);
hChannel = hsvImage(:, :, 1);
sChannel = hsvImage(:, :, 2);
vChannel = hsvImage(:, :, 3);
meanV = mean2(vChannel);
newV = meanV + .5 * (vChannel - meanV); % Decrease contrast by factor of 2
newHSVImage = cat(3, hChannel, sChannel, newV);
newRGBImage = hsv2rgb(newHSVImage);
imshow(rgbImage);
figure;
imshow(newRGBImage);
imwrite(newRGBImage,'contrast_dec.png');
```

I)

```
y=imread('Inc_bright.png');
info=imfinfo('Inc_bright.png');
M=info.BitDepth;
G=2^M;

[m n]=size(y);

imshow(y)
title('original image')
```

```
%%%%%%%%%Contrast Stretching%%%%%%%%%%%
a=input('Enter the value of input graylevel r1 for contrast stretching:');
b=input('Enter the value of input graylevel r2 for contrast stretching:');
for i=1:m
   for j=1:n
      if y(i,j)<=a
         zz(i,j)=0.5*y(i,j);
      else if y(i,j)<=b
            zz(i,j)=2*(y(i,j)-a)+0.5*a;
         else
            zz(i,j)=0.5*(y(i,j)-b)+0.5*a+2*(b-a);
         end
      end
   end
end
figure
imshow(zz);
title('contrast stretched image');
```

J)
AutoBrightness:

```
input_image =imread('Dec_bright.png');
output_img = 'autoImg.png';
my_limit=0.5;
if size(input_image,3)==3
   a=rgb2ntsc(input_image);
else
   a=double(input_image)./255;
end
mean_adjustment=my_limit-mean(mean(a(:,:,1)));
a(:,:,1)=a(:,:,1)+mean_adjustment*(1-a(:,:,1));
if size(input_image,3)==3
   a=ntsc2rgb(a);
end
imwrite(uint8(a.*255),output_img);
```

AutoContrast:

```
f = imread('contrast_dec.png');
f1 = im2double(f); % Cast to double.
channelCount = size(f1 ,3);
%f3 = zeros(size(f1,1) ,size(f1, 2), size(f1, 3));
for i = 1:channelCount
   fmatrix = f1(:,:,i);
   min1 = min(min(fmatrix));
   max1 = max(max(fmatrix));
   f2 = round( ((fmatrix - min1)*255)/(max1- min1));
   f3(:,:,i) = uint8(255 * mat2gray(f2));
end

myNumOfColors = 200;
myColorScale = [[0:1/(myNumOfColors - 1):1]',[0:1/(myNumOfColors - 1):1]' ,
[0:1/(myNumOfColors - 1):1]' ];

for i=1:channelCount
   subplot(channelCount,2,(2*i) -1);
   imagesc (f(:, :, i));
   if (channelCount == 1)
      title('Original');
   elseif (i == 1)
      title('Original R');
   elseif (i == 2)
      title('Original G');
   else
      title('Original B');
   end
   colormap (myColorScale);
   colormap (jet);
   daspect ([1 1 1]);
   axis tight;
   colorbar
   subplot(channelCount,2,2*i);
   imagesc (f3(:,:,i));
   if (channelCount == 1)
      title('Contrast-streched');
   elseif (i == 1)
      title('Contrast-streched R');
   elseif (i == 2)
      title('Contrast-streched G');
```

```
    else
        title('Contrast-streched B');
    end
    colormap (myColorScale);
    colormap (jet);
    daspect ([1 1 1]);
    axis tight;
    colorbar
end

set(gcf,'Position',get(0,'ScreenSize'));%maximize figure
imwrite(f3,'canyon_LinearContrastStretching.png');
```

k)

```
img = imread('AtlasMercury.tif');
img2 = uint8(size(img));
gam = 2;
for i=1:3
    for j=1:413
        for k=1:284
            img2(j,k,i)= 255*((double(img(j,k,i))/255)^double(1/gam));
        end
    end
end

imshow(img);
figure;
imshow(img2);
imwrite(img2,'Inc_gamma.png');
```

L)

```
img = imread('AtlasMercury.tif');
img2 = uint8(size(img));
gam = 1/2;
for i=1:3
    for j=1:413
        for k=1:284
            img2(j,k,i)= 255*((double(img(j,k,i))/255)^double(1/gam));
        end
    end
end
```

```
imshow(img);
figure;
imshow(img2);
imwrite(img2,'Dec_gamma.png');
```

M)

<u>Compute_hist.m</u>

```
function h = compute_hist( img )
%compute histogram of an image

[m, n, c] = size(img);
h = zeros([256, c]);
for i = 1 : 256
    h(i, :) = sum(sum(img == (i - 1)));
end
end
```

<u>Histogram_equalization.m</u>

```
function [ h, lut, eq_I] = HistogramEqualization( I )
%HistogramEqualization
%   compute histogram of an image and do histogram equalization
[m, n, c] = size(I);
h = compute_hist(I);
[m_h, n_h] = size(h);
total = sum(h);

% construct lut
lut = zeros([m_h, n_h]);
lut(1, :) = h(1, :) ./ total;
for i = 2 : m_h
    lut(i, :) = lut(i - 1, :) + h(i, :) ./ total;
end
for i = 1 : m_h
    lut(i, :) = round(lut(i, :) .* 255);
end
eq_I = zeros(size(I));
for i = 1 : m
    for j = 1 : n
        for k = 1 : c
            eq_I(i, j, k) = lut(I(i, j, k) + 1, k);
        end
    end
end
eq_I = uint8(eq_I);
end
```

Test_eq.m

```
I = imread('AtlasMercury.tif');
[h, lut, eq_I] = HistogramEqualization(I);
eq_h = compute_hist(eq_I);
figure;
image(I);
title('origin image');
figure;
image(eq_I);
title('hist equalization image');
imwrite(eq_I, 'Hist_eqed.png');

x = 0 : 255;
h1_img = figure;
bar(x, h(:, 1));
title('origin histogram');
saveas(h1_img, 'original_hist.png');
h2_img = figure;
bar(x, eq_h(:, 1));
title('equalized histogram');
saveas(h2_img, 'eqed_hist.png');
```

N)

Compute_hist.m

```
function h = compute_hist( img )
%compute histogram of an image

[m, n, c] = size(img);
h = zeros([256, c]);
for i = 1 : 256
    h(i, :) = sum(sum(img == (i - 1)));
end
end
```

Histogram_matching.m

```
function [ K ] = HistogramMatching( I, target )
I_h = compute_hist(I);
total_I = sum(I_h);
target_h = compute_hist(target);
total_target = sum(target_h);
[m, n] = size(I_h);
% define pdf for image I
pdf_h = zeros([m, n]);
pdf_h(1, :) = I_h(1, :) ./ total_I;
for i = 2 : m
   pdf_h(i, :) = pdf_h(i - 1, :) + I_h(i, :) ./ total_I;
end
% define pdf for image target
pdf_target = zeros([m, n]);
pdf_target(1, :) = target_h(1, :) ./ total_I;
for i = 2 : m
   pdf_target(i, :) = pdf_target(i - 1, :) + target_h(i, :) ./ total_target;
end
%define LUT for image transform with target
LUT = zeros([m, n]);
for i = 1 : 256
   for k = 1 : 3
      temp = pdf_h(i, k);
      for j = 1 : 256
         if pdf_target(j, k) > temp
            LUT(i, k) = j;
            break;
         end
      end
   end
end                              % get new image close to target image
K = zeros(size(I));
[r, g, b] = size(K);
for i = 1 : r
   for j = 1 : g
      for c = 1 : b
         K(i, j, c) = LUT(I(i, j, c) + 1, c);
      end
   end
end
K = uint8(K);
end
```

Test_match.m

```
I = imread('AtlasMercury.tif');
h_I = compute_hist(I);
target = imread('target.jpg');
h_target = compute_hist(target);
K = HistogramMatching(I, target);
h_K = compute_hist(K);

x = 0 : 255;
figure;
image(I);
title('origin image');

h1_img = figure;
bar(x, h_I(:, 1));
title('origin image histogram');
saveas(h1_img, 'original_hist.png');

figure;
image(target);
title('target image');

h2_img =figure;
bar(x, h_target(:, 1));
title('target image histogram');
saveas(h2_img, 'target_hist.png');

figure;
image(K);
title('transform image');
imwrite(K, 'Transformed_img.png');

h3_img =figure;
bar(x, h_K(:, 1));
title('transform image histogram');
saveas(h3_img, 'transformed_hist.png');
```