COMP 6771 Image Processing
Assignment 1
Submitted by: Rajat Sharma (40196467)

1. **(6 point)** A greyscale image A has a normalized histogram p_r (modeled by a probability density function) that can be presented by

$$p_r(r) = \begin{cases} -2r + 2, & 0 \le r \le 1 \\ 0, & otherwise \end{cases}$$

Now we want to transform the grey levels of this image so that they will have the normalized histogram p_z, which is given by:

$$p_z(z) = \begin{cases} 2z, & 0 \le z \le 1 \\ 0, & otherwise \end{cases}$$

You can assume continuous quantities. Please find the transform between r and z that will accomplish that.

The transformation between the grayscale levels r and z that will accomplish the desired normalized histogram p" can be found by solving the following equation:

∫p!(r)dr = ∫p"(z)dz

Given the probability density functions for p!(r) and p"(z), the equation becomes:

∫(2 - 2r)dr = ∫(2z)dz

Now, let's solve this equation:

∫(2 - 2r)dr = ∫(2z)dz

Integrating the left side with respect to r:

2r - r^2 = ∫(2z)dz

Integrating the right side with respect to z:

2r - r^2 = z^2 + C

Where C is the constant of integration.

Now, we want to solve this equation for z in terms of r:

z^2 = 2r - r^2 - C

z = ±√(2r - r^2 - C)

Since we are dealing with normalized histograms, z should lie in the range [0, 1]. We can choose the positive square root for simplicity:

z = √(2r - r^2 - C)

Now, we need to determine the constant C to satisfy the conditions. Since we want p!(r) and p"(z) to be normalized histograms, we can set C such that z lies in the range [0, 1] for r in the range [0, 1].

When r = 0:

z = √(2(0) - (0)^2 - C) = √(-C)

To ensure that z is within [0, 1], we can set C = 0. Therefore, C = 0.

So, the transformation function between r and z that accomplishes the desired normalized histogram p" is:

z = √(2r - r^2)

First, apply the histogram equalization:

$$u = T(r) = \int_0^r p_r(k)dk = \int_0^r (-2k+2)dk = -r^2 + 2r$$

And for the second diagram:

$$v = H(z) = \int_0^z p_z(k)dk = \int_0^z 2kdk = z^2,$$

Let $u = v$, then

$$-r^2 + 2r = z^2$$

$$\Rightarrow z = \pm\sqrt{-r^2 + 2r}$$

Finally, because the intensity levels are all positive numbers, so

$$z = \sqrt{-r^2 + 2r}$$

This function will map the grayscale levels in image A to the desired histogram p".

2. (**20 point**) The purpose of this question is to perform histogram equalization to a given histogram and plot the result. Given the following histogram with their gray levels, and the corresponding number of pixels:

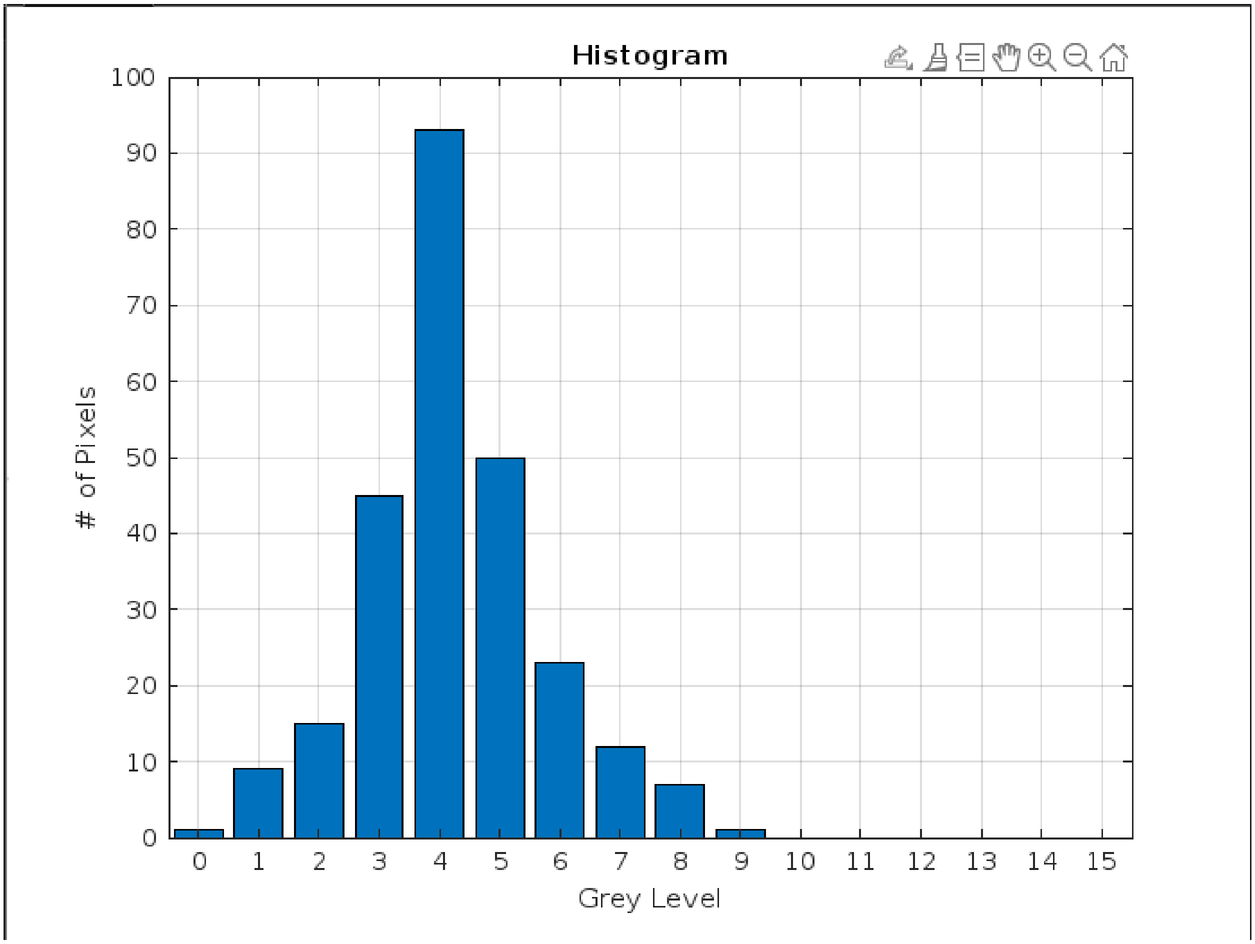
Grey level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of pixels	1	9	15	45	93	50	23	12	7	1	0	0	0	0	0	0

a) (**5 point**) Plot the histogram of the image given in the table above.

b) (**10 point**) Let r_k be the grey level given in the table, perform histogram equalization by:
i) Calculate the grey level s_k in the image after histogram equalization from the table.
ii) Plot the probability density functions $p_r(r_k)$ and $p_s(s_k)$.

c) (**5 point**) Plot the new histogram after performing the histogram equalization.

a)



b) (i)

Grey Level	Number of pixels	$P_r(r_k) = n_k/MN$	$S_k = (L-1)*\Sigma P_r(r_k)$	Ns	$P_s(s_k)$
0	1	1/256 = 0.004	15*0.004 = 0.06 ≈ 0	1	0.004
1	9	9/256 = 0.035	15*(0.004+0.035)= 0.585 ≈ 1	24	0.094
2	15	25/256 = 0.097	15*(0.004+0.035+0.097)= 1.464 ≈ 1	0	0
3	45	45/256 = 0.18	15*(0.004+0.035+0.097+0.18)= 4.10 ≈ 4	45	0.18

4	93	93/256 = 0.36	15*(0.004+0.035+0.097+0.18+0.36)=9.55 ≈ 10	93	0.36
5	50	50/256 = 0.20	15*(0.004+0.035+0.097+0.18+0.36+0.20)=12.485 ≈ 12	50	0.20
6	23	23/256 = 0.09	15*(0.004+0.035+0.097+…) = 13.828 ≈ 14	23	0.09
7	12	12/256 = 0.047	15*(0.004+0.035+0.097+…) = 14.53 ≈ 15	20	0.08
8	7	7/256 = 0.027	15*(0.004+0.035+0.097+…)= 14.94 ≈ 15	0	0
9	1	2/256 = 0.008	15*(0.004+0.035+0.097+…)=15.00≈ 15	0	0
10	0	0	15	0	0
11	0	0	15	0	0
12	0	0	15	0	0
13	0	0	15	0	0
14	0	0	15	0	0
15	0	0	15	0	0

M*N (total number of pixels) = 256

L(grey levels) = 16 **=>** L-1=15

r0=0 is mapped to s0 **=>** $n_{s0} = n_{r0} = 1$

$P(s_0) = 1/256 = 0.004$

r1=1 and r2=2 is mapped to s1=s2=1 **=>** $n_{s1} = n_{s2} = n_{r1} + n_{r2} = 24$

$P(s_1) = 24/256 = 0.094$

r3=3 is mapped to s3=4 **=>** $n_{s3} = n_{r3} = 45$
$P(s_3) = 45/256 = 0.18$

r4=4 is mapped to s4=10 **=>** $n_{s4} = n_{r4} = 93$

$P(s_4) = 93/256 = 0.36$

$r5=5$ is mapped to $s5= 12 \Rightarrow n_{s5}=n_{r5}=50$

$P(s_5)= 50/256=0.20$

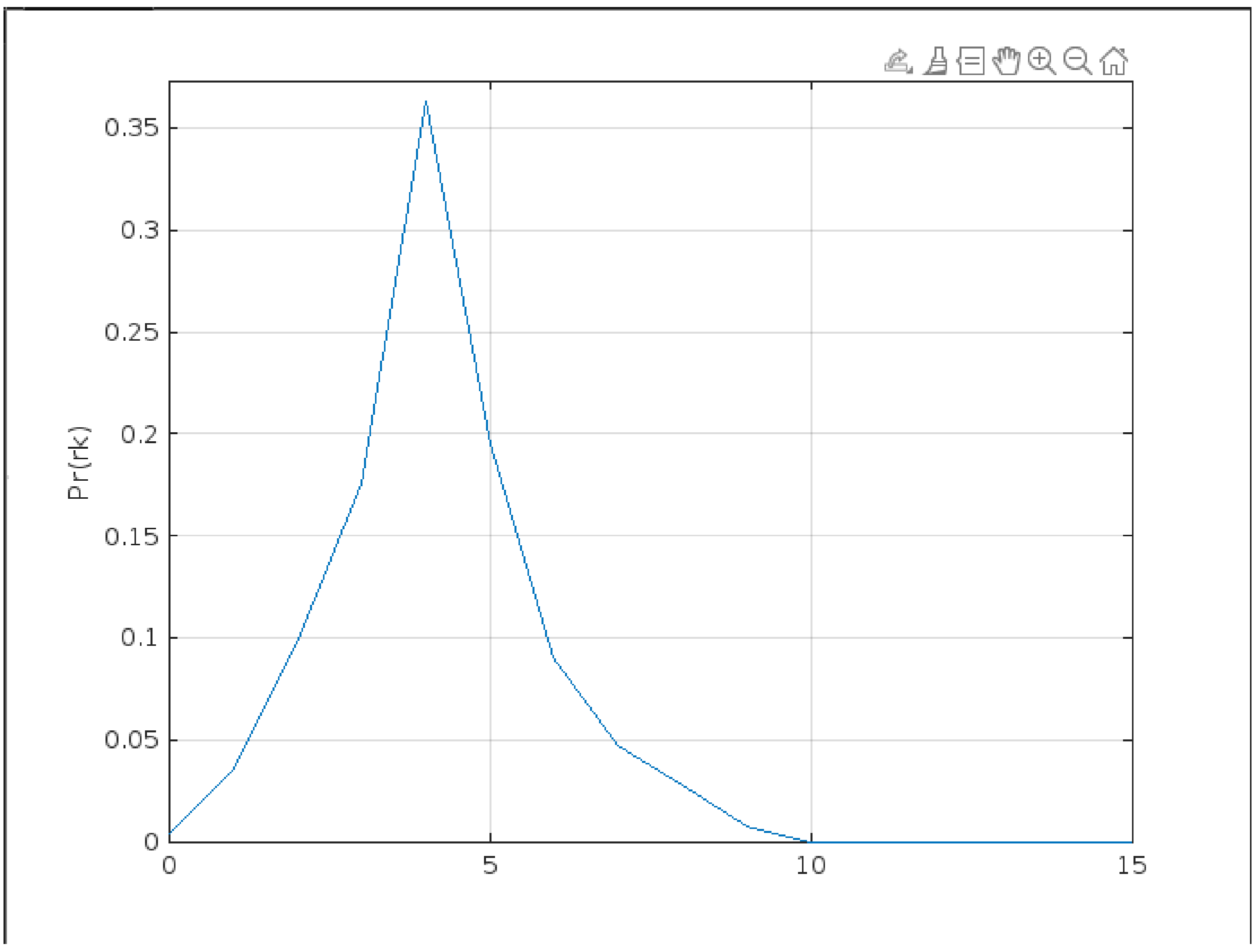
$r6=6$ is mapped to $s6=14 \Rightarrow n_{s6}=n_{r6}=23$

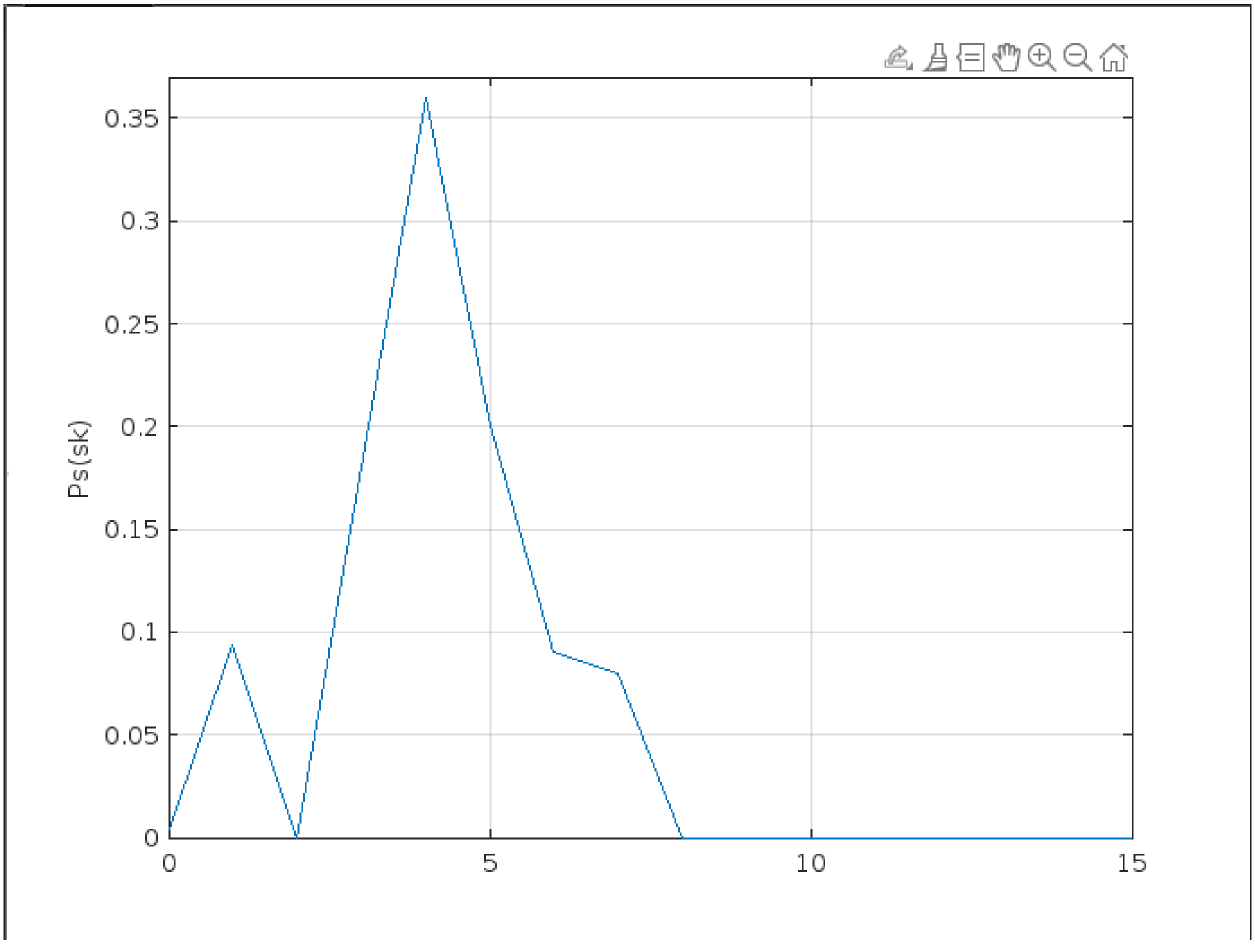
$P(s_6)= 23/256=0.09$

$r7,r8,r9,r10,..,r15$ all mapped to $s7=s8=s9=s10=\ldots=s15=15 \Rightarrow$

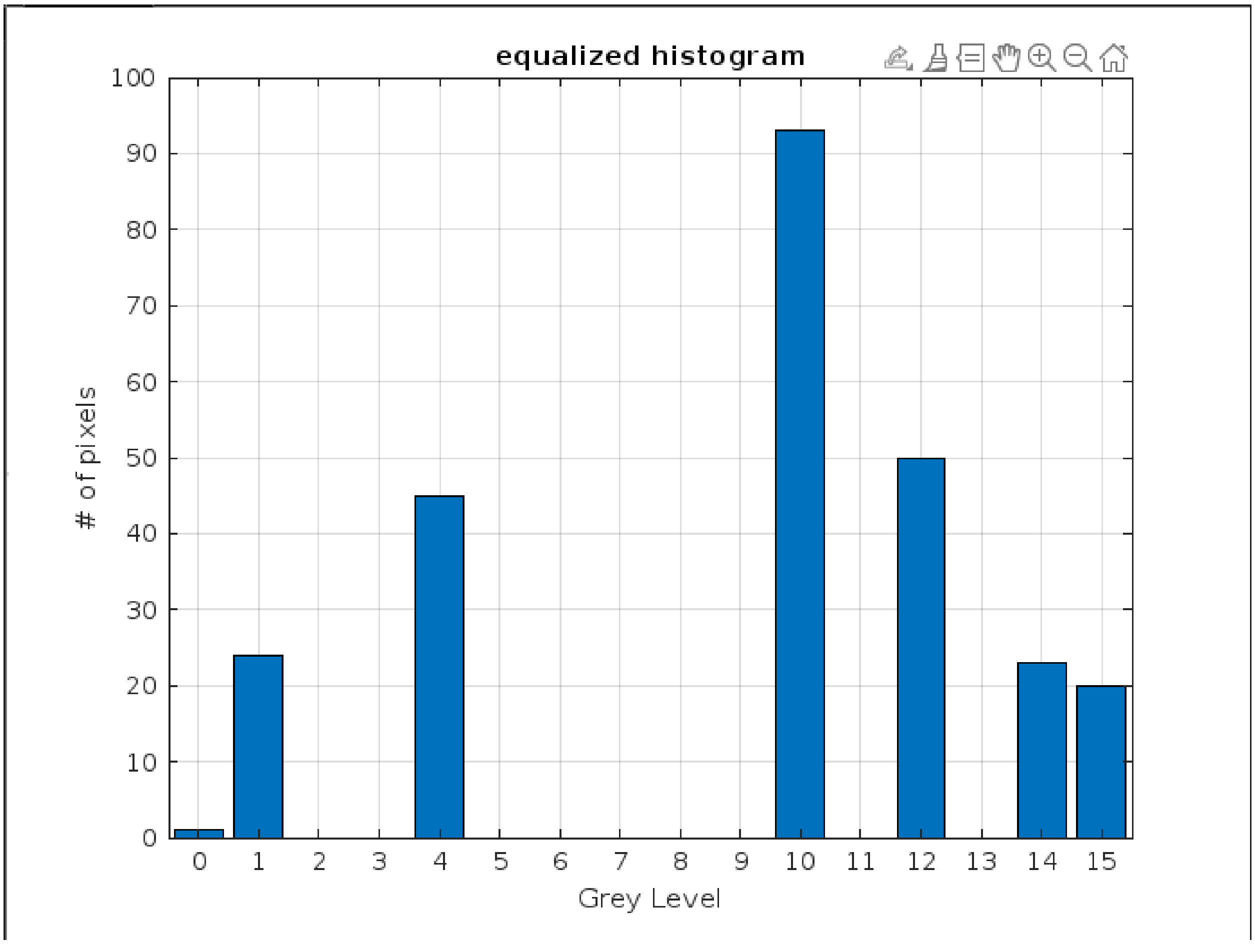
$n_{s7}=n_{s8}=\ldots=ns15=n_{r7}+n_{r8}+\ldots+n_{r15}$ $P(s_7)=20/256= 0.08$

(b) (ii)





(c)



3. (4 point) Given two images $f(x, y)$ and $g(x, y)$ with their histograms hf and hg . Assuming that all the pixels of $g(x, y)$ have the same constant intensity value of c (c>0) and both images

have positive values. Let uk be the intensity levels of the pixels formed by an element-wise arithmetic operation applied to the two images: $f(x, y) - g(x, y)$. Explain the relationship between the histograms of $f(x, y) - g(x, y)$ and $f(x, y)$.

When you perform an element-wise arithmetic operation of subtracting g(x, y) from f(x, y), where g(x, y) is a constant value 'c' (c > 0), several things happen to the histogram of the resulting image, f(x, y) − g(x, y):

Shift in Histogram:

The histogram of f(x, y) − g(x, y) will be shifted to the left by a constant value of 'c'. This is because, for each pixel in f(x, y), 'c' is subtracted from its intensity value. As a result, all the intensity values in the histogram of f(x, y) − g(x, y) will be reduced by 'c' units compared to the histogram of f(x, y). This shift implies that the intensity values of the pixels in the resulting image will be lower than the corresponding pixels in the original f(x, y).

Preservation of Shape:

The shape of the histogram of f(x, y) − g(x, y) will remain similar to the shape of the histogram of f(x, y). This is because the relative differences in intensity values within f(x, y) are maintained during the subtraction operation. Only the absolute intensity values are affected by the subtraction of 'c'. Therefore, the relationships and proportions between different intensity levels within the image are preserved.

Brightness Change:

Subtracting a constant value 'c' from every pixel in f(x, y) will result in an overall change in brightness. If 'c' is positive, the image will become darker because all pixel intensities are reduced by 'c'. If 'c' is negative, the image will become brighter because all pixel intensities are increased by the magnitude of 'c'.

In conclusion, the histogram of f(x, y) − g(x, y) is a shifted version of the histogram of f(x, y) with a shift of 'c' units to the left. The shape of the histogram remains similar, but the overall brightness of the resulting image changes based on the sign and magnitude of 'c'.

4. **(6 point)** Find all the bit planes of the following 4-bit image:

0	1	8	6
2	2	1	1
1	15	14	12
3	6	9	10

Answer)

To find the bit plane 4 for the given numbers and create four matrices based on the significant bits, you'll need to follow these steps:

Step 1: Convert the numbers to binary and pad them to make sure they all have the same number of bits (4 bits in this case).

The given numbers:

```
0 1 8 6

2 2 1 1

1 15 14 12

3 6 9 10
```

Binary representation (padded to 4 bits):

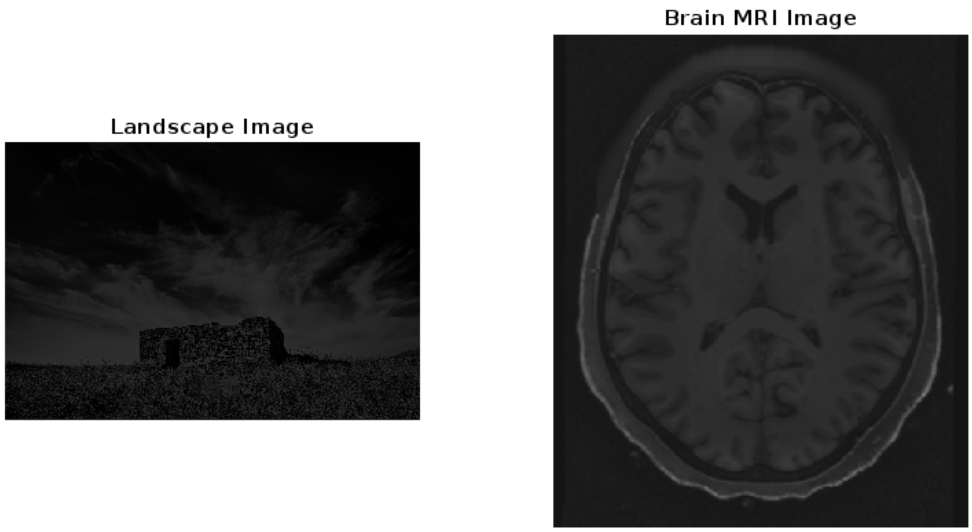
```
0000 0001 1000 0110

0010 0010 0001 0001

0001 1111 1110 1100

0011 0110 1001 1010
```

Step 2: Create four matrices for each bit position (1st, 2nd, 3rd, and 4th significant bit) by extracting that bit from each binary number.

```
Bit plane 0 :

0 0 1 0

0 0 0 0

0 1 1 1

0 0 1 1

Bit plane 1 :

0 0 0 1

0 0 0 0

0 1 1 1

0 1 0 0

Bit plane 2 :

0 0 0 1

1 1 0 0

0 1 1 0

1 1 0 1

Bit plane 3 :

0 1 0 0

0 0 1 1

1 1 0 0

1 0 1 0
```

Part II: Programming question

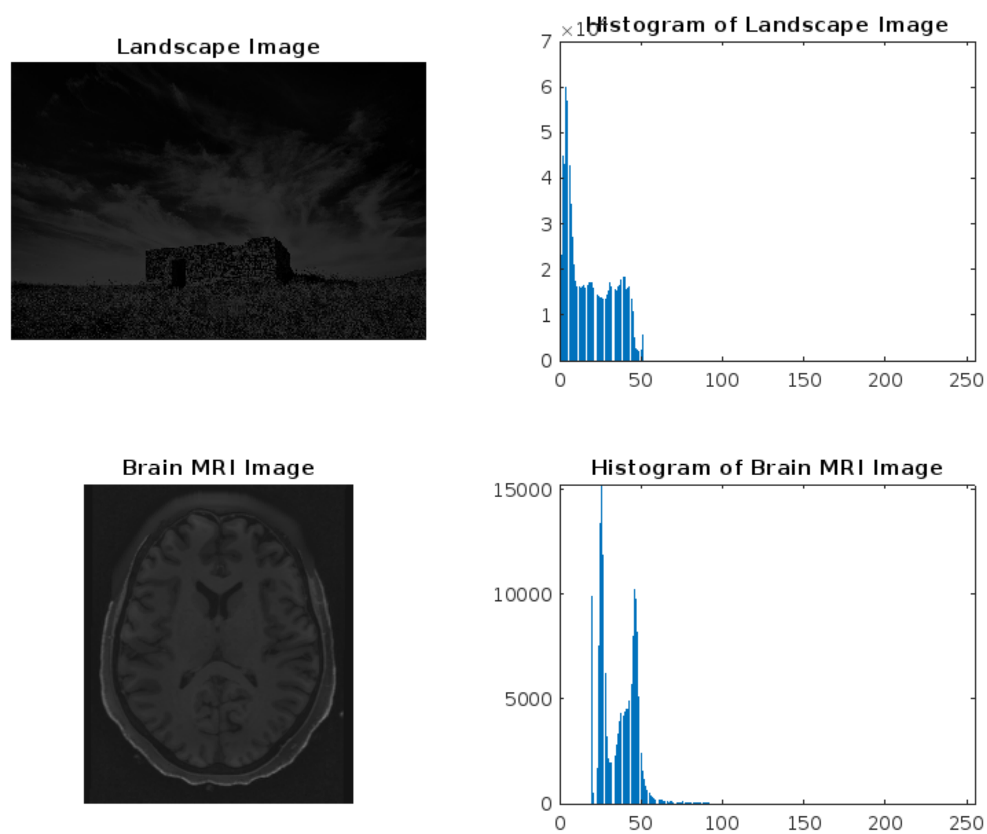
1) (1 point) Write a program to read the images.




```
%% 1

% Read the grayscale images

land_img = imread('landscape.png');

brain_img = imread('brain-mri.png');

% Display the landscape image

subplot(1, 2, 1);

imshow(land_img);
```

```
title('Landscape Image');

% Display the brain MRI image

subplot(1, 2, 2);

imshow(brain_img);

title('Brain MRI Image');
```

2) (3 point) Write a program to calculate the histograms of the images and display the histograms.

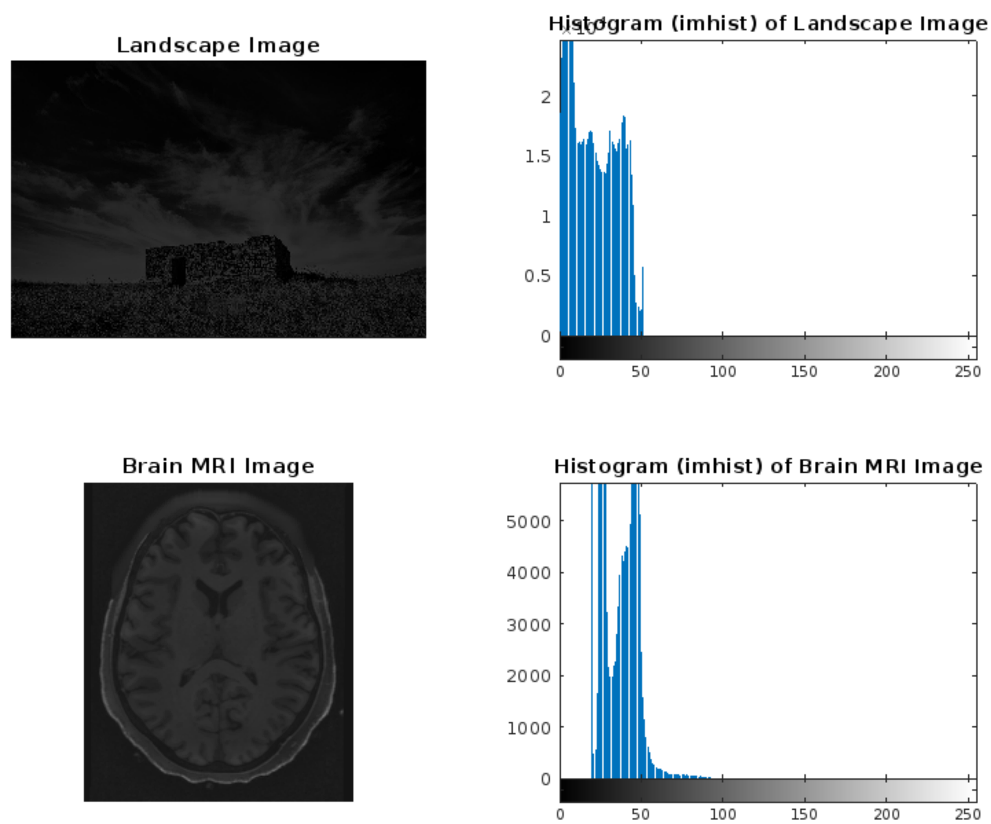



```
%% 2

% Calculate the histogram of the landscape image
```

```
[height_land, width_land] = size(land_img);
hist_data_land = zeros(1, 256);

% Loop to calculate the occurrence numbers of all gray levels
for landscape image
for rr = 1:height_land
    for cc = 1:width_land
        hist_data_land(land_img(rr, cc) + 1) = 1 +
hist_data_land(land_img(rr, cc) + 1);
    end
end

% Calculate the histogram of the brain MRI image
[height_brain, width_brain] = size(brain_img);
hist_data_brain = zeros(1, 256);

% Loop to calculate the occurrence numbers of all gray levels
for brain MRI image
for rr = 1:height_brain
    for cc = 1:width_brain
        hist_data_brain(brain_img(rr, cc) + 1) = 1 +
hist_data_brain(brain_img(rr, cc) + 1);
    end
end
```

```
% Display the histograms
figure;
subplot(2, 2, 1);
imshow(land_img);
title('Landscape Image');

subplot(2, 2, 2);
bar([0:255], hist_data_land);
set(gca, 'xtick', [0:50:255]);
set(gca, 'XLim', [0 255]);
title('Histogram of Landscape Image');

subplot(2, 2, 3);
imshow(brain_img);
title('Brain MRI Image');

subplot(2, 2, 4);
bar([0:255], hist_data_brain);
set(gca, 'xtick', [0:50:255]);
set(gca, 'XLim', [0 255]);
title('Histogram of Brain MRI Image');
```

3) (2 point) Compare the calculated histograms obtained by using your own program with the ones using the imhist function in MATLAB (or an equivalent function in the software library of your choice).



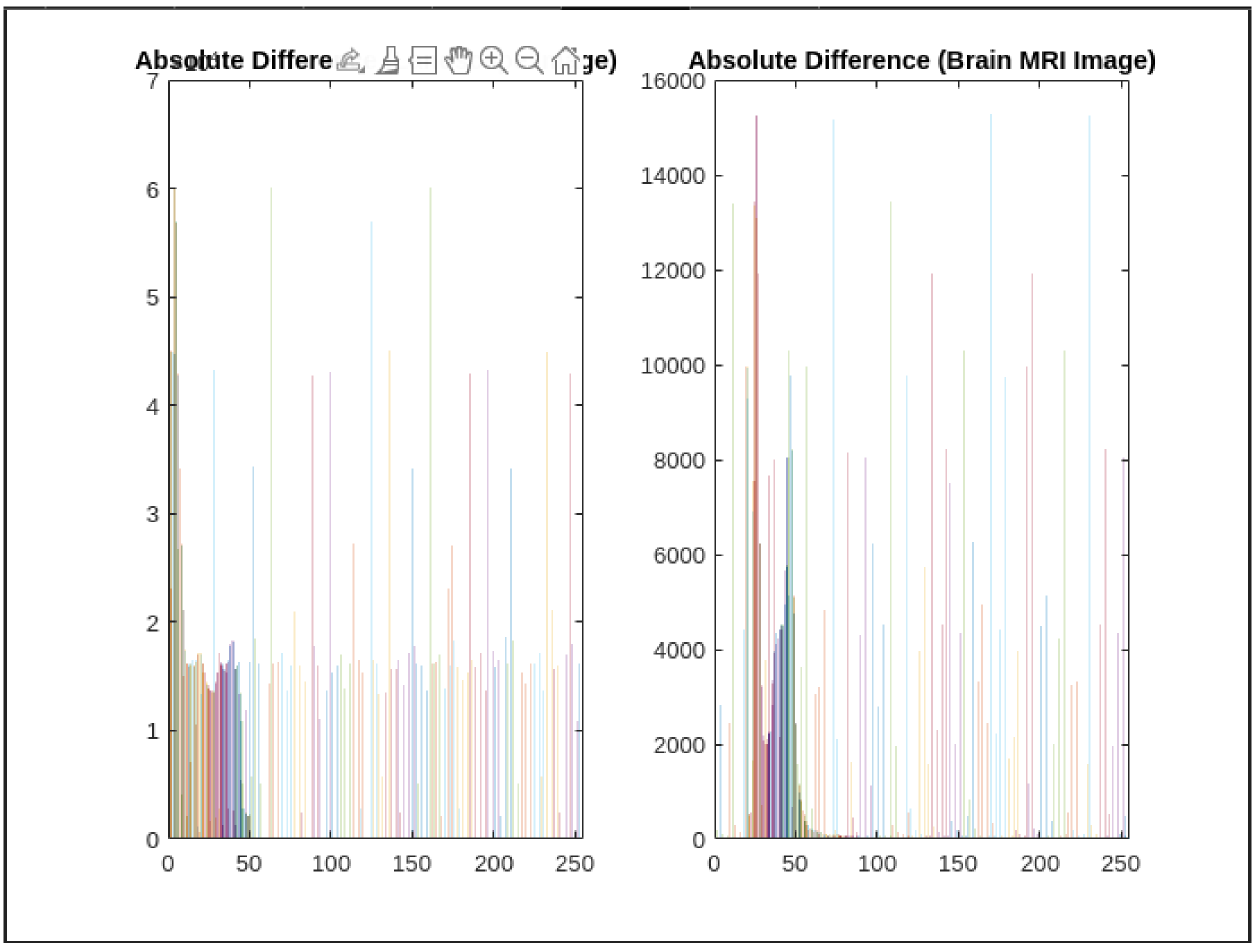

```
%% 3

% Calculate the histogram using your own program for both
images
```

```
[height_land, width_land] = size(land_img);

hist_data_land = zeros(1, 256);


for rr = 1:height_land

    for cc = 1:width_land

        hist_data_land(land_img(rr, cc) + 1) = 1 +
hist_data_land(land_img(rr, cc) + 1);

    end

end


[height_brain, width_brain] = size(brain_img);

hist_data_brain = zeros(1, 256);


for rr = 1:height_brain

    for cc = 1:width_brain

        hist_data_brain(brain_img(rr, cc) + 1) = 1 +
hist_data_brain(brain_img(rr, cc) + 1);

    end

end


% Calculate the histogram using imhist function for both
images

imhist_land = imhist(land_img);

imhist_brain = imhist(brain_img);
```

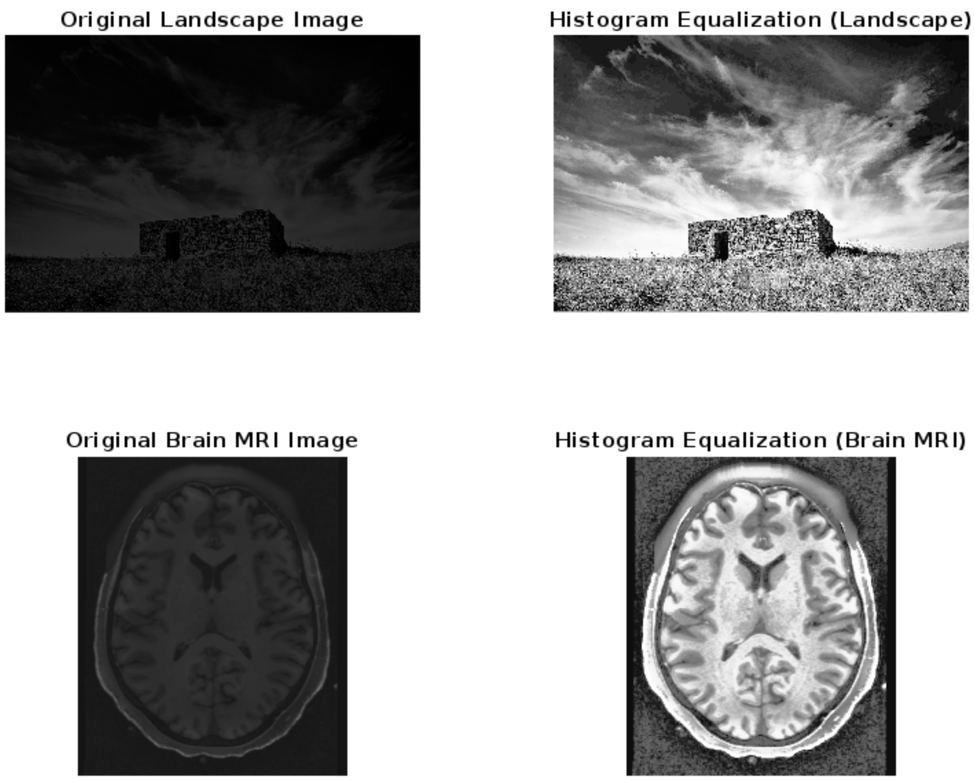
```
% Compare the histograms visually or using metrics
figure;
subplot(2, 2, 1);
imshow(land_img);
title('Landscape Image');

subplot(2, 2, 2);
bar([0:255], hist_data_land);
set(gca, 'xtick', [0:50:255]);
set(gca, 'XLim', [0 255]);
title('Histogram (Custom) of Landscape Image');

subplot(2, 2, 3);
imshow(brain_img);
title('Brain MRI Image');

subplot(2, 2, 4);
bar([0:255], hist_data_brain);
set(gca, 'xtick', [0:50:255]);
set(gca, 'XLim', [0 255]);
title('Histogram (Custom) of Brain MRI Image');

figure;
```

```
subplot(2, 2, 1);
imshow(land_img);
title('Landscape Image');

subplot(2, 2, 2);
imhist(land_img);
title('Histogram (imhist) of Landscape Image');

subplot(2, 2, 3);
imshow(brain_img);
title('Brain MRI Image');

subplot(2, 2, 4);
imhist(brain_img);
title('Histogram (imhist) of Brain MRI Image');

% Display the absolute difference between histograms
figure;
subplot(1, 2, 1);
bar([0:255], abs(hist_data_land - imhist_land));
set(gca, 'xtick', [0:50:255]);
set(gca, 'XLim', [0 255]);
title('Absolute Difference (Landscape Image)');
```

```
subplot(1, 2, 2);

bar([0:255], abs(hist_data_brain - imhist_brain));

set(gca, 'xtick', [0:50:255]);

set(gca, 'XLim', [0 255]);

title('Absolute Difference (Brain MRI Image)');
```



4) (3 point) Write a program to do histogram equalization on these images.




```
%% 4

% Calculate the histogram equalization for the landscape image

prob_data_land = zeros(1, 256);

totalPixelNum_land = height_land * width_land * 1.0;

for ii = 1:256

prob_data_land(ii) = hist_data_land(ii) / totalPixelNum_land;

end

cumn_data_land = zeros(1, 256);
```

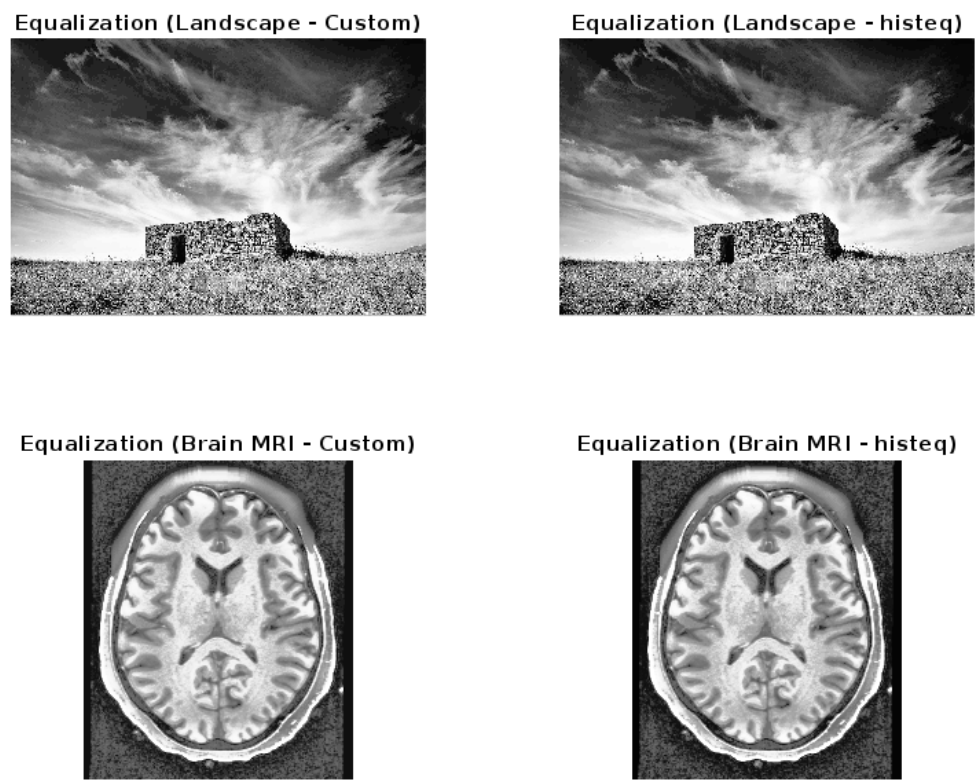
```
for ii = 1:256
if ii == 1
cumn_data_land(ii) = prob_data_land(ii);
else
cumn_data_land(ii) = cumn_data_land(ii - 1) + prob_data_land(ii);
end
end
cumn_data_land = uint8(255 .* cumn_data_land + 0.5);
hieq_img_land = zeros([height_land, width_land]);
for rr = 1:height_land
for cc = 1:width_land
hieq_img_land(rr, cc) = cumn_data_land(land_img(rr, cc) + 1);
end
end
hieq_img_land = uint8(hieq_img_land);
% Repeat the process for the brain MRI image
prob_data_brain = zeros(1, 256);
totalPixelNum_brain = height_brain * width_brain * 1.0;
for ii = 1:256
prob_data_brain(ii) = hist_data_brain(ii) / totalPixelNum_brain;
end
cumn_data_brain = zeros(1, 256);
for ii = 1:256
if ii == 1
cumn_data_brain(ii) = prob_data_brain(ii);
else
```

```
cumn_data_brain(ii) = cumn_data_brain(ii - 1) +
prob_data_brain(ii);

end

end

cumn_data_brain = uint8(255 .* cumn_data_brain + 0.5);

hieq_img_brain = zeros([height_brain, width_brain]);

for rr = 1:height_brain

for cc = 1:width_brain

hieq_img_brain(rr, cc) = cumn_data_brain(brain_img(rr, cc) + 1);

end

end

hieq_img_brain = uint8(hieq_img_brain);

% Display the original and histogram-equalized images

figure;

subplot(2, 2, 1);

imshow(land_img);

title('Original Landscape Image');

subplot(2, 2, 2);

imshow(hieq_img_land);

title('Histogram Equalization (Landscape)');

subplot(2, 2, 3);

imshow(brain_img);

title('Original Brain MRI Image');

subplot(2, 2, 4);

imshow(hieq_img_brain);

title('Histogram Equalization (Brain MRI)');
```

5) (1 point) Compare the histogram-equalized images obtained by using your own program with

the one by using histeq function in MATLAB (or an equivalent function in the software library

of your choice).




```
%% 5

% Perform histogram equalization using your own program for the landscape image

hieq_img_land = uint8(hieq_img_land);
```

```
% Perform histogram equalization using histeq function for the
landscape image

histequ_img_land = histeq(land_img);

% Perform histogram equalization using your own program for the
brain MRI image

hieq_img_brain = uint8(hieq_img_brain);

% Perform histogram equalization using histeq function for the
brain MRI image

histequ_img_brain = histeq(brain_img);

% Display the histogram-equalized images for both landscape and
brain MRI

figure;

subplot(2, 2, 1);

imshow(hieq_img_land);

title('Equalization (Landscape - Custom)');

subplot(2, 2, 2);

imshow(histequ_img_land);

title('Equalization (Landscape - histeq)');

subplot(2, 2, 3);

imshow(hieq_img_brain);

title('Equalization (Brain MRI - Custom)');

subplot(2, 2, 4);

imshow(histequ_img_brain);

title('Equalization (Brain MRI - histeq)');
```