Department of Computer Science and Software Engineering
Comp 6771 Image Processing
Assignment 1

Mandana Samiei
ID: 40059116

Theoretical Questions:

1-

$$H[af(x,y)+bg(x,y)] = aH[f(x,y)] + bH[g(x,y)]$$

Given two arbitrary images f(x,y) , g(x,y):
two arbitrary constants: a,b
answer: No, the operator that computes the median of a set of pixels of a sub-image is not Linear. I am going to prove this fact through an example:
Consider two 3*3 images:

1- f(x,y)

3	3	4
4	2	5
5	3	6

2,3,3,3,4,4,5,5,6 → m1= 4

2- g(x,y)

7	5	4
7	10	5
8	9	6

4,5,5,6,7,7,8,9,10 → m2=7

In the next step we want to multiply each image with a constant such as a and b:
As an example: a=2, b=3

af(x,y):

6	6	8
8	4	10
10	6	12

bg(x,y):

21	15	12
21	30	15
24	27	18

If we add these two pictures to each other:

af(x,y)+bg(x,y)

6	6	8
8	4	10
10	6	12
21	15	12
21	30	15
24	27	18

4,6,6,6,8,8,10,10,12,12,15,15,18,21,21,24,27,30
→ M= (12+12)/2= 12

According the above-mentioned example, result of the first side of equation would be: M = 12 and of the other side of equation would be: a*m1 + b*m2 = 2*4 + 3*7 = 8+21 = 29

result: $12 \neq 29$

So we prove that, this equation doesn't support median operator by using a counterexample.

2-

Grey Level	Number of pixels	$P_r(r_k)= n_k/MN$	$S_k= (L-1)*\Sigma P_r(r_k)$	Ns	$P_s(s_k)$	Zs (grey levels after second equalization)
0	0	0/256 = 0	0	5	0.02	15*0.02=0.3 ≈ 0
1	5	5/256 = 0.02	15*0.02= 0.3 ≈ 0	0	0	0
2	13	13/256 = 0.05	15*(0.02+0.05)= 1.05 ≈ 1	13	0.05	15*0.07=1.05≈ 1
3	57	57/256 = 0.22	15*(0.22+0.05+0.02)= 4.35 ≈ 4	57	0.22	15*(0.22+0.05+0.02)= 4.35≈ 4
4	100	100/256 = 0.39	15*(0.39+0.22+0.05+0.02)=10.2 ≈ 10	100	0.39	15*(0.39+0.22+0.05+0.02)=10.2≈10
5	39	39/256 = 0.15	15*(0.15+0.39+0.22+0.05+0.02)=12.45 ≈ 12	39	0.15	15*(0.15+0.39+0.22+0.05+0.02)= 12.45≈12
6	21	21/256 = 0.08	15*(0.08+ 0.15+0.39+…) = 13.65 ≈ 14	33	0.13	15*(0.13+0.15+0.39+0.22+0.05+0.02) =14.4≈14
7	12	12/256 = 0.05	15*(0.05+0.08+0.15+…) = 14.4 ≈ 14	0	0	14
8	7	7/256 = 0.03	15*(0.03+0.05+0.08+…)= 14.85 ≈ 15	9	0.04	15
9	2	2/256 = 0.01	15*(0.01+0.03+0.05+…)= 15	0	0	15
10	0	0	15	0	0	15
11	0	0	15	0	0	15
12	0	0	15	0	0	15
13	0	0	15	0	0	15
14	0	0	15	0	0	15
15	0	0	15	0	0	15

M*N (total number of pixels) = 256
L(grey levels) = 16 → L-1=15

r0=0 and r1= 1 both mapped to s0=s1=0 → $n_{s0} = n_{S1} = n_{r0}+n_{r1}= 5$ $P(s_0)= 5/256= 0.02$

r2=2 is mapped to s2=1 → $n_{s2}= n_{r2}= 13$ $P(s_2)=13/256=0.05$

r3=3 is mapped to s3=4 → $n_{s3}=n_{r3}=57$ $P(s_3)=57/256=0.22$

r4=4 is mapped to s4=10 → $n_{s4}=n_{r4}=100$ $P(s_4)=100/256=0.39$

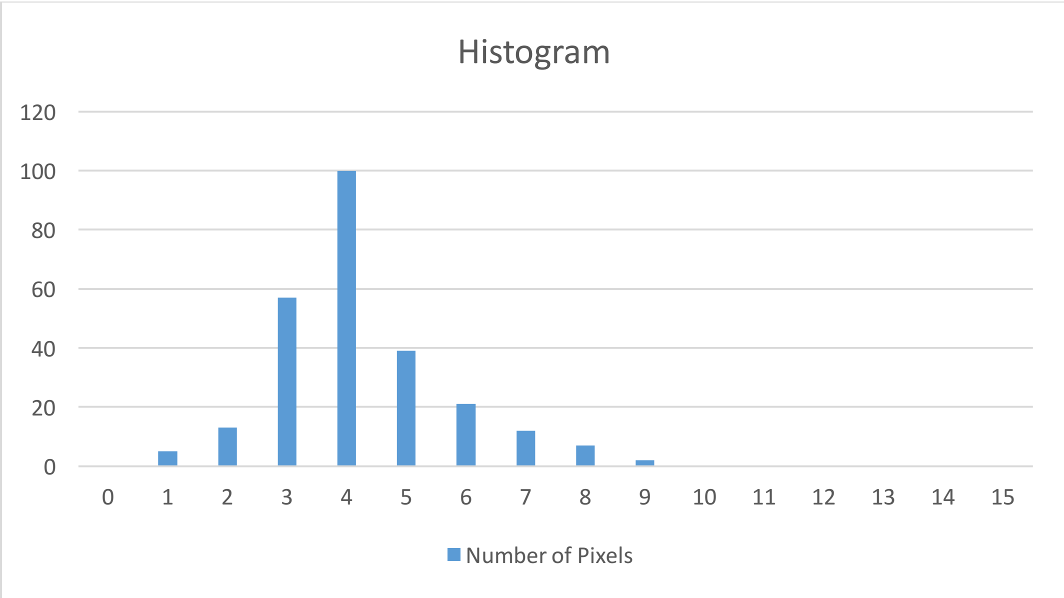
r5=5 is mapped to s5= 12 → $n_{s5}=n_{r5}=39$ $P(s_5)= 39/256=0.15$

r6=6 and r7=7 are mapped to s6= s7= 14→ $n_{s6}=n_{s7}=n_{r6}+n_{r7}$ $P(s_6)= 33/256=0.13$

r8,r9,r10,..,r15 all mapped to s8=s9=s10=…=s15=15 → $n_{s8}=\ldots=ns15=n_{r8}+\ldots+n_{r15}$

$P(s_8)=9/256= 0.04$

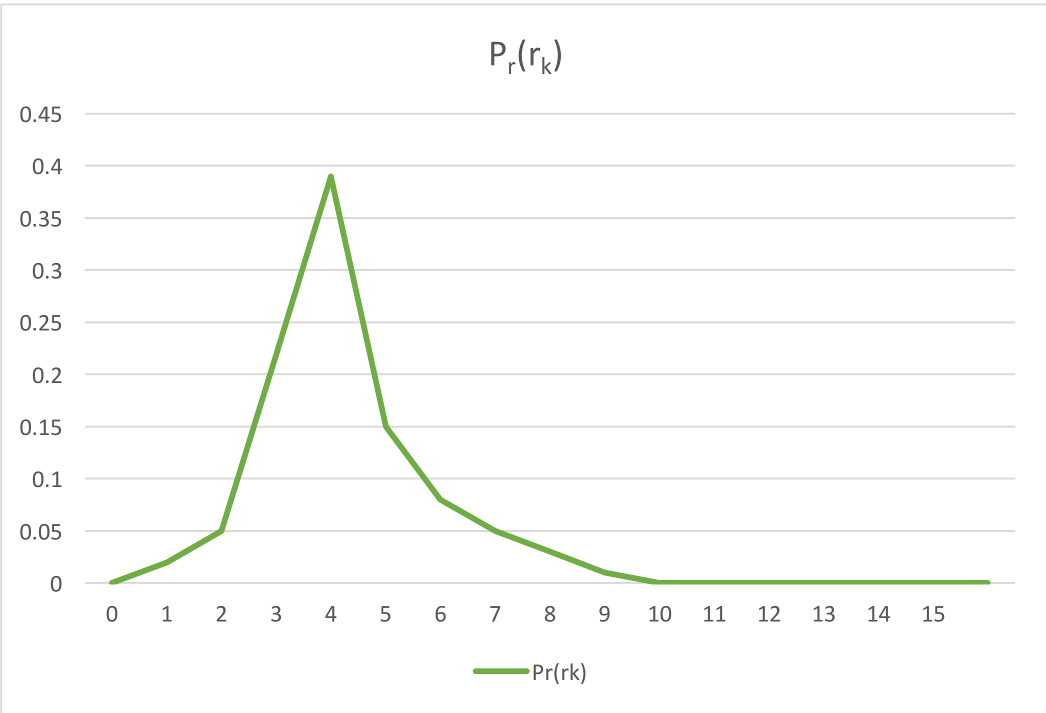
a)



Histogram
120
100
80
60
40
20
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Number of Pixels

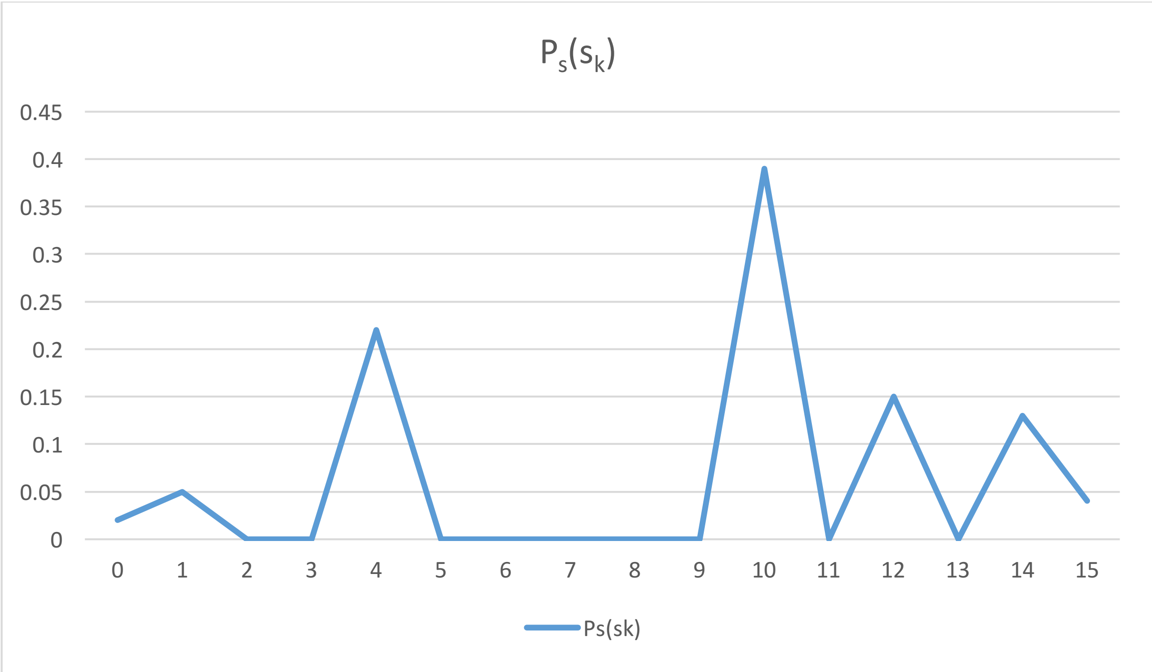
b)

i-

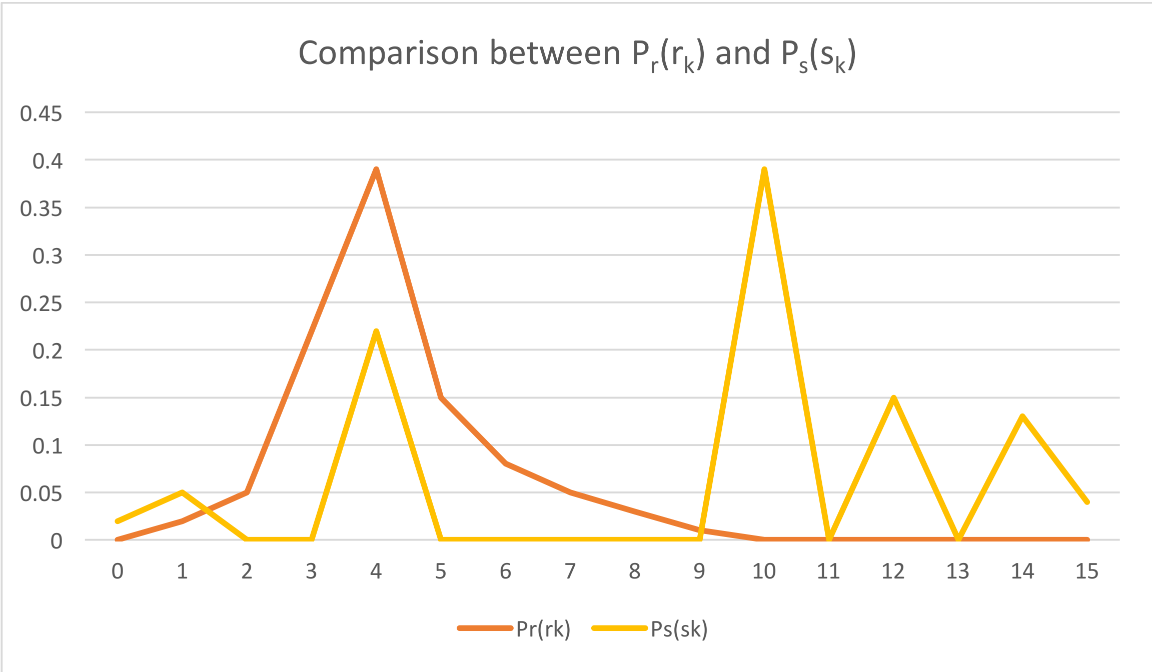


$P_r(r_k)$
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Pr(rk)

ii-

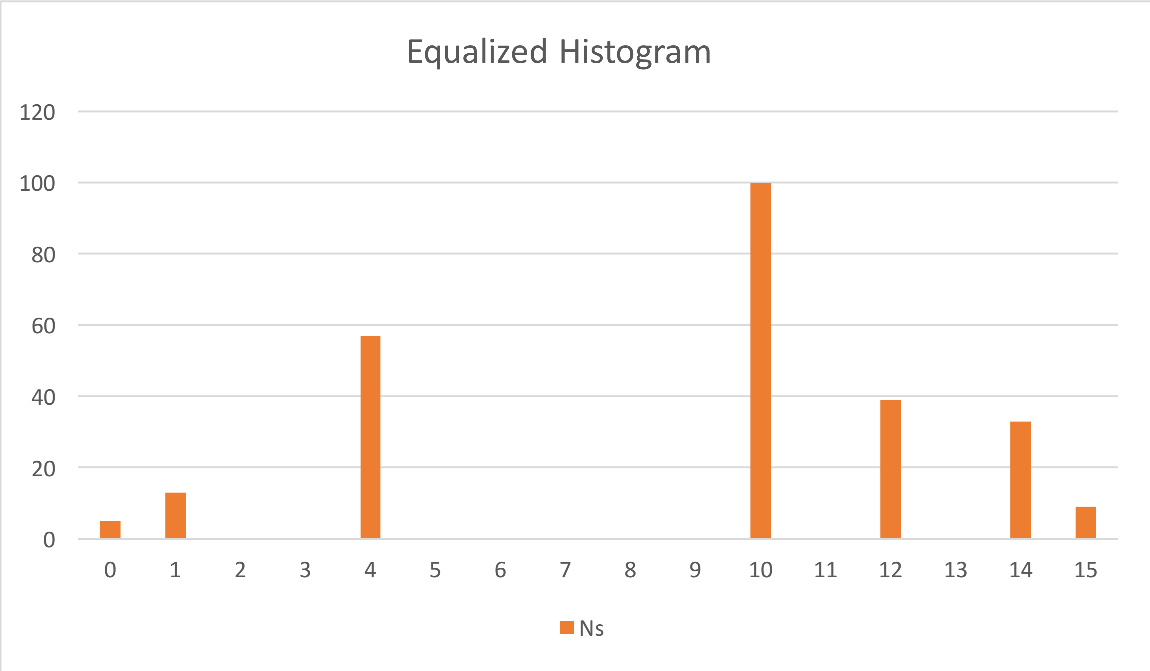


$P_s(s_k)$
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Ps(sk)



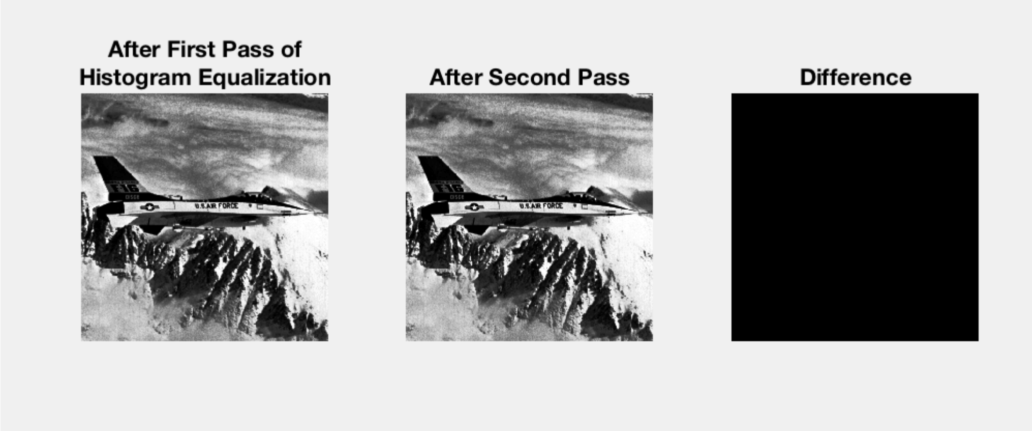
Comparison between $P_r(r_k)$ and $P_s(s_k)$
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Pr(rk)
Ps(sk)

c)



d) Histogram Equalization generally creates some fractures in the histogram, since this approach groups the greyscale components without regarding the number of pixels in each grey level. If we want to obtain a uniform histogram, it needs that greyscales be redistributed so that there are equal number of groups with equal number of pixels. The histogram equalization has no provisions for this type of redistribution process. Histogram equalization doesn't force the distribution "flat" that means the number of pixel in each grey levels distributed equally or closely.

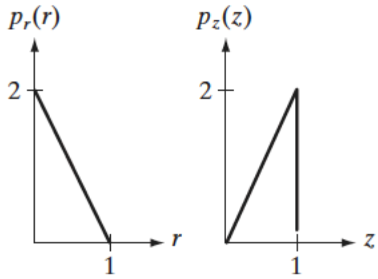
e)
The second pass of histogram equalization will produce the same result as the first pass. I calculated the second equalization in the above table, we can see that the result won't be changed. I also checked this problem in Matlab and I saw the same result:




```
I = imread('jetplaneCor.tiff');

H1=histeq(I);
figure;
subplot(1,3,1);
imshow(H1);title({'After First Pass of','Histogram Equalization'});

H2=histeq(H1);
subplot(1,3,2);
imshow(H2);title('After Second Pass');

%Compare
subplot(1,3,3);
imshow(abs(H1-H2));title('Difference');
```

3-

3.11 An image with intensities in the range [0, 1] has the PDF $p_r(r)$ shown in the following diagram. It is desired to transform the intensity levels of this image so that they will have the specified $p_z(z)$ shown. Assume continuous quantities and find the transformation (in terms of r and z) that will accomplish this.



from $P_r(r)$ diagram:
(0,2) (1,0) → $P_r(r) = -2(r-1)$

from $P_z(z)$ diagram:
(0,0) (1,2) → $P_z(z) = 2z$

First we calculate histogram equalization for $P_r(r)$:

S= T(r) = (L-1) $\int_0^r \mathrm{Pr}\,(w)dw = \int_0^r -2\,(w-1)dw = \int_0^r(-2\mathrm{w}+2)dw = -r^2+2r$

V=G(z) = (L-1) $\int_0^z \mathrm{Pz}\,(t)dt = \int_0^z (2\mathrm{t})dt = z^2$

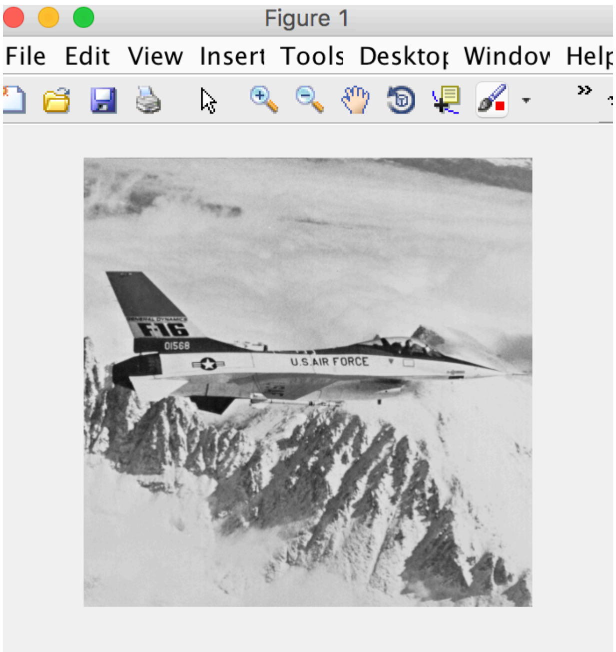
We want S = V = $G^{-1}(z)$ → $z^2 = = -r^2+2r$ → z = $\pm\sqrt{-r^2+2r}$
But Only positive grey levels are allowed so: z = $\sqrt{-r^2+2r}$

Programming Questions:

1) Read the greyscales of the image:

```
%%Part 2 - Programming
%%----------Question 1-------------
%%--------- Mandana Samiei---------
%%--------- Student Id: 40059116 --
img = imread('jetplaneCor.tiff');
figure;
imshow(img);
```

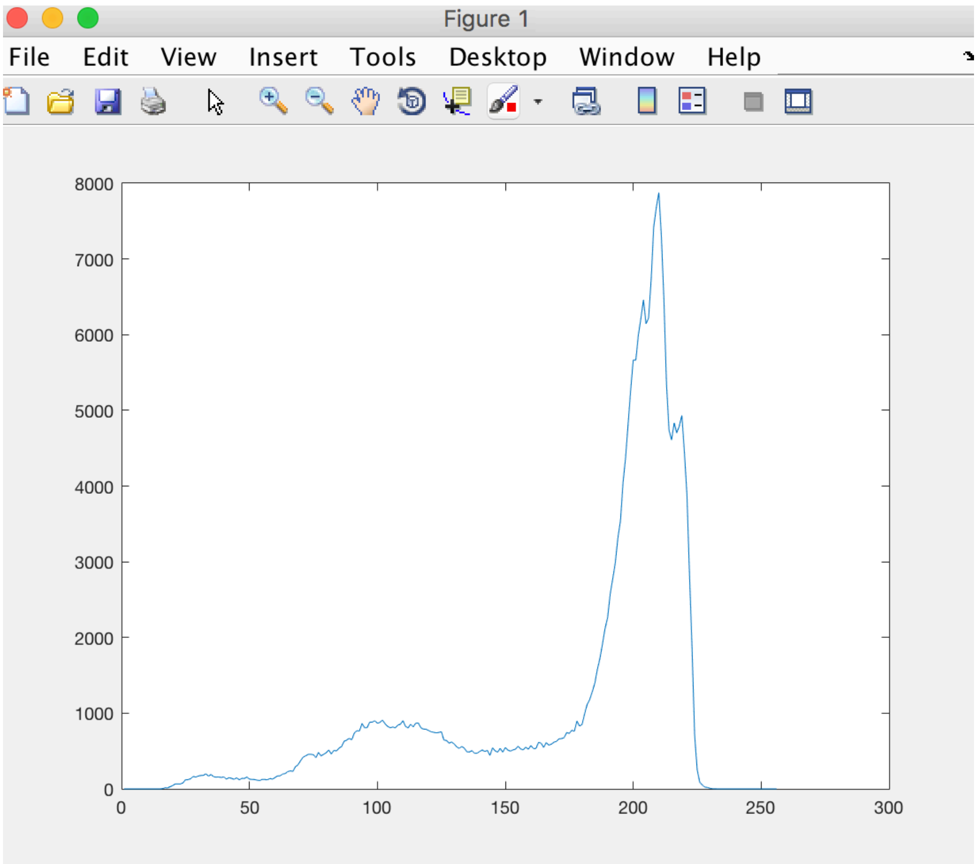
result:



2) Calculate the histogram of the image and display the histogram chart:

```
%%----------Question 2-------------
%%--------- Mandana Samiei---------
%%--------- Student Id:40059116 ---
img = imread('jetplaneCor.tiff');
H = zeros(256,1);

for i = 1: length(img)
    for j = 1: length(img)
        H(img(i,j)+1,1) =  H(img(i,j)+1,1)+1;
    end
end
figure;
plot(H);
```

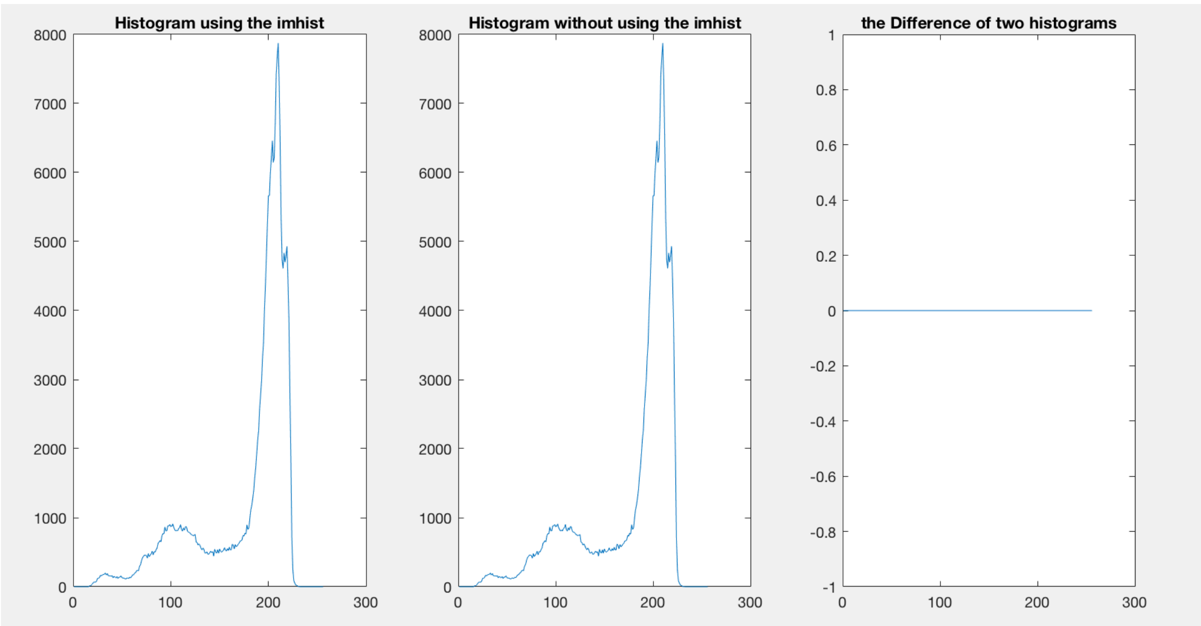
result:



3) Compare the calculate histogram obtained by using our own program and imhist function.

```
%%----------Question 3--------------
%%--------- Mandana Samiei----------
%%--------- Student Id: 40059116 ---
%%Using imhist function
img = imread('jetplaneCor.tiff');
H1 = imhist(img);
figure;
subplot(1,3,1);
plot(H1);

%%my approach
H2 = zeros(256,1);
for i = 1: length(img)
    for j = 1: length(img)
        H2(img(i,j)+1,1) =  H2(img(i,j)+1,1)+1;
    end
end
subplot(1,3,2);
plot(H2);

subplot(1,3,3);
plot(abs(H1-H2));
```

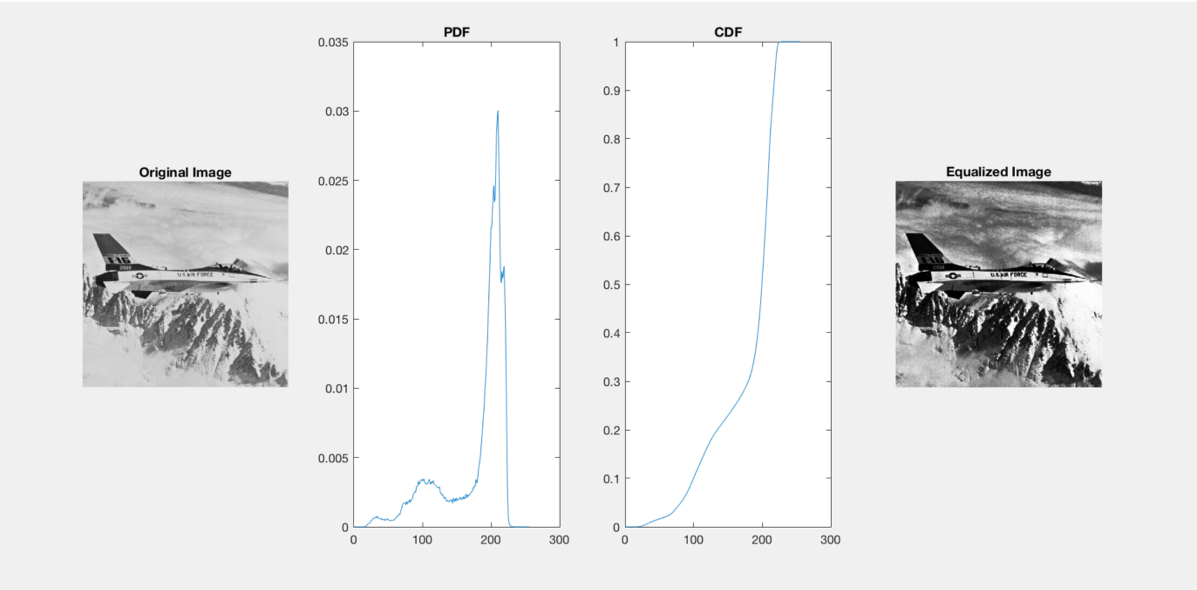
result:



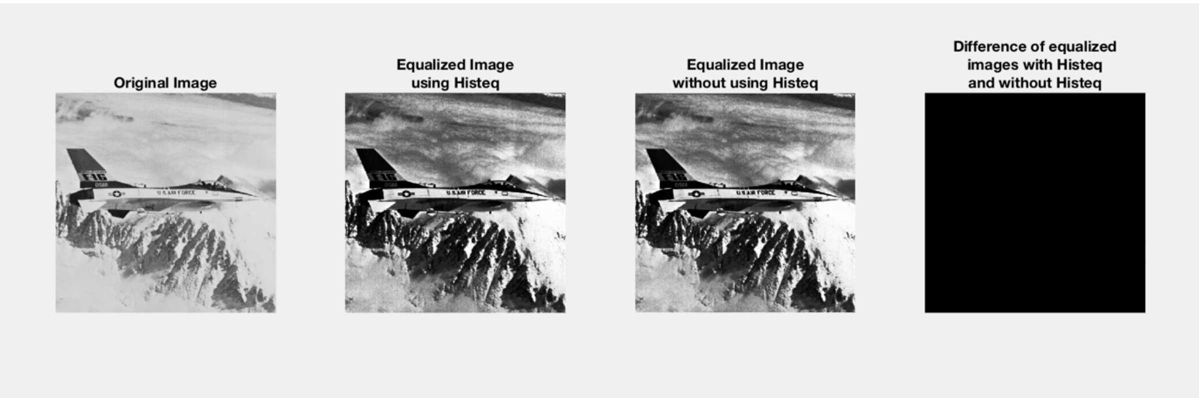
4) Write a program to do histogram equalization

```matlab
%%----------Question 4----------------
%%--------- Mandana Samiei-------------
%%--------- Student Id: 40059116 ------
%%----------Histogram Equalization-----

img = imread('jetplaneCor.tiff');
figure;
subplot(1,4,1);
imshow(img);
title('Original Image');

r = size(img,1);
c = size(img,2);
outimg = uint8(zeros(r,c));
N = r*c;
Frequency = zeros(256,1);
PDF = zeros(256,1);
CDF = zeros(256,1);

for i=1:r
    for j= 1:c
        greylevelValue = img(i,j);
        Frequency(greylevelValue+1) = Frequency(greylevelValue+1) + 1;
        PDF(greylevelValue+1)= Frequency(greylevelValue+1) / N;
    end
end
subplot(1,4,2);
plot(PDF);
title('PDF');
out = zeros(256,1);
L = 255; sum = 0;
cum = zeros(256,1);
for i = 1: size(PDF)
   sum = sum + Frequency(i);
   cum(i) = sum;
   CDF(i) = cum(i)/N;
   out(i) = round(CDF(i)*L);
end
subplot(1,4,3);
plot(CDF); title('CDF');

for i=1:r
    for j= 1:c
        outimg(i,j)= out(img(i,j)+1);
    end
end
subplot(1,4,4);
imshow(outimg);
title('Equalized Image');
```

result:



5) Compare the histogram equalized image obtained by using your own program with the one by using histeq function.



The histogram-equalized image obtained by using my own program with the one by using histeq function are not the same, by observing their matrix I found out their greyscale have a little difference, but generally they look same, I mean, human eye cannot detect their difference. And I think this difference is caused due to the different design of programs, probably histeq function in Matlab use some techniques to enhance the image effectively so, histeq function obtain a better result, but both of them look same.

6) Write a function to do image histogram stretching:

```
function  imgout  = imhiststretch( imggray,stin_min,stin_max,stout_min,stout_max )
%%----------Question 6-----------------
%%--------- Mandana Samiei-------------
%%--------- Student Id: 40059116 ------
r = size(imggray,1);
c = size(imggray,2);
M = 256;
in1 = 0:stin_min;
in2 = stin_min:stin_max;
in3 = stin_max:M;

% Slope of 3 lines
m1= (stout_min/stin_min);
m2= (stout_max-stout_min)/(stin_max-stin_min);
m3= (M-stout_max)/(M-stin_max);

% Y-intercept of lines
b1 = 0;
b2= stout_min - m2*stin_min;
b3= stout_max - m3*stin_max;

% Lines
l1=m1*(in1)+ b1;
l2=m2*(in2)+ b2;
l3=m3*(in3)+ b3;
figure;
plot(in1,l1,in2,l2,in3,l3);
imgout = zeros(r,c);

for i=1:r
    for j=1:c
        if imggray(i,j) <= stin_min
              imgout(i,j) = m1*(imggray(i,j)) + b1;
         elseif imggray(i,j) <= stin_max && imggray(i,j) > stin_min
              imgout(i,j) = m2*(imggray(i,j)) + b2;
         elseif imggray(i,j)> stin_max && imggray(i,j) <= 256
              imgout(i,j) = m3*(imggray(i,j)) + b3;
         end
    end
end
imgout = uint8(imgout);
end
```

I calculated the equation of three lines, and multiplied the greyscales of each range by the the corresponding value of the corresponding line in those ranges.

If greyscale <= stin_min → imgout(i,j) = greyscale* line1;

If greyscale <= stin_max && greyscale >= stin_min → imgout(i,j) = greyscale* line2;

If greyscale > stin_max && greyscale < 256 → imgout(i,j) = greyscale* line3;

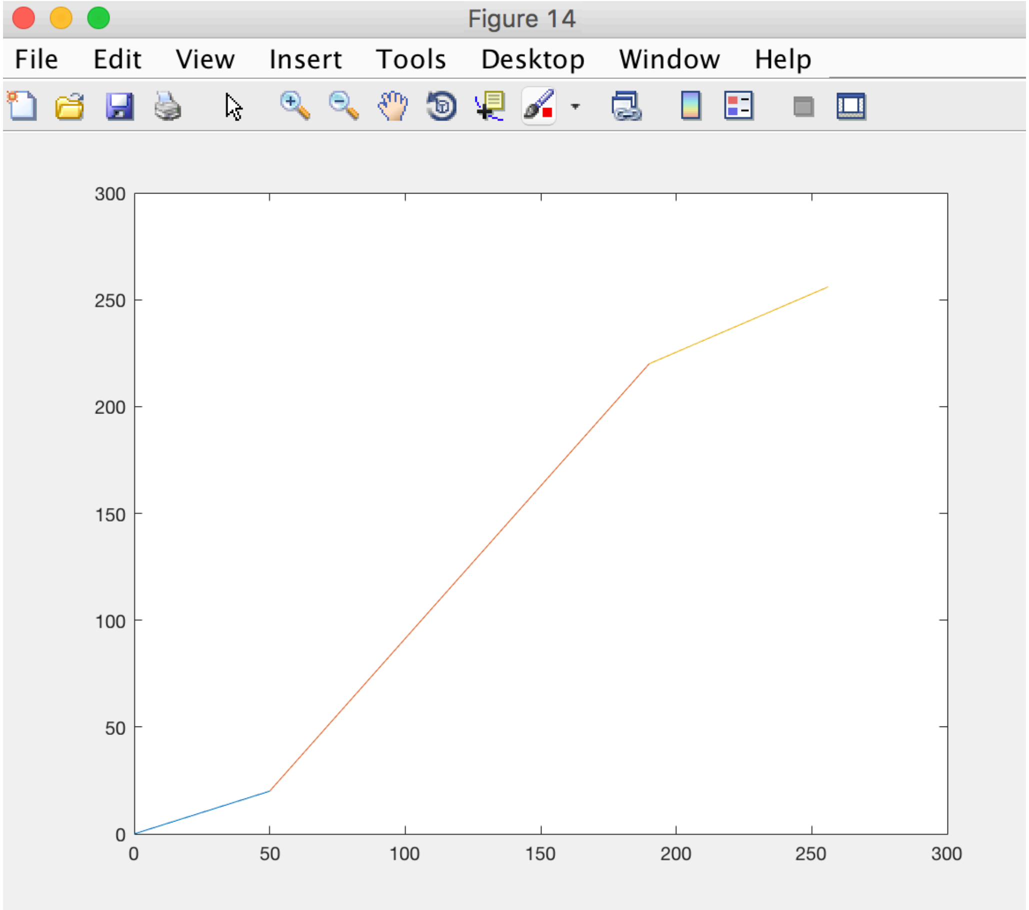
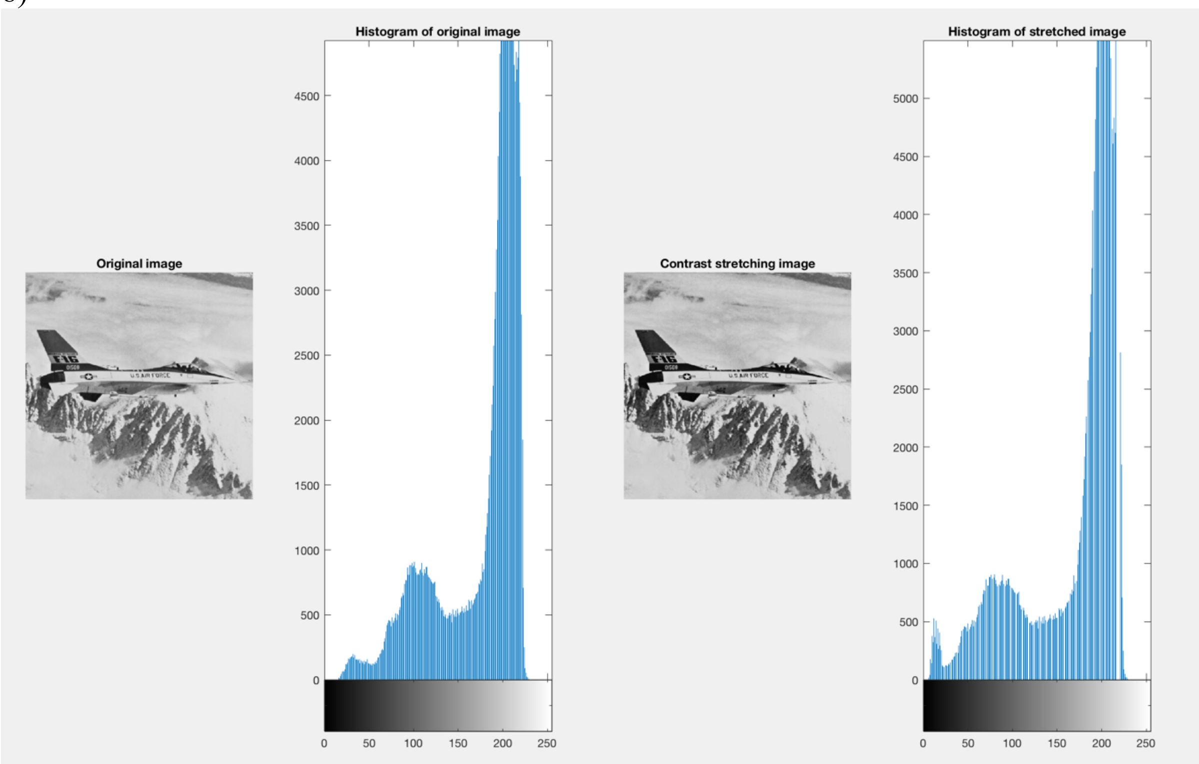


Figure 14
File Edit View Insert Tools Desktop Window Help
300
250
200
150
100
50
0
0 50 100 150 200 250 300

result:
8)


Original image
Histogram of original image
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0 50 100 150 200 250
Contrast stretching image
Histogram of stretched image
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0 50 100 150 200 250

7) It really depends on a specific image and its application as well as the the person who want to recognize the best quality. For example, in some images we prefer to darken the image to be able to observe the details better and in the other images visa versa. But, generally we prefer to have high contrasted images. Since, when we have a high contrast between background and objects and also between different objects in the same scene, we are able to see the specific details well. In this image, we have a big area of sky that has a light color, in order to see the details in the sky such as clouds, we need to darken the background but not that much dark, some kind of grey, and for the plane, we like to be able to read the written information on the plane, so we will lighten the plane body. I tested different greyscales and I found this case better than the others. Also, I think we haven't a unique best quality image because, the best quality is different from various persons point of view.