西北工業大學

June, 2021

Digital Image Processing

Assignment - 2 Report

"Histogram Equalization"

2018380038 Amirbek Raimov

Class No: 101018国际班

Histogram Equalization

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image.

This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

Source codes and Results:

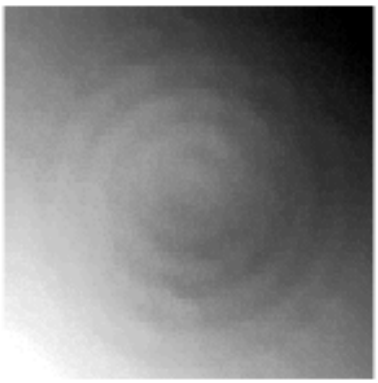
Task 3.3:

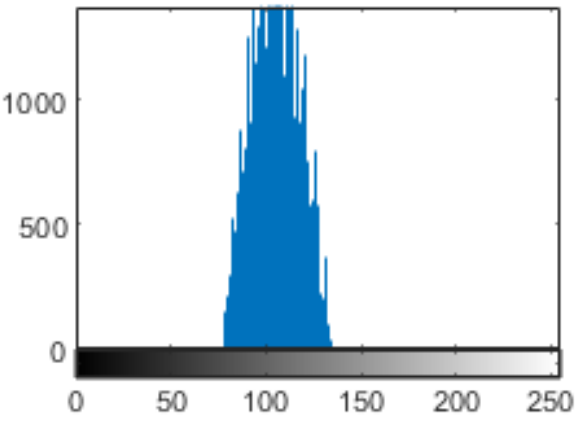
```
1. lena1 = imread("lena1.bmp");
2. lena2 = imread("lena2.bmp");
3. lena3 = imread("lena3.bmp");
4. lena4 = imread("lena4.bmp");
5. hist1 = imread("hist1.bmp");
6.
7. figure('Name','Greyscale histograms','NumberTitle','off');
8. subplot(2,4,1);
9. imshow(lena1);
10.  subplot(2,4,2);
11.  imshow(lena2);
12.  subplot(2,4,3);
13.  imshow(lena3);
14.  subplot(2,4,4);
15.  imshow(lena4);
16.
17.  subplot(2,4,5);
18.  imhist(lena1,256);
19.  subplot(2,4,6);
20.  imhist(lena2,256);
21.  subplot(2,4,7);
22.  imhist(lena3,256);
23.  subplot(2,4,8);
```

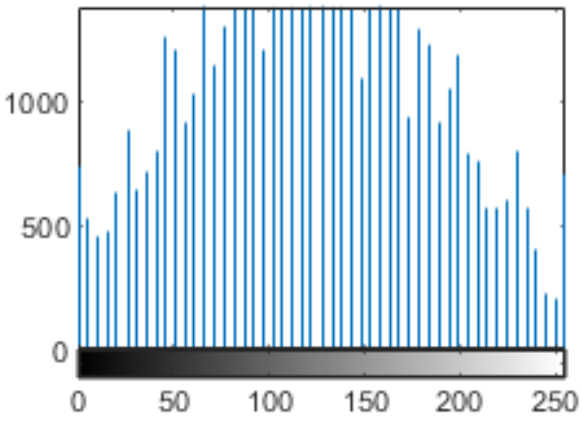
```
24.  imhist(lena4,256);
```

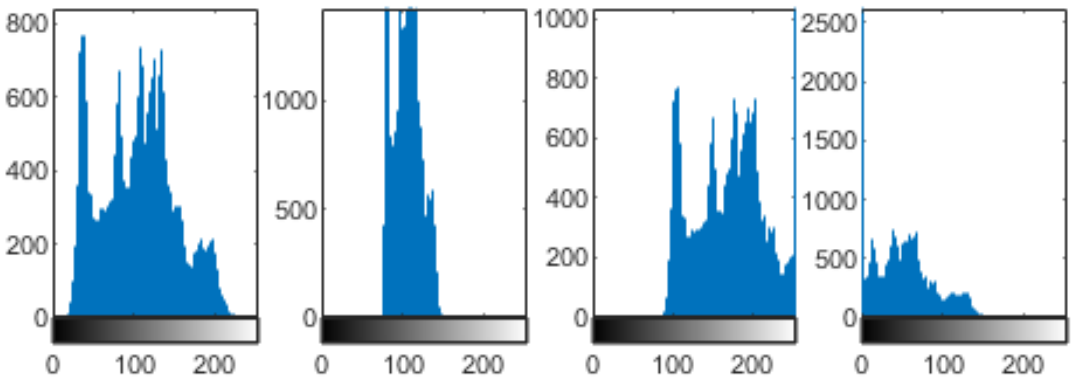
Task 3.4:

```
1. figure('Name','Greyscale histogram -
   stretching','NumberTitle','off');
2. subplot(2,2,1);
3. imshow(hist1);
4. subplot(2,2,3);
5. imhist(hist1,256);
6. subplot(2,2,2);
7. adjusted_hist1 = imadjust(hist1);
8. imshow(adjusted_hist1);
9. subplot(2,2,4);
10.  imhist(adjusted_hist1,256);
```





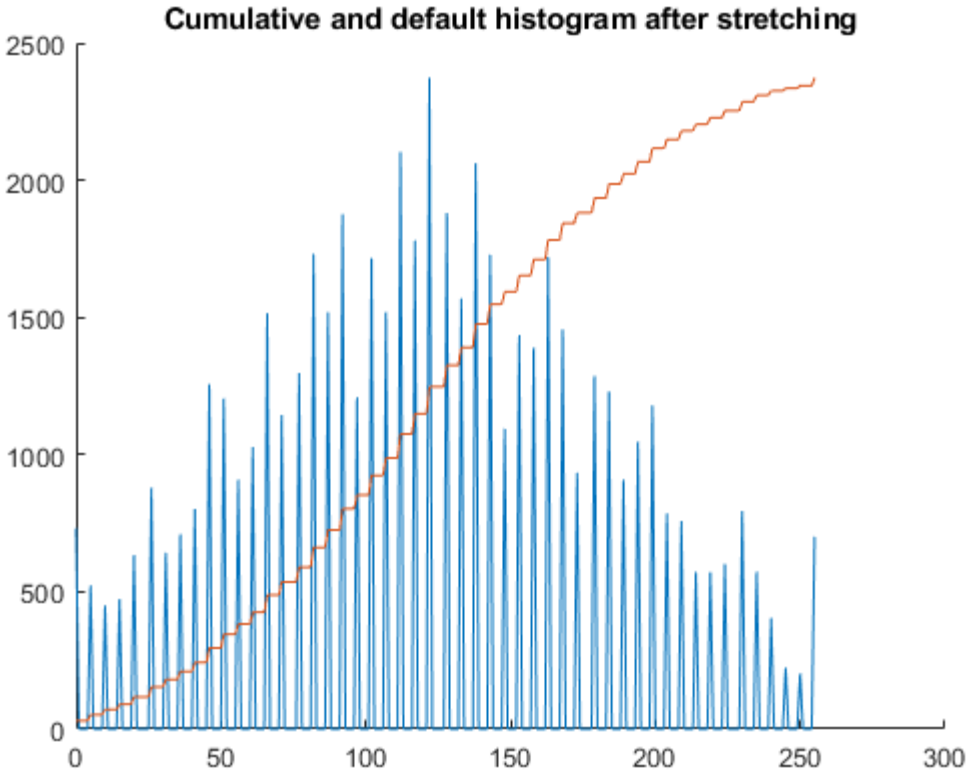




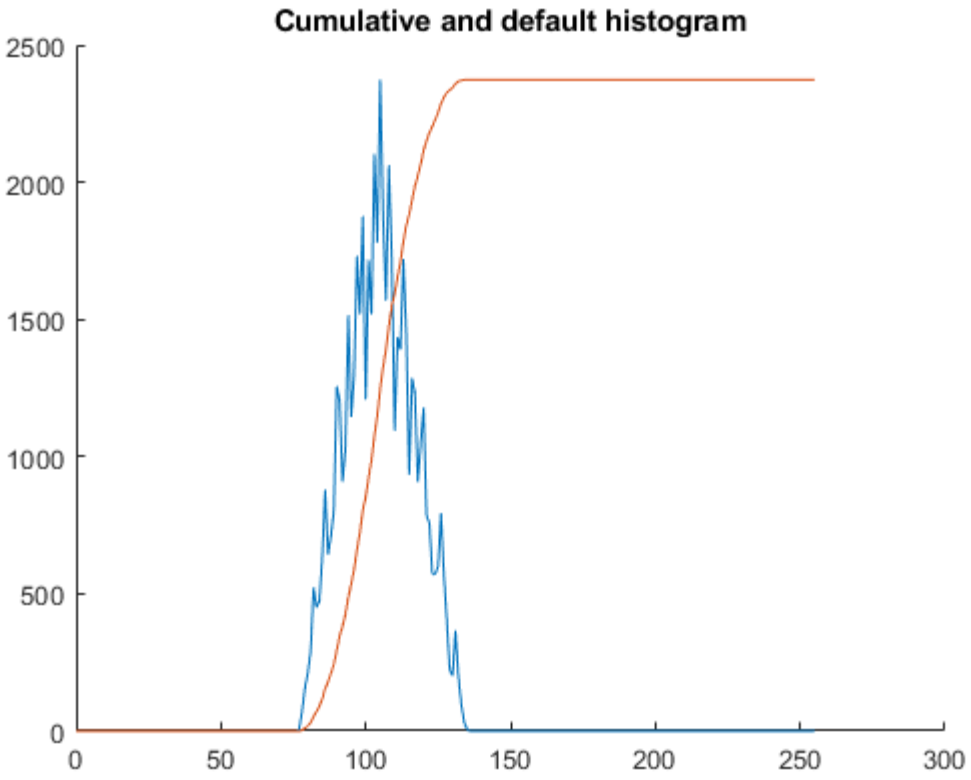
Task 3.5:

```
1. hist1 = imread("hist1.bmp");
2.
3. figure('Name','Cumulative histogram before
   stretching','NumberTitle','off');
4. [H,x] = imhist(hist1);
5.
6. C = cumsum(H);
7. k = max(C)/max(H);
8. C2 = C/k;
9.
10.  hold on
11.  plot(x,H);
12.  plot(x,C2);
13.  title("Cumulative and default histogram");
14.
15.
16.  figure('Name','Cumulative histogram after
   stretching','NumberTitle','off');
17.  adjusted_hist1 = imadjust(hist1);
18.
19.  [H,x] = imhist(adjusted_hist1);
20.
21.  C = cumsum(H);
22.  k = max(C)/max(H);
23.  C2 = C/k;
24.
25.  hold on
26.  plot(x,H);
27.
28.  plot(x,C2);
```

```
29.  title("Cumulative and default histogram after stretching");
```

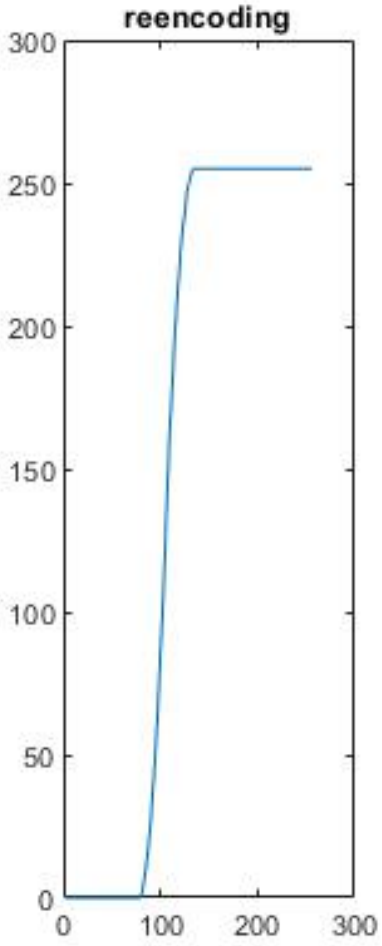


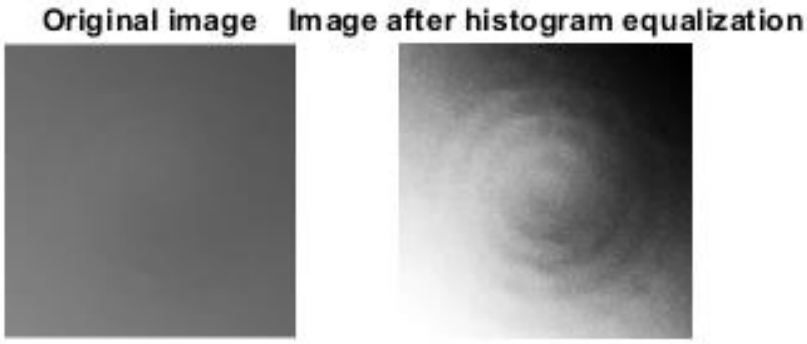

```
1. hist1 = imread("hist1.bmp");
```

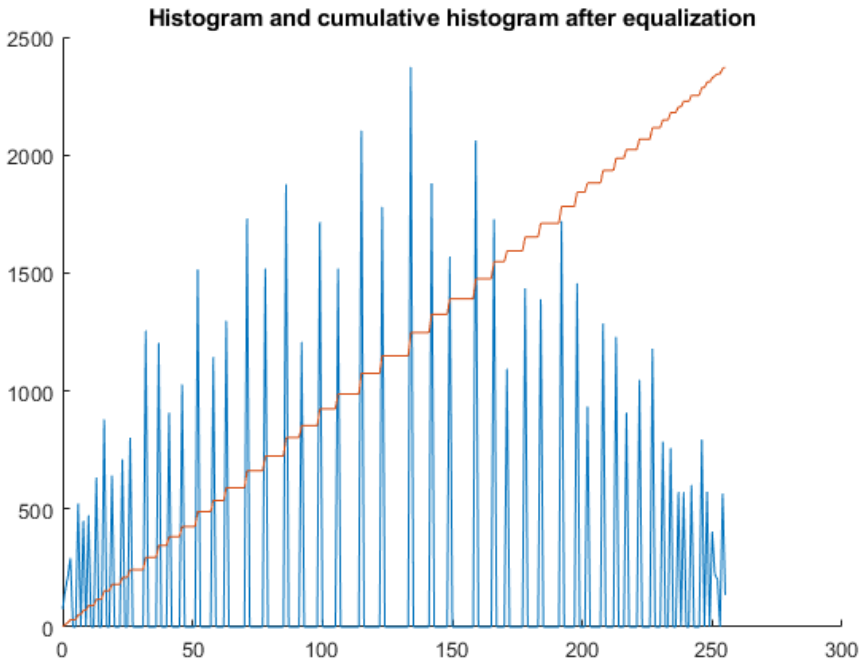


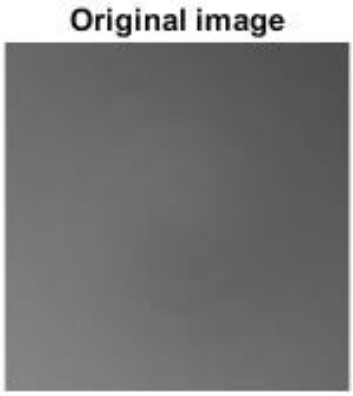
```
2.
3. figure('Name','Histogram equalization - LUT
   function','NumberTitle','off');
4. [H,x] = imhist(hist1);
5. C = cumsum(H);
6. LUT_he(hist1, C);
7.
8. figure('Name','Histogram equalization - histeq &
   adapthisteq','NumberTitle','off');
9. subplot(2,2,1:2);
10.  imshow(hist1);
11.  title("Original image");
12.  subplot(2,2,3);
13.  hist1_he = histeq(hist1,256);
14.  imshow(hist1_he);
15.  title("Image after histogram equalization - histeq");
16.
17.  subplot(2,2,4);
18.  hist1_adapthisteq = adapthisteq(hist1);
19.  imshow(hist1_adapthisteq);
20.  title("Image after histogram equalization - adapthisteq");
```

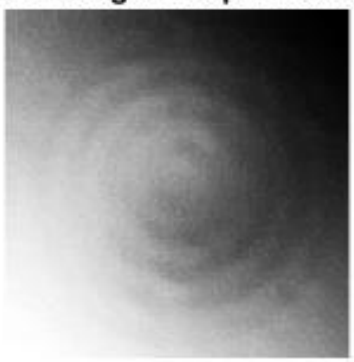
```
1. function LUT_he(image,reencoding)
2. rescaled_reencoding = rescale(reencoding,0,255);
3. rescaled_reencoding = uint8(rescaled_reencoding);
4. A = intlut(image,rescaled_reencoding);
5. subplot(1,3,1);
6. plot(rescaled_reencoding);
7. title("reencoding");
8. subplot(1,3,2);
9. imshow(image);
10.  title("Original image");
11.  subplot(1,3,3);
12.  imshow(A);
13.  title("Image after histogram equalization");
14.  end
```

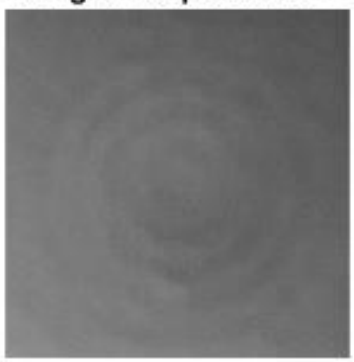

reencoding
300
250
200
150
100
50
0
0
100
200
300


Original image
Image after histogram equalization


Histogram and cumulative histogram after equalization
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

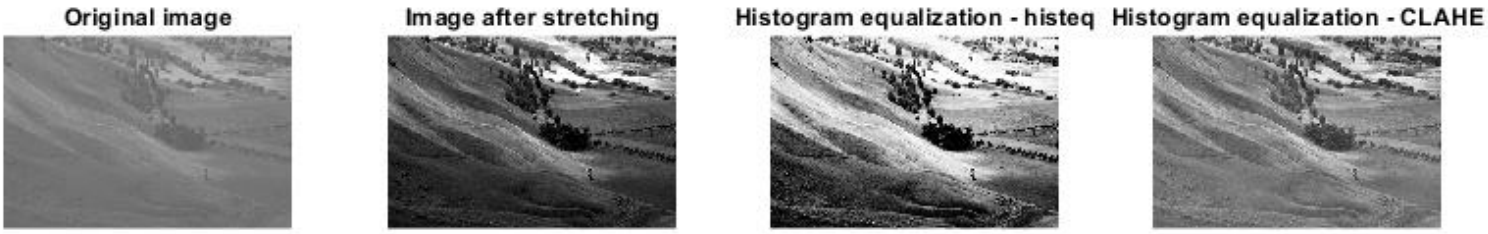


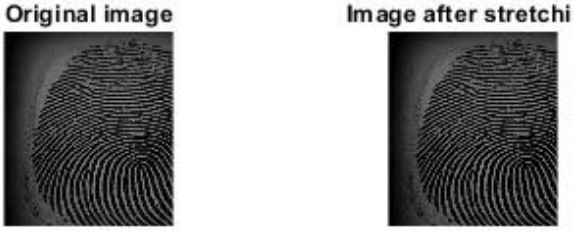


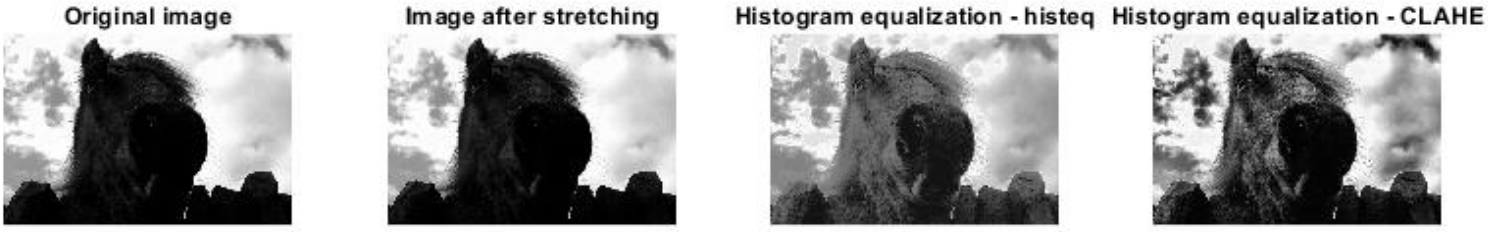



```
1. hist2 = imread("hist2.bmp");
2. hist3 = imread("hist3.bmp");
3. hist4 = imread("hist4.bmp");
4.
5. figure('Name','Real images','NumberTitle','off');
6. subplot(3,4,1);
7. imshow(hist2);
8. title("Original image");
9.
10.  subplot(3,4,2);
11.  adjusted_hist2 = imadjust(hist2);
12.  imshow(adjusted_hist2);
13.  title("Image after stretching");
14.
15.  subplot(3,4,3);
16.  hist2_he = histeq(hist2);
17.  imshow(hist2_he);
18.  title("Histogram equalization - histeq");
19.
20.  subplot(3,4,4);
21.  hist2_clahe = adapthisteq(hist2);
22.  imshow(hist2_clahe);
23.  title("Histogram equalization - CLAHE");
24.
25.  subplot(3,4,5);
26.  imshow(hist3);
27.  title("Original image");
28.
29.  subplot(3,4,6);
```

```
adjusted_hist3 = imadjust(hist3);
imshow(adjusted_hist3);
title("Image after stretching");

subplot(3,4,7);
hist3_he = histeq(hist3);
imshow(hist3_he);
title("Histogram equalization - histeq");

subplot(3,4,8);
hist3_clahe = adapthisteq(hist3);
imshow(hist3_clahe);
title("Histogram equalization - CLAHE");

subplot(3,4,9);
imshow(hist4);
title("Original image");

subplot(3,4,10);
adjusted_hist4 = imadjust(hist4);
imshow(adjusted_hist4);
title("Image after stretching");

subplot(3,4,11);
hist4_he = histeq(hist4);
imshow(hist4_he);
title("Histogram equalization - histeq");

subplot(3,4,12);
hist4_clahe = adapthisteq(hist4);
imshow(hist4_clahe);
title("Histogram equalization - CLAHE");
```

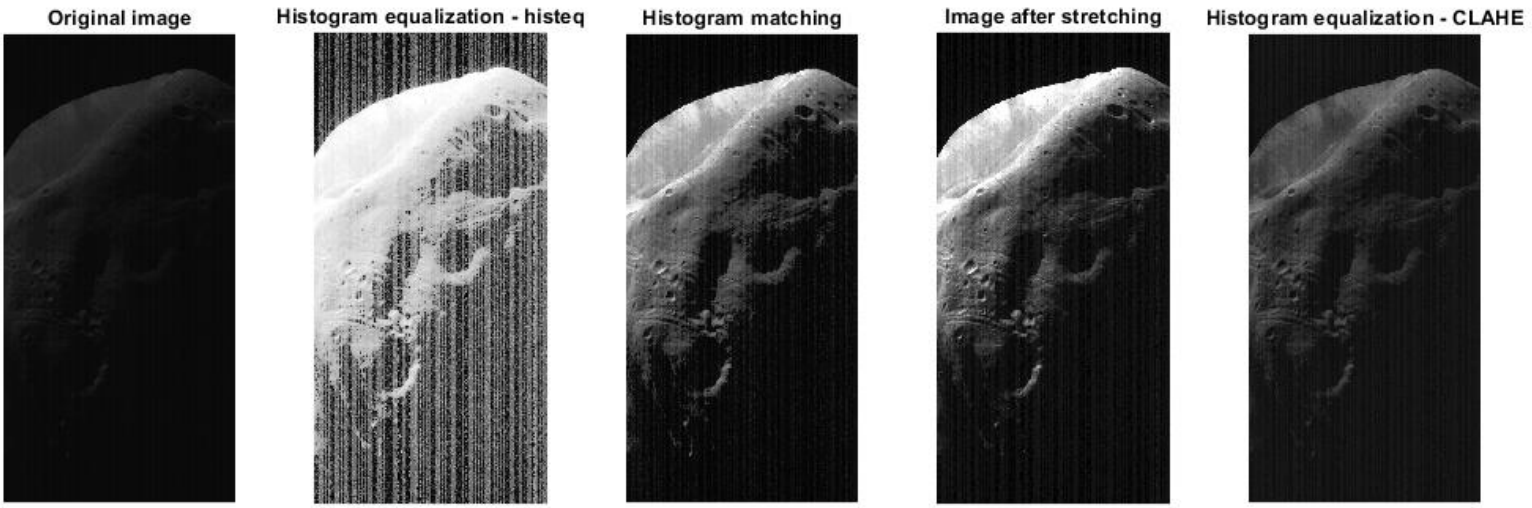






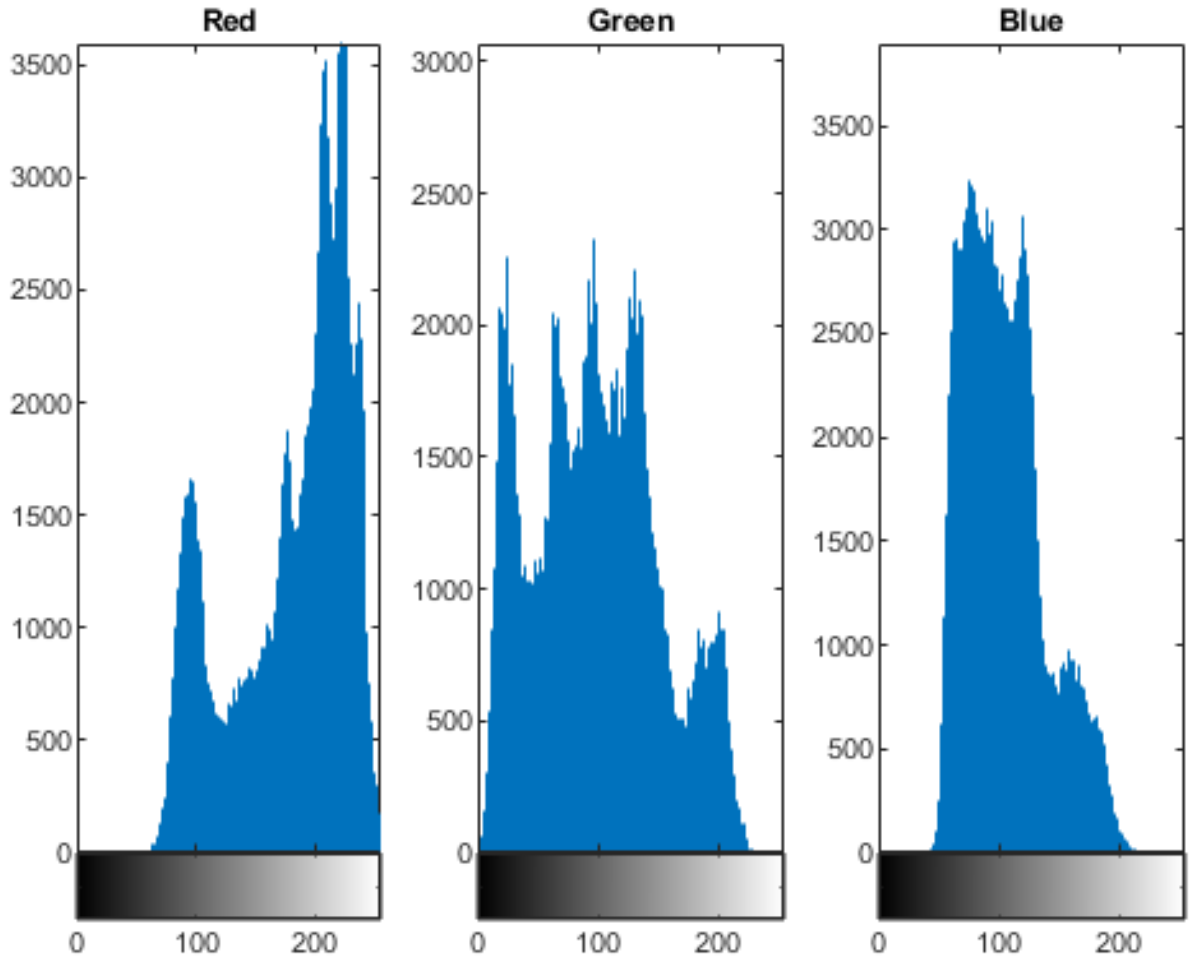
Task 3.6:

```
load desiredHistogram.mat
phobos = imread("phobos.bmp");

figure('Name','Histogram matching','NumberTitle','off');

subplot(1,5,1);
imshow(phobos);
title("Original image");
subplot(1,5,2);
 phobos_he = histeq(phobos);
 imshow(phobos_he);
 title("Histogram equalization - histeq");
 subplot(1,5,3);
 phobos_he = histeq(phobos,desiredHistogram);
 imshow(phobos_he);
 title("Histogram matching");
 subplot(1,5,4);
 adjusted_phobos = imadjust(phobos);
 imshow(adjusted_phobos);
 title("Image after stretching");
 subplot(1,5,5);
 phobos_clahe = adapthisteq(phobos);
 imshow(phobos_clahe);
 title("Histogram equalization - CLAHE");
```



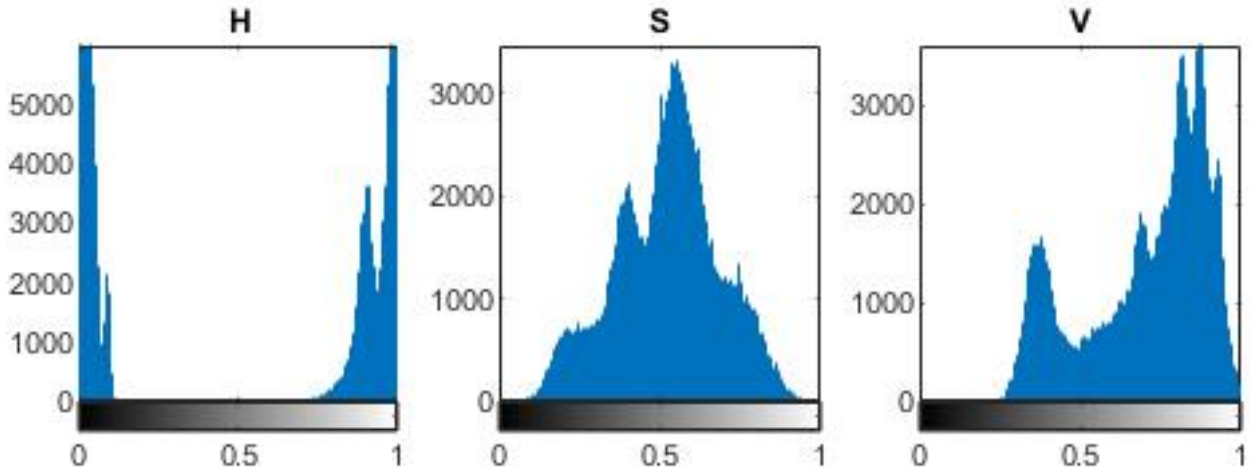
Task 3.7:

```
lena = imread("lake.jpg");

figure('Name','RGB histogram','NumberTitle','off');
lenaR = lena(:,:,1);
lenaG = lena(:,:,2);
lenaB = lena(:,:,3);
subplot(1,3,1);
imhist(lenaR);
title("Red");
 subplot(1,3,2);
 imhist(lenaG);
 title("Green");
 subplot(1,3,3);
 imhist(lenaB);
 title("Blue");

 figure('Name','RGB histogram equalization','NumberTitle','off');
 lenaR = histeq(lenaR);
 lenaG = histeq(lenaG);
 lenaB = histeq(lenaB);
 lena_eq = lena;
 lena_eq(:,:,1) = lenaR;
 lena_eq(:,:,2) = lenaG;
 lena_eq(:,:,3) = lenaB;
 subplot(1,2,1);
 imshow(lena);
 title("Original image");
 subplot(1,2,2);
 imshow(lena_eq);
 title("Image after equalization");

 figure('Name','HSV','NumberTitle','off');
 lena_hsv = rgb2hsv(lena);
 lena_h = lena_hsv(:,:,1);
 lena_s = lena_hsv(:,:,2);
 lena_v = lena_hsv(:,:,3);
 subplot(2,3,1);
 imhist(lena h);
 title("H");
 subplot(2,3,2);
 imhist(lena_s);
 title("S");
 subplot(2,3,3);
 imhist(lena_v);
 title("V");

 lena_hsv_eq = lena_hsv;
 lena_v = histeq(lena_v);
 lena_hsv_eq(:,:,1) = lena_h;
 lena_hsv_eq(:,:,2) = lena_s;
 lena_hsv_eq(:,:,3) = lena_v;

 RGB = hsv2rgb(lena_hsv_eq);
 subplot(2,3,4:6);
 imshow(RGB);
 title("HDV equalization");
```

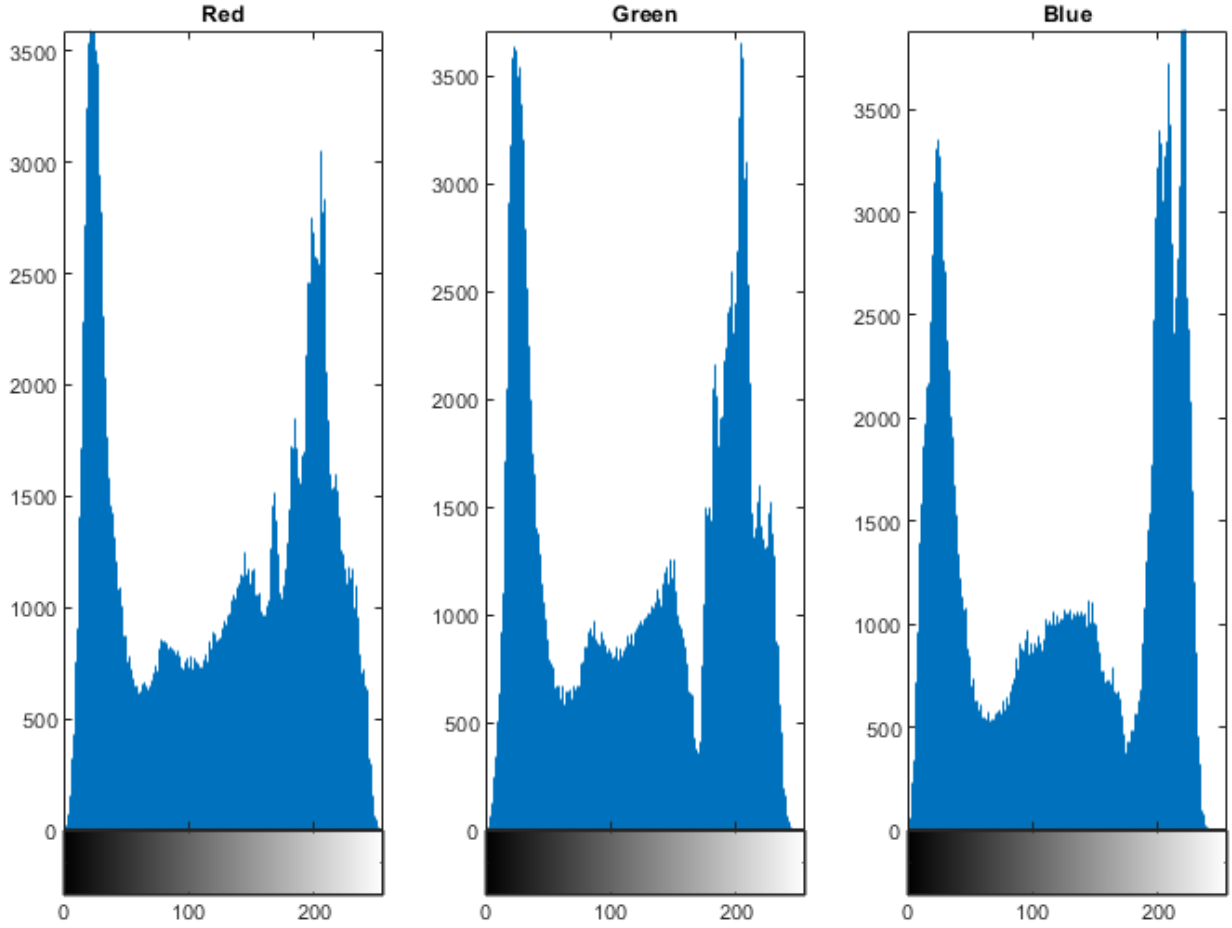


Original image

Image after equalization

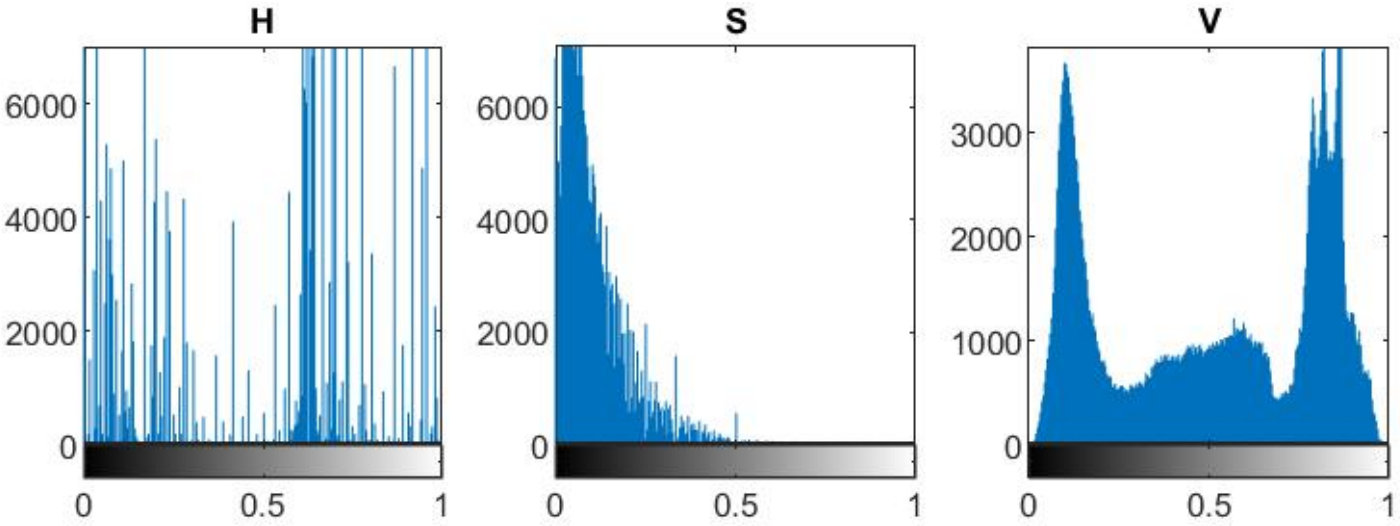


HDV equalization



Original image

Image after equalization



HDV equalization

Conclusions

If the histogram is on the right side of the scale it means that the value of pixels is closer maximal value of greyscale so the whole image is brighter. Otherwise, values of pixels are closer to 0 which means darker color. If the shape of a histogram is wider, the range of pixels' value is also greater. The image is sharper and the draft is clear. There is hard to observe what is on the image called hist1.bmp but after stretching, the range of greyscale is wider and the output is more clear. The number of details is greater.

The shape of a cumulative histogram is growing. Function histeq and implemented LUT function give the same result. I think that the best result in case of real images gives histeq functions and histogram stretching.

The image of Phobos has a bit of noise so the output from histeq function gives a more noised result with white vertical lines. The other methods are presenting well and again in my opinion the stretching gives the sharpest image.

The simple Histogram equalization method in case of RGB images consisting in splitting the image into three separated colors is more complicated and might be very inefficient at high resolutions. It also makes the image to lose color. Using HSV format does not lead to this situation.