**EXPERIMENT NO. 2**

**PERFORM AND COMPARE CONTRAST STRETCHING AND HISTOGRAM EQUALIZATION**

**EXPERIMENT NO. 2**

**AIM**: To perform and compare contrast stretching and histogram equalization

**OBJECTIVES:**

1. To understand concept of contrast enhancement
2. To perform enhancement of image using contrast stretching.
3. To perform enhancement of image using histogram equalization.
4. To compare performance of both the methods.

**EQUIPMENTS/SOFTWARE:** SCILAB 6.0.0

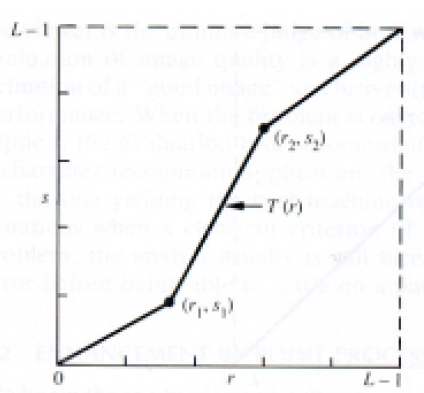
**THEORY:**

**Contrast Stretching/Compression**

Stretch gray-level ranges where we desire more information.

Low-contrast images can result from poor illumination, lack of dynamic range in the image sensor, or even wrong setting of a lens aperture during image acquisition.

The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.



**Steps:**

1. Read the input image and its size.
2. Obtain values of a, b, v, w.
3. For every pixel of input image check its value and modify to new value accordingly.
4. Display input and output images with title.

**Histogram Equalisation:**

Histogram of a digital image with gray levels in range [0,L-1] is a discrete function h(rk) = nk where rk -kth gray level and nk = no. of pixels of an image having gray level rk.

In histogram there are 3 possibilities as follows,

1. For a dark image the components of histogram on the low (dark) side.

2. For a bright image the component are on high ( bright ) side

3. For an image with low contrast they are in the middle of gray side.

Histogram equalization is done to spread there component uniformly over the gray scale as far as possible.

This is obtained by function

Where

Thus processed image is obtained by mapping each pixel with level rk into a corresponding pixel with level Sk in output image. This transformation is called Histogram equalization

**ALGORITHM:**

1. Read the input image and its size.
2. Obtain probability of each the gray level values of each pixel from the image.
3. Compute CDF for each gray value.
4. Compute new value for each input grey level by multiplying its CDF by 255.
5. Replace the input gray values with corresponding new values Sk.
6. Plot the equalized histogram and original histogram
7. Display the original and the equalized image.

**PROGRAM OF CONTRAST STRETCHING:**

*//Contrast stretching*

*//118A2044 Dikshita Kambri*

clc;

clear all;

im = imread(')

imshow(im);

im = double(im);

[r,c] = size(im);

a =100;

b =170;

v=50;

w =200;

l = v/a;

m = (w-v)/(b-a);

n = (255-w)/(255-b);

for(i = 1:r)

for j=1:c

if im(i,j) < a then

jm(i,j) =

end

**PROGRAM OF HISTOGRAM EQUALIZATION:**

*// Histogram Equalisation*

*//118A2044 Dikshita Kambri*

clc;

clear all;

im=imread("C:\Users\hp\Documents\Image Processing-Scilab\Images\onion.png");

figure(1),

imshow(im);

title('Imput image');

im=double(im);

[r c]=size(im);

jm=zeros(r,c);*//output matrix*

*//input histogram*

nk=zeros(1,256);

*//for lvl=0:255*

*// a=find(im==lvl);*

*// l=lvl+1;*

*// nk(l)=length(a);*

*//end*

t=0;

for p=1:r

for q=1:c

t=im(p,q);

nk(t+1)=nk(t+1)+1;

end

end

figure(2), bar(nk);

title('Original image histogram');

pdf=nk/(r\*c);

cdf(1)= pdf(1);

for i=2:256

cdf(i)=cdf(i-1)+pdf(i);

end

sk=round(255\*cdf);

*//for rk=0:255*

*// b=find(im==rk);*

*// r1=rk+1;*

*// jm(b)=sk(r1);*

*//end*

*// Replacing the input pixels by new value*

for x=1:r

for y=1:c

temp=im(x,y);

jm(x,y)=sk(temp+1);

end

end

mk=zeros(r,c);

*//for v=0:255*

*// p=find(jm==v);*

*// v1=v+1;*

*// mk(v1)=length(p);*

*//end*

*// Histogram of the equalised image*

s=0;

for p=1:r

for q=1:c

s=jm(p,q);

mk(s+1)=mk(s+1)+1;

end

end

figure(3), imshow(uint8(jm));

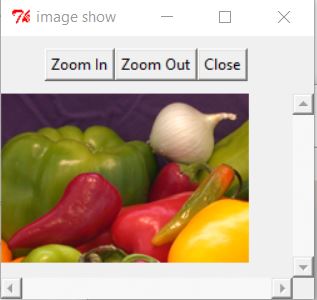
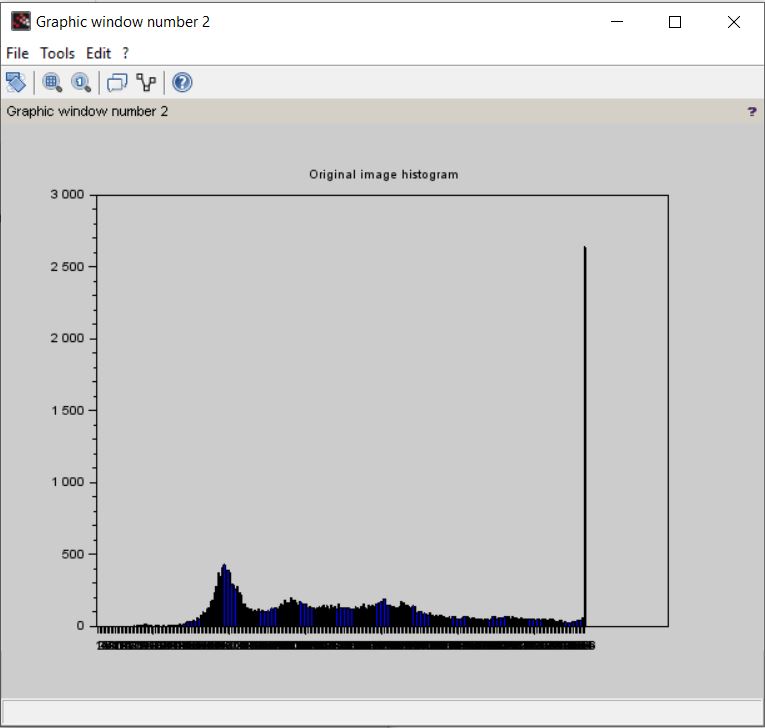
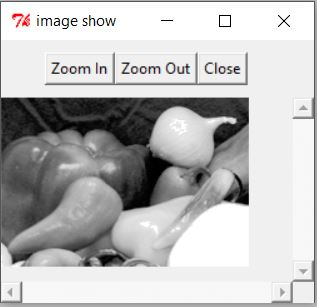
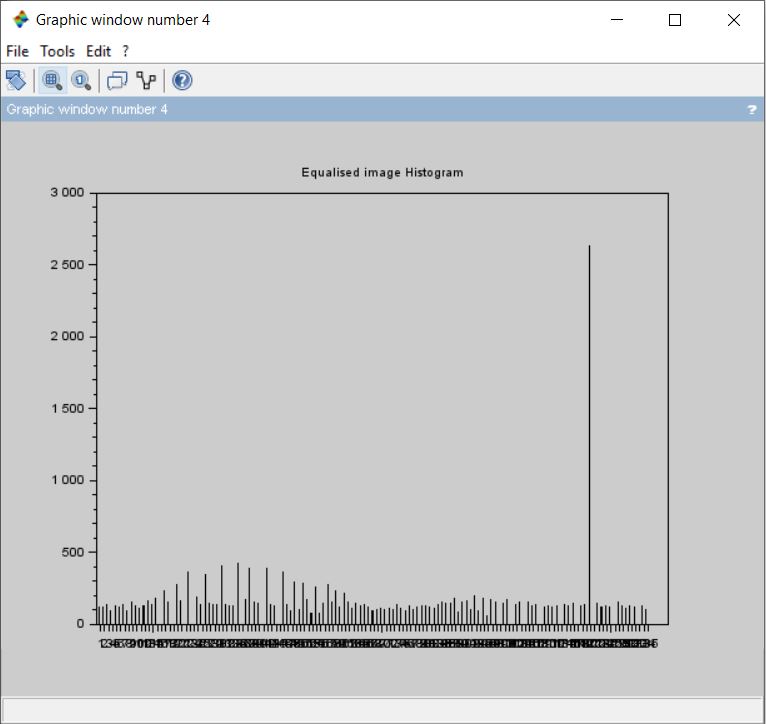
title('Equalised image');

figure(4), bar(mk);

title('Equalised image Histogram');

**OUTPUT OF CONTRAST STRETCHING:**

**OUTPUT OF HISTOGRAM EQUALIZATION:**

1. **Original image:  
   **
2. **Original image Histogram:  
   **
3. **Equalized image:  
   **
4. **Equalized image histogram:  
   **

**CONCLUSION:**