

Image Enhancement

This technique including smoothing, sharpening, highlighting features, normalizing illumination...etc.

There are two approaches for image enhancement:

1. Image Enhancement in the Spatial Domain.
2. Image Enhancement in the Frequency Domain.

Spatial domain techniques

1. Point operations
2. Histogram

1. Point operations

- **Image Negative**

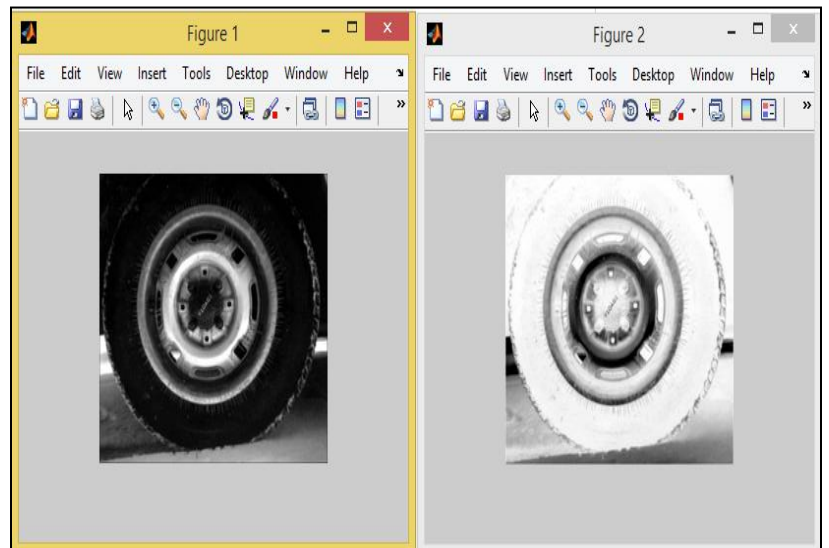
image negative applying by using function:

$$IM2 = \text{imcomplement}(IM)$$

computes the complement of the image IM. IM can be a binary, grayscale, or RGB image. IM2 has the same class and size as IM.

Ex:

```
clc
clear all
I=imread('tire.tif');
figure,imshow(I)
J=imcomplement(I);
figure, imshow(J)
```



❖ Notice how you can now see the writing in the middle of the tire better than before.

- **Contrast Stretching**

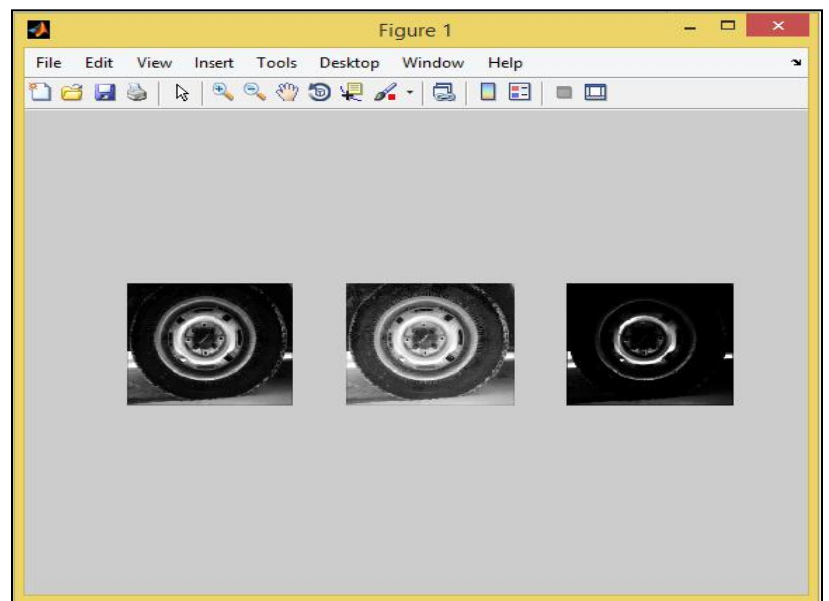
This technique will enhance contrast of given image by using function:

```
J = imadjust(I,[low_in; high_in],[low_out; high_out],gamma)
```

maps the values in I to new values in J, where gamma specifies the shape of the curve describing the relationship between the values in I and J. If gamma is less than 1, the mapping is weighted toward higher (brighter) output values. If gamma is greater than 1, the mapping is weighted toward lower (darker) output values.

Ex:

```
clc
clear all
J=imread('tire.tif');
J2=imadjust(J,[],[],0.4);
J3=imadjust(J,[],[],2);
subplot(1,3,1),imshow(J)
subplot(1,3,2),imshow(J2)
subplot(1,3,3),imshow(J3)
```



2. Histogram

Histogram can tell us whether image was scanned properly or not. It gives us idea about tonal distribution in the image.

MATLAB easily displays image histograms using the function:

```
imhist(I)
```

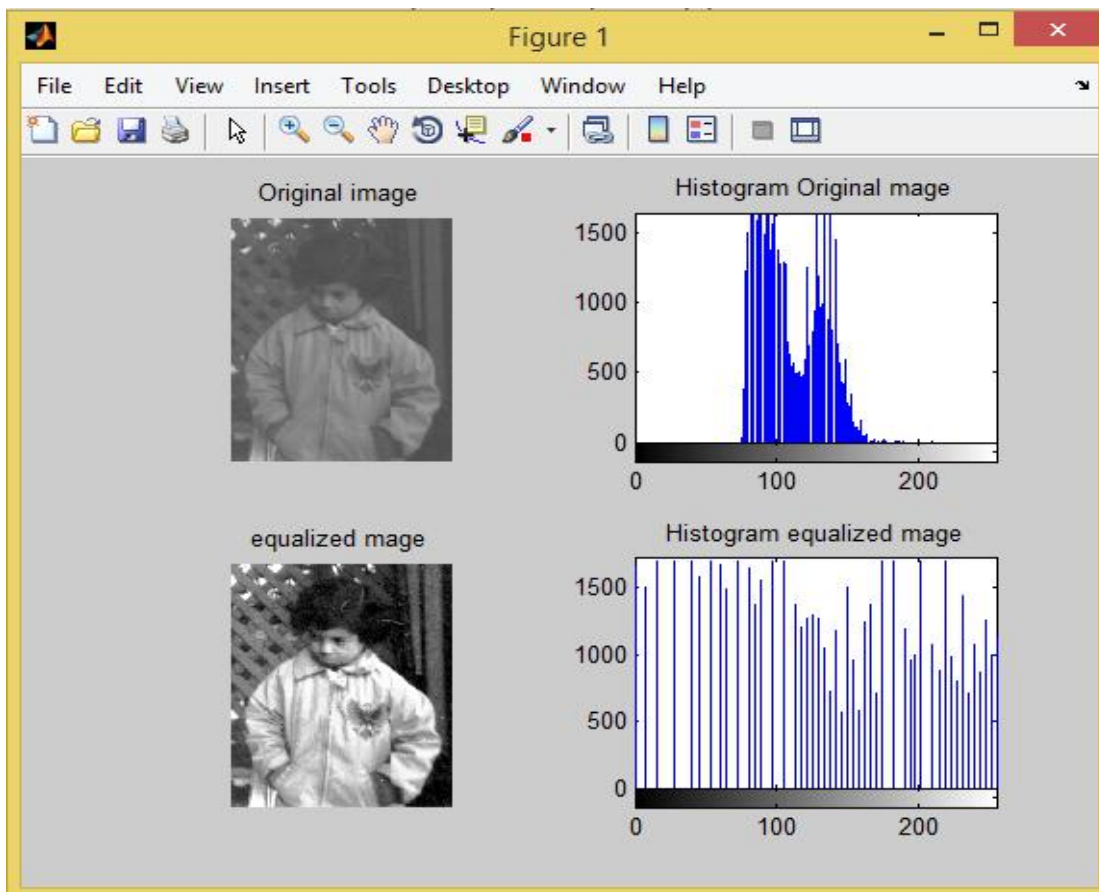
displays a histogram for the image I .

Histogram equalization can be applied to improve appearance of the image using the function:

```
histeq (I)
```

EX:

```
clc
clear all
a=imread('pout.tif');
subplot(2,2,1) ,imshow(a),title('Original image')
subplot(2,2,2), imhist(a),title('Histogram Original mage')
b=histeq(a);
subplot(2,2,3), imshow(b),title('equalized mage')
subplot(2,2,4), imhist(b),title('Histogram equalized mage')
```



Enhancement Using Arithmetic/Logic Operations

1. Arithmetic Operations

Arithmetic operations like image **addition**, **subtraction**, **multiplication** and **division** is possible. These operations act by applying a simple function.

Addition operation can be used to increase brightness of the image by add constant value.

to **add** two images or add constant to image using the following function:

$$Z = \text{imadd}(X, Y)$$

adds each element in array X with the corresponding element in array Y and returns the sum in the corresponding element of the output array Z.

Subtraction operation can be used to decrease brightness of the image by subtract constant value.

to **subtract** one image from another or subtract constant from image using the following function:

$$Z = \text{imsubtract}(X, Y)$$

subtracts each element in array Y from the corresponding element in array X and returns the difference in the corresponding element of the output array Z.

Multiplication operation can also be used to increase brightness of the image.

to **multiply** two images or multiply image by constant using the following function:

$$Z = \text{immultiply}(X, Y)$$

multiplies each element in array X by the corresponding element in array Y and returns the product in the corresponding element of the output array Z.

Division operation can be used to decrease the brightness of the image.

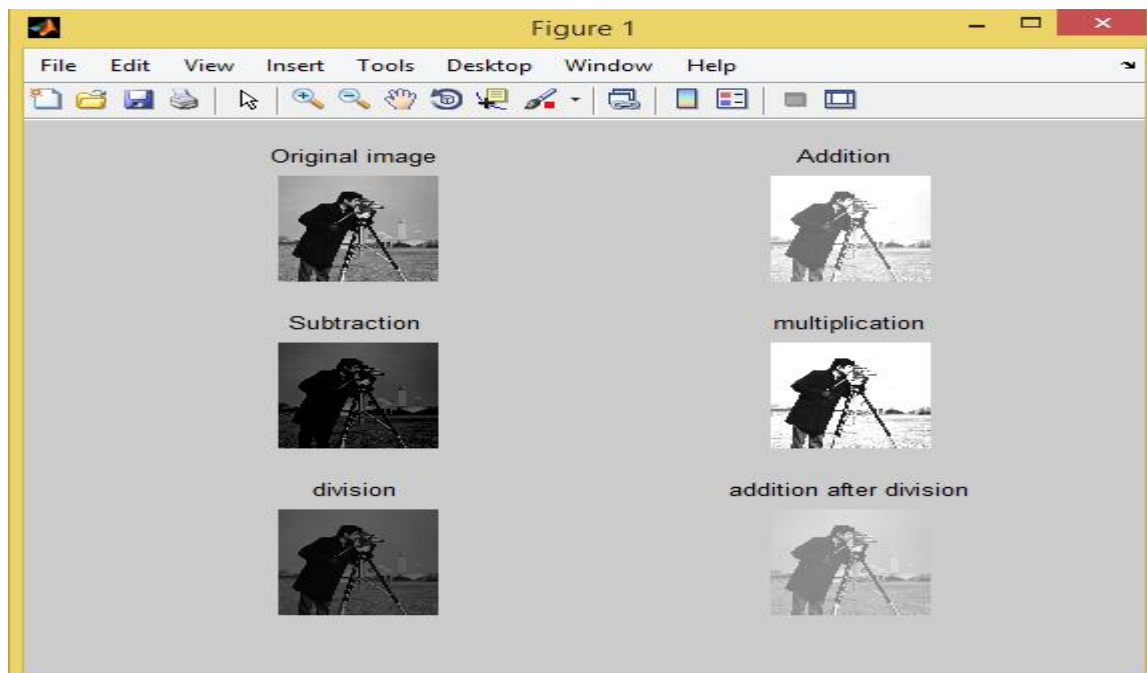
to **divide** one image into another or divide image by constant using the following function:

$$Z = \text{imdivide}(X, Y)$$

divides each element in the array X by the corresponding element in array Y and returns the result in the corresponding element of the output array Z.

EX:

```
clc
b=imread('cameraman.tif');
subplot(3,2,1),imshow(b),title('Original image');
b1=imadd(b,100);
subplot(3,2,2),imshow(b1),title('Addition ');
b2=imsubtract(b,100);
subplot(3,2,3),imshow(b2),title('Subtraction');
b3=immultiply(b,2);
subplot(3,2,4),imshow(b3),title('multiplication');
b4=imdivide(b,2);
subplot(3,2,5),imshow(b4),title('division');
b5=imadd(imdivide(b,2),128);
subplot(3,2,6),imshow(b5),title('addition after division');
```



2. Logical operations

Logical operations like (**AND, OR, XOR, NOT**) is possible. These operations act by applying a simple function.

AND Logical operations can be used for some applications such as (Compute intersection of the images, Design of filter masks and Slicing of gray scale images), it applies by using the function:

$$C=\text{and}(A, B)$$

performs a logical AND of all input arrays A, B and returns an array containing elements set to either logical 1 (true) or logical 0 (false).

OR logical operations can be used for some applications such as merging of two images; it applies by using the function:

$$C = \text{or}(A, B)$$

performs a logical OR of all input arrays A, B and returns an array containing elements set to either logical 1 (true) or logical 0 (false).

XOR logical operations can be used for some applications such as (To detect change in gray level in the image and Check similarity of two images), it applies by using the function:

$$C = \text{xor}(A, B)$$

performs an exclusive OR operation on the corresponding elements of arrays A and B. The resulting element $C(i,j,...)$ is logical true (1) if $A(i,j,...)$ or $B(i,j,...)$, but not both, is nonzero.

NOT logical operation is used to obtain negative image and making some features clear, it applies by using the function:

$$C=\text{not}(A)$$

performs a logical NOT of input array A, and returns an array containing elements set to either logical 1 (true) or logical 0 (false).

EX:

```
clc
AA=imread('office_5.jpg');
A=rgb2gray(AA);
A=im2bw(A);
A=imresize(A,[256,256]);
subplot(3,2,1),imshow(AA),title('Image AA ');
%=====
BB=imread('football.jpg');
B=rgb2gray(BB);
B=im2bw(B);
B=imresize(B,[256,256]);
subplot(3,2,2),imshow(BB),title('Image BB');
xorimage= xor(A,B);
subplot(3,2,3),imshow(xorimage), title('Image A XOR Image B');
orimage= or(A , B);
subplot(3,2,4),imshow(orimage), title('Image A OR Image B ');
andimage= and(A ,B);
subplot(3,2,5),imshow(andimage), title('Image A AND Image B ');
notA= not(A) ;
subplot(3,2,6),imshow(notA), title('Not of Image A');
```

