A Project report on

Digital Image Processing using matlab

Submitted by

O161636
O161817
O160793
O161881
O161724

Submitted to

Under the Supervision of
Mr.V.Leeladhar

Assistant Professor

As a part of
Partial fulfilment of the degree of Bachelor of Technology in Electronics and Communication Engineering
Date:

Certificate

This is to certify that the report entitled “Digital Image Processing using matlab ” submitted by K.Kamal Hanesha,K.Ganapathi Reddy,G.Naveen,M.Swetha and V.Jenifer bearing ID numbers O161636, O161817,O160793,O181881 and O161724 respectively in partial fulfilment of the requirements for the award of Bachelor of Technology in Electronics and Communication Engineering is a bona fide work carried by them under my supervision and guidance.

Head of the Department

V.S.Sriharsha Kasukarthy

Project Internal Guide,

V.Leeladhar

Acknowledgement

I would like to express my sincere gratitude to **V.Leeladhar**, my project guide for valuable suggestions and keen interest throughout the progress of my course and research.

I am grateful to **V.S.Sriharsha Kasukarthy** , HOD of **Electronics & Communication Engineering**, for providing excellent computing facilities and a congenial atmosphere for progressing with my project.

At the outset, I would like to thank **Rajiv Gandhi University of Knowledge Technologies, R.K Valley** for providing all the necessary resources for the successful completion of my course work. At last, but not the least I thank my teammates and other students for their physical and moral support.

With Sincere Regards,
K.Kamal Hanesha
K.Ganapathi Reddy
G.Naveen
M. Swetha
V. Jenifer

1.Abstract

Human eye is most attracted by an image more than an audio.In this project we are dealing with digital images with the help of matlab software.Matlab is a very simple software for coding.Here,In this project we deal with basic image processing like Image conversions,Image Arithmetic,Geometric transformations of images,Image smoothening,image sharpening,image blurring,histograms of different contrast images,histogram equalization,histogram matching,bit plane slicing,RGB components of image,Noise in images and its reduction,image fusion,edge detection,morphological image processing and image compression.These all are implemented using matlab software.Here,in this project we implement matlab code for particular operation on an image and process the images into the code ,output images are plotted using figures in matlab software.

Image processing helps to improve images for human interpretation.The processing of images is faster and cost-effective Digital Image Processing has different techniques in processing the digital images.It has been applying in many fields with technological advances such as Medical field,Remote sensing,color processing,pattern recognition,video processing,microscopic imaging.

Keywords:

Digital image processing ,matlab, code ,human interpretation,faster and cost-effective,Imageconversions,smoothening,sharpening,bluring,bitplane slicing,image compression,histogram equalization and matching,geometric transformation in images,Medical field,Remote sensing,color processing,pattern recognition,video processing,microscopic imaging.

1) Kandukuri Kamal Hanesha-O161636
2)Kosuru Ganapathi Reddy -O161817
3)Guduru Naveen-O160793
4)Mallu Swetha-O161881
5)Vannela Jenifer-O161724 V.Leeladhar

Signature of the student Signature of the Guide

2.Introduction

Human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. We not only look at things to identify andclassify them, but we can scan for differences, and obtain an overall rough “feeling” for a scene witha quick glance.Humans have evolved very precise visual skills: we can identify a face in an instant; we can differentiate colours; we can process a large amount of visual information very quickly. Even a large solid structure, like a building or a mountain, will change its appearance depending on the time of day (day or night); amount of sunlight (clear or cloudy), or various shadows falling upon it.Human Eye attracts more for an image more than audio.We can understand more when we see content in the picture and it will be remembered for more time.Images play the single most important role in human perception.

1. Digital Image Processing is mainly used for two important applications.
    - improvement of pictorial information for human interpretation,
    - processing of image data for tasks such as storage,transmission, and extraction of pictorial information.
2. An image is a two-dimensional function, f (x , y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x , y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of ‘f ‘are all finite, discrete quantities, then this is called as digital image.
3. Digital image is composed of a finite number of elements, each of which has a particular location and value are called picture elements, image elements, pels, and pixels.

Digital Image Processing helps to improve images for human interpretation. Information can be processed and extracted from images for machine interpretation. The pixels in the image can be manipulated to any desired density and contrast. Images can be stored and retrieved easily.

Image processing has an enormous range of applications almost every area of science and technology can make use of image processing methods.

1. Medicine
    - Inspection and interpretation of images obtained from X-rays, MRI or CAT scans,
    - analysis of cell images, of chromosome karyotypes.
2. Agriculture
    - Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops.
    - inspection of fruit and vegetables—distinguishing good and fresh produce from old.
3. Industry
    - Automatic inspection of items on a production line,
    - inspection of paper samples.
4. Law enforcement
    - Fingerprint analysis,
    - sharpening or de-blurring of speed-camera images.
5. Intelligent Transportation Systems
6. Remote Sensing
7. Moving a tracking objectSecurity monitoring
8. Automatic Testing Program

3.Matlab Software Introduction

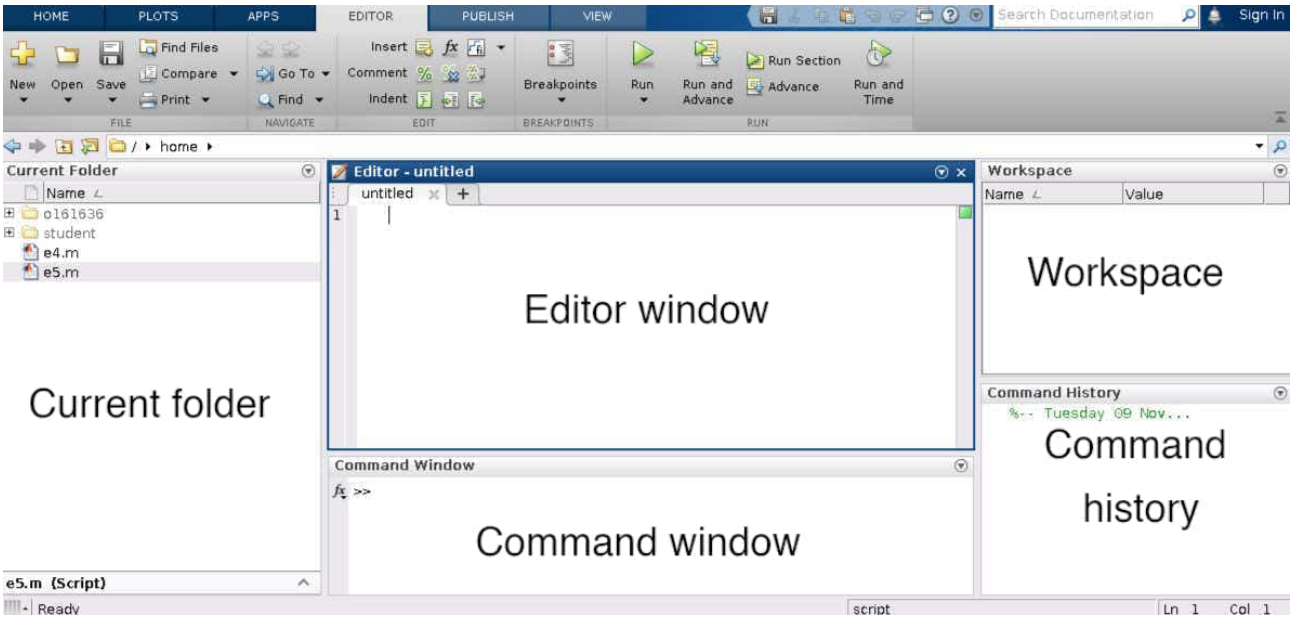
What is Matlab?

Matlab is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

Matlab is also called as Matrix Laboratory.“MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++ and Fortran.”MATLAB is an interactive, interpreted language that is designed for fast numerical matrix calculations.

Matlab is a data analysis and visualization tool which has been designed with powerful support for matrices and matrix operations. As well as this, Matlab has excellent graphics capabilities,and its own powerful programming language. One of the reasons that Matlab has become suchan important tool is through the use of sets of Matlab programs designed to support a particular task.

Matlab software looks like this.



Command Window:-

The Command Window enables you to enter individual statements at the command line and view the generated results.

Editor window:-

The matlab Editor Window is a simple text editor where you can load, edit and save complete MATLAB programs. The Editor window also has a menu command (Debug/Run) which allows you to submit the program to the command window. ... It also has a number of example programs and tutorials.

Workspace:-

The workspace contains variables that you create or import into MATLAB from data files or other programs. You can view and edit the contents of the workspace in the Workspace browser or in the Command Window. Workspace variables do not persist after you exit MATLAB. To use your data across multiple sessions, save it to a

compressed file with a .mat extension called a MAT-file. You can restore saved data by loading a MAT-file back into MATLAB.

Command History

The Command History window displays a log of statements that you ran in the current and previous MATLAB sessions. The Command History lists the time and date of each session in the short date format for your operating system, followed by the statements from that session. Brackets in the left margin indicate statements that were processed as a group. A colored mark precedes each statement that generated an error.

Current folder

The Current Folder browser enables you to interactively manage files and folders in MATLAB. Use the Current Folder browser to view, create, open, move, and rename files and folders in the current folder.

Variable Names

Valid Names

A valid variable name starts with a letter, followed by letters, digits, or underscores. MATLAB is case sensitive, so A and a are *not* the same variable. The maximum length of a variable name is the value that the namelengthmax command returns.

You cannot define variables with the same names as MATLAB keywords, such as if or end. For a complete list, run the iskeyword command.

Examples of valid names:	Examples of invalid names:
x6	6x
lastValue	end
n_factorial	n!

Matlab help

doc

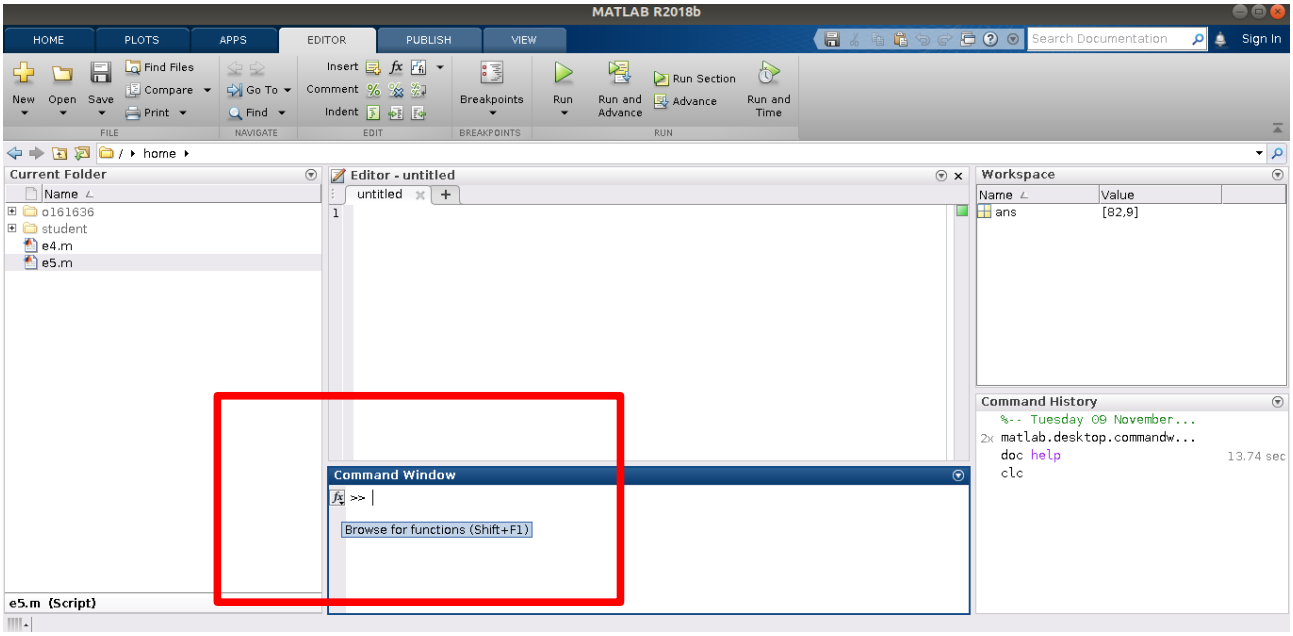
doc opens the Help browser. If the Help browser is already open, but not visible, then doc brings it to the foreground and opens a new tab.

Example

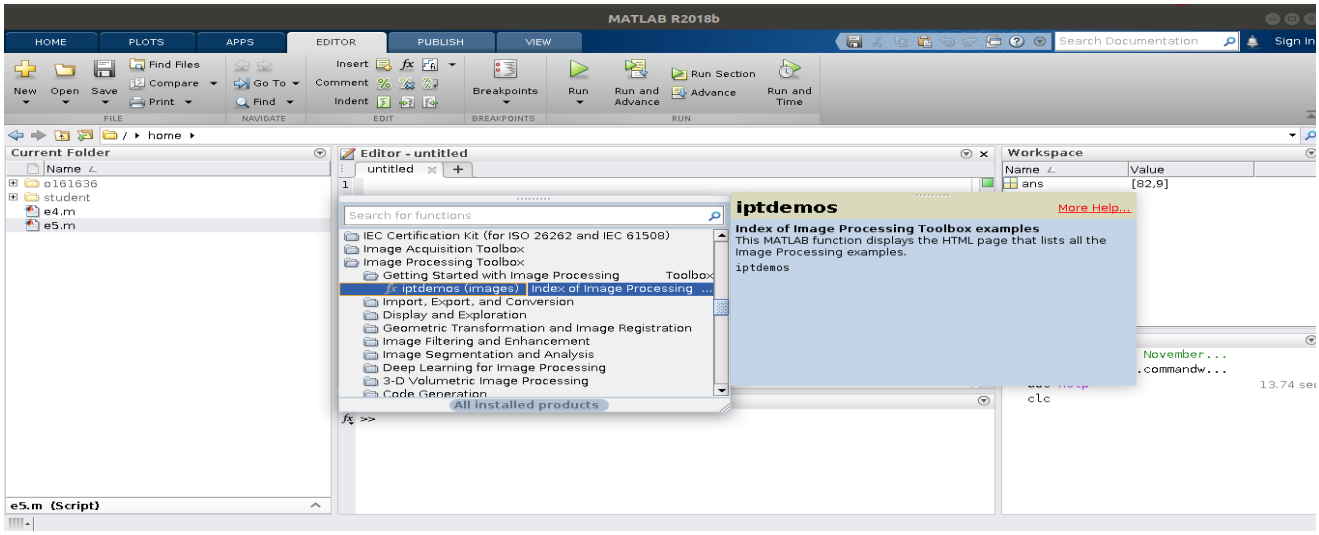
doc name displays documentation for the functionality specified by name, such as a function, class, or block.

- If name corresponds to a MathWorks reference page, then doc displays the page in the Help browser. The doc command does not display third-party or custom HTML documentation.
- If name does not correspond to a reference page, then doc searches for help text in a file named *name*.m or name.mlx. If help text is available, doc displays it in the Help browser.
- If name does not correspond to a reference page and there is no associated help text, then doc searches the documentation for name and displays the search results in the Help browser.

Search functions in matlab



You can see fx in the red rectangle box.This is used to search functions in matlab.When we click on it the following will be seen.



In this way we can access function details using this feature.

Help command

help name displays the help text for the functionality specified by name, such as a function, method, class, toolbox, or variable.

Functions used:-

slno	Function Name	Syntax	Description	Example
1	**Imread ( )**	A = imread(filename)	A = imread(filename) reads the image from the file specified by filename.	image=imread('**lion.jpeg**');
2	**Imrotate( )**	J = imrotate(I,angle);	J = imrotate(I,angle) rotates image I by angle degrees in a counterclockwise direction around its center point. To rotate the image clockwise, specify a negative value for angle.	zr=imrotate(image,45);
3	**Imtranslate( )**	B = imtranslate(A,translation)	B = imtranslate(A,translation) translates image A by the translation vector specified in translation.	t=imtranslate(image, [50,50]);
4	**Subplot( )**	subplot(m,n,p)	subplot(m,n,p) divides the current figure into an m-by-n grid and creates axes in the position specified by p. MATLAB® numbers subplot positions by row.	subplot(2,3,1)
5	**Sgtitle( )**	sgtitle(txt)	sgtitle(txt) adds a title above the grid of subplots in the current figure. If a figure does not it exist, then this command creates one.	sgtitle("Geometric transformations of image");
6	**Imshow()**	imshow(B)	imshow(I) displays the grayscale image I in a figure. imshow optimizes figure, axes, and image object properties for image display.	imshow(image);
7	**Title( )**	title(txt)	title(txt) adds the specified title to the axes or chart returned by the gca command. Reissuing the title command causes the new title to replace the old title.	**title(**"Geometric transformations of image");
8	**Affine2d( )**	tform = affine2d(A);	tform = affine2d(A) sets the property T with a valid affine transformation defined by nonsingular matrix A.	tform=affine2d([1 0 0;0 1 0;0 0 1;]);
9	**Imwrap( )**	B = imwarp(A,tform);	A = imwarp(A,tform) transforms the image A according to the geometric transformation defined by tform, which is a geometric transformation object. B is the transformed image.	k1=imwarp(image,tform1);
10	**Rgb2gray( )**	I = rgb2gray(RGB)	I = rgb2gray(RGB) converts the truecolor image RGB to the grayscale intensity image I. The rgb2gray	b=rgb2gray(a);

			function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.	
11	**Im2bw( )**	BW=im2bw(I,level);	BW = im2bw(I,level) converts the grayscale image I to binary image BW, by replacing all pixels in the input image with luminance greater than level with the value 1 (white) and replacing all other pixels with the value 0 (black).	im2bw(a);
12	**Imadd( )**	Z = imadd(X,Y)	Z = 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.	b=imadd(a,50);
13	**Imsubtract( )**	Z = imsubtract(X,Y)	z = 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.	b=imsubtract(a,50);
14	**Imcomplement( )**	J = imcomplement(Y)	J = imcomplement(I) computes the complement of the image I and returns the result in J.	b=imcomlement(a);
15	**Imdivide( )**	Z = imdivide(X,Y)	Z = 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.	b=imdivide(a,5);
16	**Immultiply( )**	Z = immultiply(X,Y)	Z = 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.	i=immultiply(a,2)
17	**Imgaussfilt( )**	B = imgaussfilt(A,sigma)	B = imgaussfilt(A,sigma) filters image A with a 2-D Gaussian smoothing kernel with standard deviation specified by sigma.	i=imgaussfilt(a,3)
18	I**msharpen( )**	B = imsharpen(A)	B = imsharpen(A) sharpens the grayscale or truecolor (RGB) input image A by using the unsharp masking method.	b=imsharpen(image)

19	**Histogram( )**	histogram(X)	histogram(X) creates a histogram plot of X.	histogram(image)
20	**Histeq( )**	J = histeq(I,hgram)	Enhance contrast using histogram equalization.J = histeq(I,hgram) transforms the grayscale image I so that the histogram of the output grayscale image J with length(hgram) bins approximately matches the target histogram hgram.	a=histeq(image)
21	**Imhistmatch( )**	J = imhistmatch(I,ref)	Adjust histogram of 2-D image to match histogram of reference image.J = imhistmatch(I,ref) transforms the 2-D grayscale or truecolor image I returning output image J whose histogram approximately matches the histogram of the reference image ref.If both I and ref are truecolor RGB images, then imhistmatch matches each color channel of I independently to the corresponding color channel of ref.	k=imhistmatch(im1,im2)
22	**Double( )**	Y = double(X)	Double-precision arrays.Y = double(X) converts the values in X to double precision	a=double(x);
23	**Mod( )**	b = mod(a,m)	Remainder after division (modulo operation).b = mod(a,m) returns the remainder after division of a by m, where a is the dividend and m is the divisor.	mod(c1,2)
24	**Floor( )**	Y = floor(X)	Round toward negative infinity Y = floor(X) rounds each element of X to the nearest integer less than or equal to that element.	floor(c1,2);
25	**Uint8( )**	Y = uint8(X)	Y = uint8(X) converts the values in X to type uint8. Values outside the range $[0,2^8-1]$ map to the nearest endpoint.	uint8(image)
26	**Imsplit( )**	[c1,c2,c3,...,ck] = imsplit(I)	Split multichannel image into its individual channels.[c1,c2,c3,...,ck] = imsplit(I) returns a set of *k* images representing the individual channels in the *k*-channel image I.	[r,g,b]=imsplit(im)

27	**Imnoise( )**	J = imnoise(I,'gaussian') J = imnoise(I,'salt & pepper')	Add noise to image.J = imnoise(I,'gaussian') adds zero-mean, Gaussian white noise with variance of 0.01 to grayscale image I.J = imnoise(I,'salt & pepper') adds salt and pepper noise, with default noise density 0.05. This affects approximately 5% of pixels.	k1=imnoise(image,'gaussian'); k2=imnoise(image,'salt & pepper')
28	**Medfilt2( )**	J = medfilt2(I,[m n])	J = medfilt2(I,[m n]) performs median filtering, where each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image.	j=medfilt2(image,[3,3]);
29	**Wiener2( )**	J = wiener2(I,[m n],noise)	J = wiener2(I,[m n],noise) filters the grayscale image I using a pixel-wise adaptive low-pass Wiener filter. [m n] specifies the size (m-by-n) of the neighborhood used to estimate the local image mean and standard deviation. The additive noise (Gaussian white noise) power is assumed to be noise.	k=wiener2(i,[3,3]);
30	**figure**	figure;	Create a new figure	figure;
31	**Imfuse( )**	C = imfuse(A,B)	C = imfuse(A,B) creates a composite image from two images, A and B.	imfuse(image1,image2);
32	**Edge( )**	BW = edge(I,method)	BW = edge(I,method) detects edges in image I using the edge-detection algorithm specified by method.	i=edge(i,'canny')
33	**Strel( )**	SE = strel('line',len,deg)	Morphological structuring element.A strel object represents a flat morphological *structuring element*, which is an essential part of morphological dilation and erosion operations.SE = strel('line',len,deg) creates a linear structuring element that is symmetric with respect to the neighborhood center. deg specifies the angle (in degrees) of the line as measured in a counterclockwise direction from the horizontal axis. len is approximately the distance between the centers of the structuring element members at opposite ends of the line	se=strel('line',11,90);
34	**Imerode( )**	J = imerode(I,SE)	J = imerode(I,SE) erodes the grayscale,	z=imerode(y,se);

			binary, or packed binary image I, returning the eroded image, J. SE is a structuring element object or array of structuring element objects, returned by the strel or offsetstrel functions.	
35	**Imdilate( )**	J = imdilate(I,SE)	J = imdilate(I,SE) dilates the grayscale, binary, or packed binary image I,returning the dilated image, J. SE is astructuring element object or array of structuring element objects, returned by the strel or offsetstrel functions.	d=imdilate(y,se);
36	**Imresize( )**	J = imresize(I,scale)	J = imresize(I,scale) returns image J that is scale times the size of grayscale, RGB, or binary image I. If I has more than two dimensions, then imresize only resizes the first two dimensions. By default, imresize uses bicubic interpolation and performs antialiasing.	j=imresize(img,[256,256]);
37	**Imwrite( )**	imwrite(A,filename)	imwrite(A,filename) writes image data A to the file specified by filename, inferring the file format from the extension. imwrite creates the new file in your current folder.	imwrite(Xc,"testc.jpg");
38	**Wcompress( )**	wcompress('c',X,SAV_FILENAME, COMP_METHOD) XC = wcompress('u',SAV_FILENAME)	True compression of images using waveletsThe wcompress command performs either compression or uncompression of grayscale or truecolor images.	[cr,bpp] = wcompress('c',X,'wpeppers.wtc','spiht','maxloop',12);Xc = wcompress('u','wpeppers.wtc')
39	**Delete( )**	delete(obj)	Deletes files or objects.delete(obj) deletes the specified object. If obj is an array, then delete deletes all objects in the array. obj remains in the workspace, but is no longer valid.	delete('wpeppers.wtc')
40	**Imfinfo( )**	info = imfinfo(filename)	info = imfinfo(filename) returns a structure whose fields contain information about an image in a graphics file, filename.	imfinfo(img)

4.Image Conversions

Theory:-

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image.

Convert between the image types, such as RGB (truecolor), binary, grayscale, and indexed images, and change the data type of an image. Image Processing Toolbox Supports binary, indexed grayscale, and truecolor image types. In each image type, pixels are stored in different formats.

- The Image Converter is a simple tool that provides a quick way to perform basic image conversion tasks
- It can convert from any image format Opus understands to BMP, GIF, PNG and JPG formats.
- It can rotate images in 90 degree increments, and also automatically rotate to compensate for EXIF orientation.
- It can resize or enlarge images, by a percentage or to an absolute size, optionally retaining the original aspect ratio.

Original Image

Gray Image

Brightness Image

Binary Image

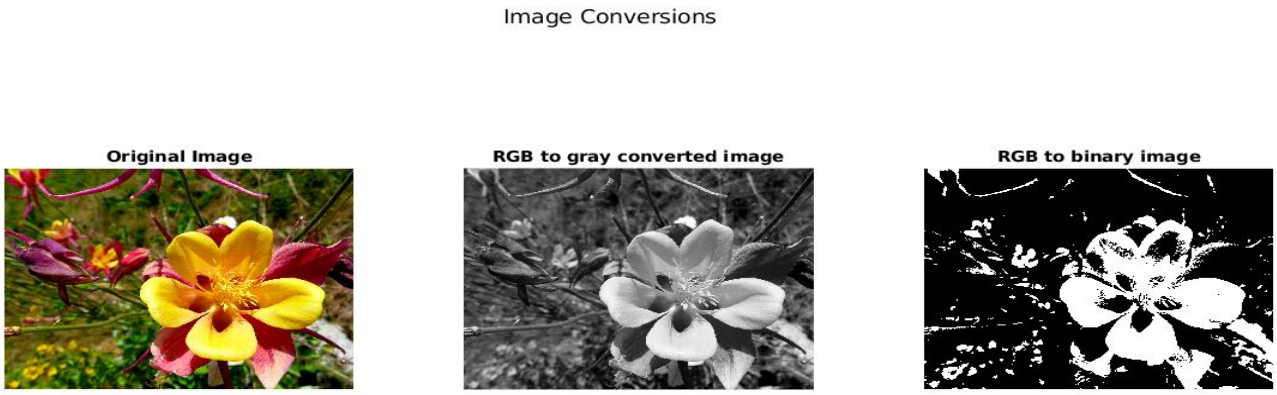
Code:-

```
clc;
close all;
clear;
a=imread('flower.jpg');
sgtitle("Image Conversions");
subplot(1,3,1);imshow(a);title("Original Image");
b=rgb2gray(a);
subplot(1,3,2);imshow(b);title("RGB to gray converted image");
c=im2bw(a);
subplot(1,3,3);imshow(c);title("RGB to binary image");
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.A image is read into the variable ‘a’ and then we convert the rgb image into grayscale image and gray scale image into binary and we display them.

Output:-



Result:-

For many applications of image processing, color information doesn't help us identify important edges or other features. There are exceptions.. Binary images are important when we wish to reduce the amount of information in the image and focus only on regions of the image we need.So we need gray scale images sometimes and sometimes we need rgb image and sometimes binary image.Hence,We have successfully converted RGB images to gray scale images and gray scale images to binary images.

5. Image Arithmetic

Theory:

There are two primary categories of algebraic operations applied to image:

(i) Arithmetic operations.

(ii) Logic operations.

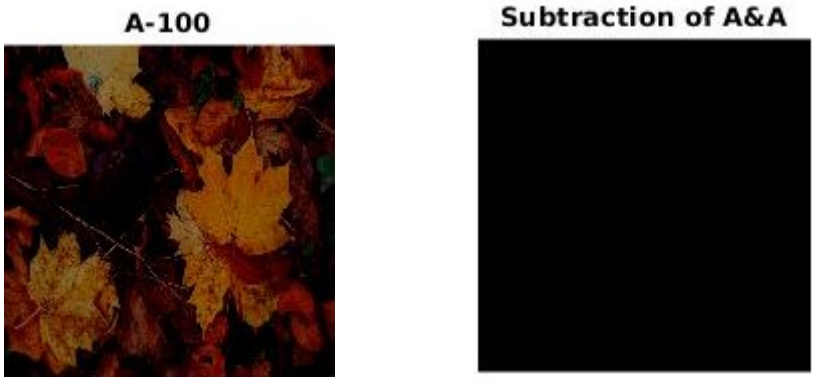
(1) Arithmetic Operations: Addition, subtraction, division and multiplications comprise the arithmetic operations.

(2) Logical Operations: AND, OR and NOT makeup the logic operations.

- These operations which require only one image, and are done on a pixel –by-pixel basis.
- To apply the arithmetic operations to two images, we simply operate on corresponding pixel values.

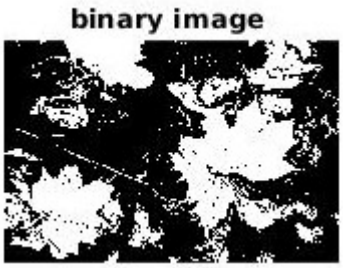
(i) <u>Addition</u>: Addition is used to combine the information in two images. Applications include development of image restoration algorithm for molding additive noise, and special effects, such as image morphing in motion pictures.Lets see an example.

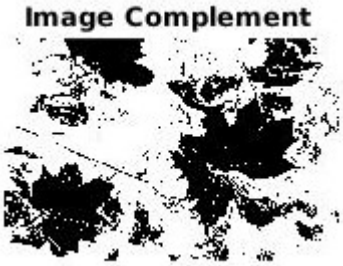
(ii) <u>Subtraction</u>: Subtraction of two images is often used to **detect motion**, consider the case where nothing has changed in a sense; the image resulting from subtraction of two sequential image is filled with zero-a black image. If something has moved in the scene, subtraction produces a nonzero result at the location of movement. Applications include Object tracking , Medical imaging, Law enforcement and Military applications. Lets see an example.



(iii) <u>Multiplication and Division</u> : Multiplication and Division are used **to adjust the brightness of an image**. One image typically consists of a constant number greater than one. Multiplication of the pixel values by a number greater than one.

(iv): <u>Binary image and Image Complement</u> :





Code:

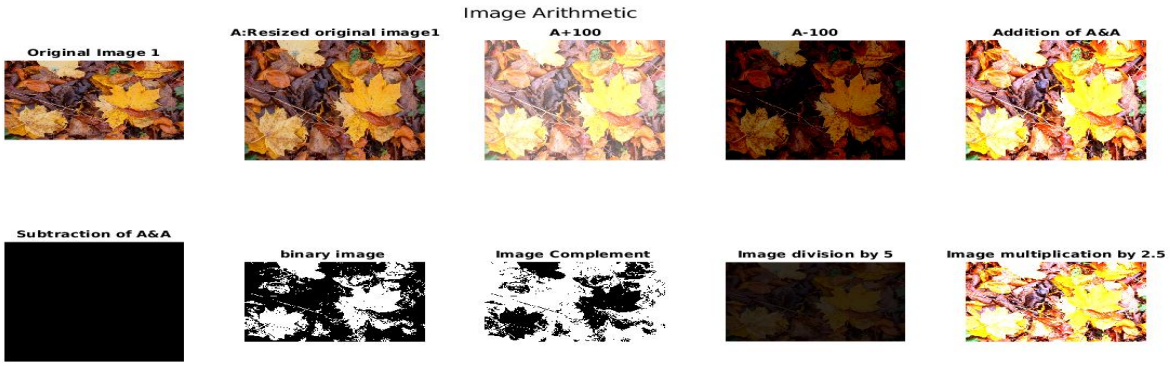
```
clc;
close all;
clear all;
a=imread('leaves.jpg');
b=imresize(a,[500,500]);
con=im2bw(a);
x=imadd(b,100);
y=imsubtract(b,100);
e=imadd(b,b);
f=imsubtract(b,b);
g=imcomplement(con);
```

```
h=imdivide(a,5);
i=immultiply(a,2.5);
sgtitle("Image Arithmetic")
subplot(2,5,1);imshow(a);title("Original Image 1");
subplot(2,5,2);imshow(b);title("A:Resized original image1");
subplot(2,5,3);imshow(x);title('A+100');
subplot(2,5,4);imshow(y);title('A-100');
subplot(2,5,5);imshow(e);title("Addition of A&A");
subplot(2,5,6);imshow(f);title("Subtraction of A&A");
subplot(2,5,7);imshow(con);title("binary image ");
subplot(2,5,8);imshow(g);title("Image Complement ");
subplot(2,5,9);imshow(h);title("Image division by 5");
subplot(2,5,10);imshow(i);title("Image multiplication by 2.5");
```

Code Explanation:

clc, clear all, close all are the statements which will clear command window, workspace and figures respectively. An image is read into the variable image and resized. After that, Arithmetic operations happen like Addition, Subtraction, Multiplication and Division by using functions like “imadd, imsub, imdiv, immul”.

Output:-



Result: Image subtraction can be used to detect differences between two or more images of the same scene or object.Image arithmetic is used to analyse an image in a convineint way these helps in human interpretation.When we add a costant to an image then image brightens according the value,when we subtract a constant from an image then image brightness,contrast decreases,Image compliment is helpful to analyse the images related to diseases.etc.Hence we have successfully completed some of image arithmetic concepts.

6.Geometric Transformations of images

Theory:-

Geometric transformations modify the spatial arrangement of pixels in an image. These transformations are called rubber-sheet transformations.Geometric transfor-mations of digital images consist of two basic operations:

- Spatial transformation of coordinates.
- intensity interpolation that assigns intensity values to the spatially transformed pixels.

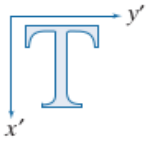
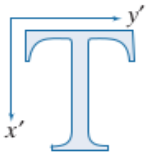
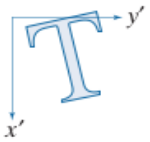
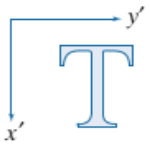
The transformation of coordinates may be expressed as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

where (x , y) are pixel coordinates in the original image and (x ′ , y ′) are the corresponding pixel coordinates of the transformed image.
Affine transformations, which include scaling, translation,rotation, and shearing. The key characteristic of an affine transformation in 2-D is that it preserves points, straight lines, and planes.However, it is possible to use homogeneous coordinates to express all four affine transformations using a single 3 × 3 matrix in the following general form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

This transformation can scale, rotate, translate, or sheer an image, depending on the values chosen for the elements of matrix A.

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to –1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	

Code:-

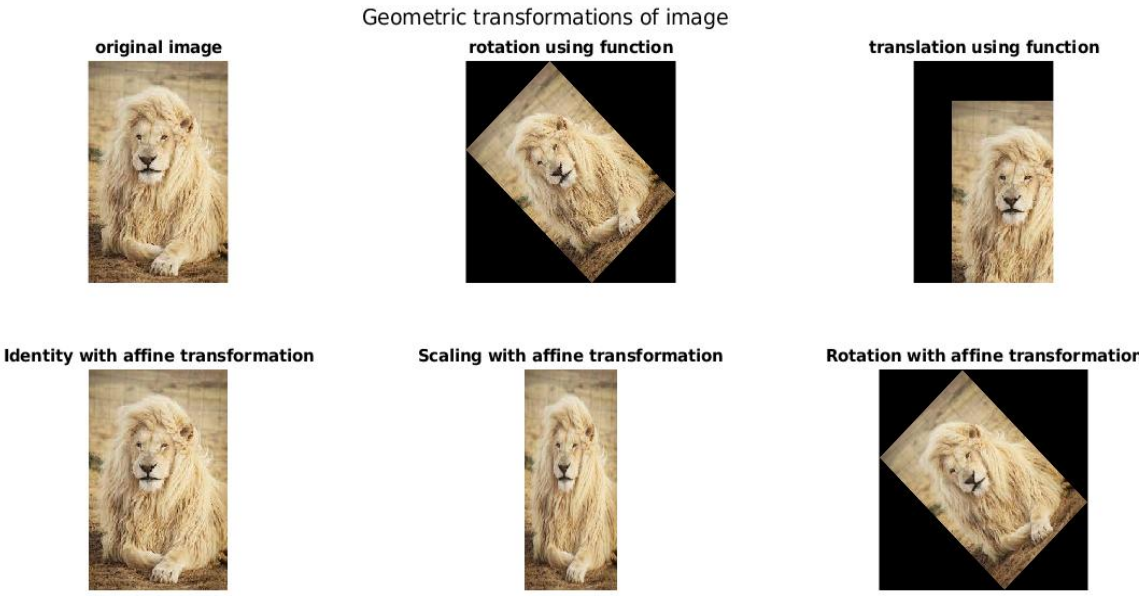
```
clc;
clear all;
close all;
image=imread('lion.jpeg');
r=imrotate(image,45);
t=imtranslate(image,[50,50]);
sgtitle("Geometric transformations of image");
subplot(2,3,1);imshow(image);title('original image');
subplot(2,3,2);imshow(r);title('rotation using function')
subplot(2,3,3);imshow(t);title('translation using function')
tform=affine2d([1 0 0;0 1 0;0 0 1;]);
tform1=affine2d([2 0 0;0 3 0;0 0 1;]);
tform2=affine2d([cosd(45) -sind(45) 0;sind(45) cosd(45) 0;0 0 1;]);
k=imwarp(image,tform);
k1=imwarp(image,tform1);
k2=imwarp(image,tform2);
subplot(2,3,4);imshow(k);title('Identity with affine transformation');
subplot(2,3,5);imshow(k1);title('Scaling with affine transformation');
```

```
subplot(2,3,6);imshow(k2);title('Rotation with affine transformation');
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.A image is read into the variable image and resized,then the image is rotated and translated using functions first.We have also used affine transformations to rotate and scale the input image by using imwarp function and then the image shown using imshow function.

Output:-



Result:-

The affine transformation technique is typically used to correct for geometric distortions or deformations that occur with non-ideal camera angles.Geometric transformations in image like rotation ,translation,scaling are useful in setting the images that are defected or deformed at image aquisition state itself.Hence,we have successfully executed some of geometric transformations in an image.

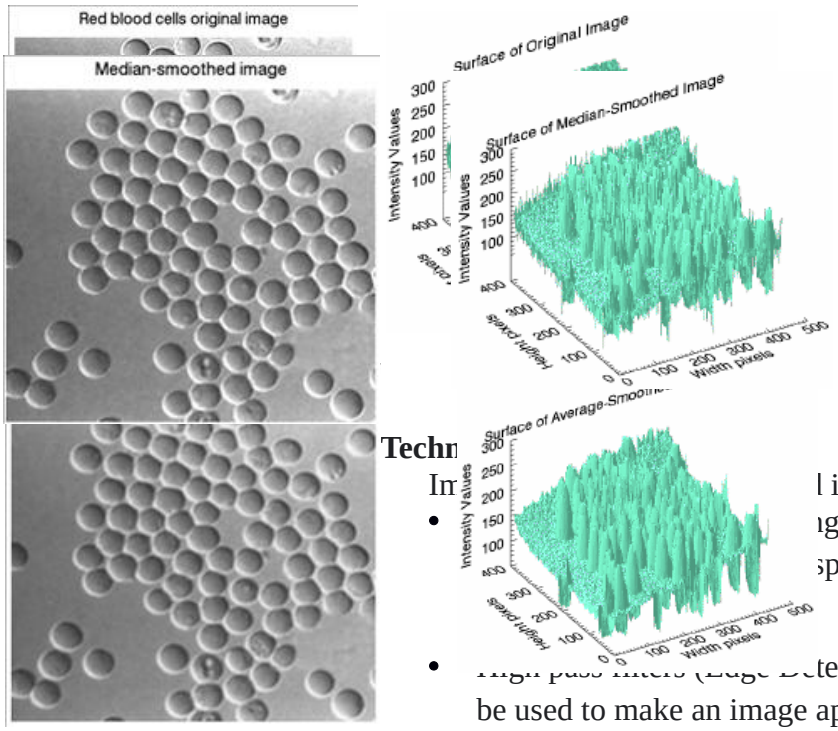
7.Image smoothing & sharpening

Theory:-

Image smoothing: Smoothing is used to reduce noise or to produce a less pixelated image. Most smoothing methods are based on low-pass filters, but you can also smooth an image using an average or median value of a group of pixels (a kernel) that moves through the image.

- Smoothing filters are used for noise reduction and blurring operations.
- It takes into account the poxels surrounding it in order to make a determination of a more accurate version of this pixel.
- By taking neighboring pixels into consideration, extreme “noisy” picels can be filtered out.

- Unfortunately,extreme pixels can also represent original fine details, which can also be lost due to the smoothing process.



Techn

Im l in two depending on the effects:

- ıg) Low pass filtering (aka smoothing), is spatial frequency noise from a digital
- tection, Sharpening) A high-pass filter can be used to make an image appear sharper.

Image sharpening:

Image sharpening refers to any enhancement technique that highlights edges and fine details in an image. Image sharpening is widely used in printing and photographic industries for increasing the local contrast and sharpening the images.

Image sharpening encompasses any enhancement technique that highlights the edges and fine details of an image. Image sharpening is done by adding to the original image a signal proportional to a high-pass filtered version of the image.

Code:-

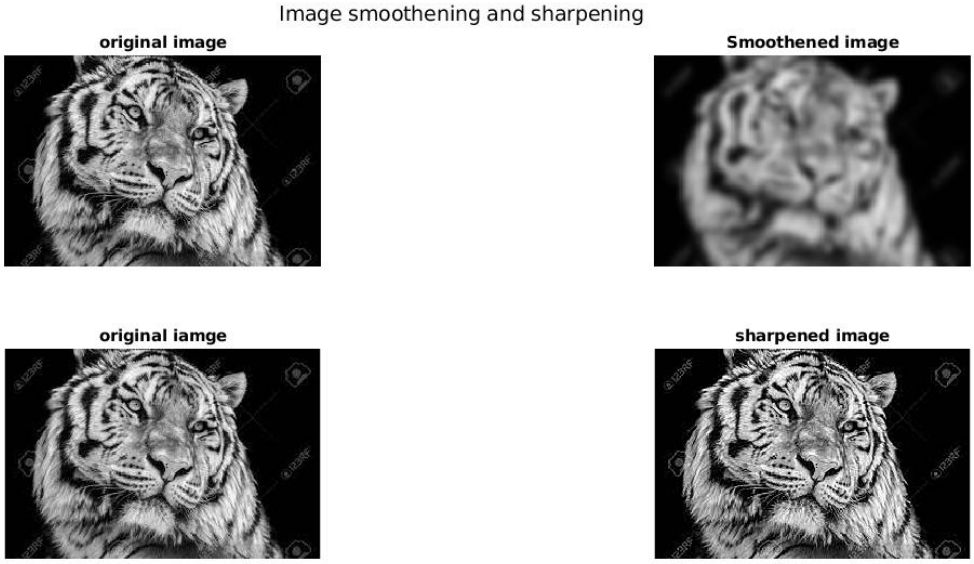
```
clc;close all;clear all;

im1=imread("hc.jpeg");

ismooth=imgaussfilt(im1,3);

isharp=imsharpen(im1);

sgtitle("Image smoothening and sharpening");

subplot(2,2,1);imshow(im1);title("original image");

subplot(2,2,2);imshow(ismooth);title("Smoothened image");

subplot(2,2,3);imshow(im1);title("original image");
```

```
subplot(2,2,4);imshow(isharp);title("sharpened image");
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.A image is read into the variable im1 then we have applies smoothening and sharpening functions for image and then the outputs are displayed using figures.

Output:-



Result:-Smoothing is used to reduce noise or to produce a less pixelated image. Most smoothing methods are based on low-pass filters.sharpening filters to accentuate the edges of structures.Hence,we have successfully executed image smoothening and image sharpening.

8.Image Blurring

Theory:-

In blurring, we simple blur an image. An image looks more sharp or more detailed if we are able to perceive all the objects and their shapes correctly in it. For example. An image with a face, looks clear when we are able to identify eyes, ears, nose, lips, forehead e.t.c very clear. This shape of an object is due to its edges. So in blurring, we simple reduce the edge content and makes the transition form one color to the other very smooth.

Blurring is an example of applying a low-pass filter to an image. In computer vision, the term “low-pass filter” applies to removing noise from an image while leaving the majority of the image intact. A blur is a very common operation we need to perform before other tasks such as edge detection.

Uses of blurring:

In blurring, we simple blur an image. An image looks more sharp or more detailed if we are able to perceive all the objects and their shapes correctly in it. ... This shape of an object is due to its edges. So in blurring, we simple reduce the edge content and makes the transition form one color to the other very smooth.

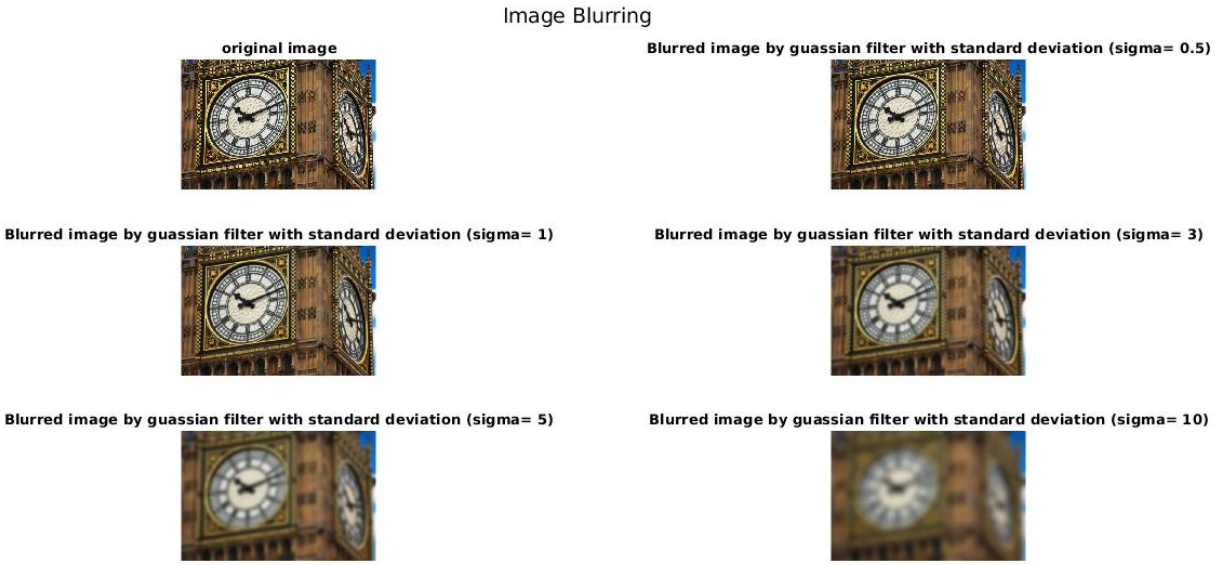
Code:-

```
clc;close all;clear all;

a=imread('clock.jpg');

k=imgaussfilt(a);

l=imgaussfilt(a,1);

m=imgaussfilt(a,3);

n=imgaussfilt(a,5);

o=imgaussfilt(a,10);

sgtitle(''Image Blurring'');

subplot(3,2,1);imshow(a);title('original image');

subplot(3,2,2);imshow(k);title('Blurred image by guassian filter with standard deviation (sigma= 0.5)');

subplot(3,2,3);imshow(l);title('Blurred image by guassian filter with standard deviation (sigma= 1)');

subplot(3,2,4);imshow(m);title('Blurred image by guassian filter with standard deviation (sigma= 3)');

subplot(3,2,5);imshow(n);title('Blurred image by guassian filter with standard deviation (sigma= 5)');

subplot(3,2,6);imshow(o);title('Blurred image by guassian filter with standard deviation (sigma= 10)');
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.An image is read and stored in a variable.We have applied gaussian function using different standard deviation values and these are applied for an image and its output was plotted.

Output:-



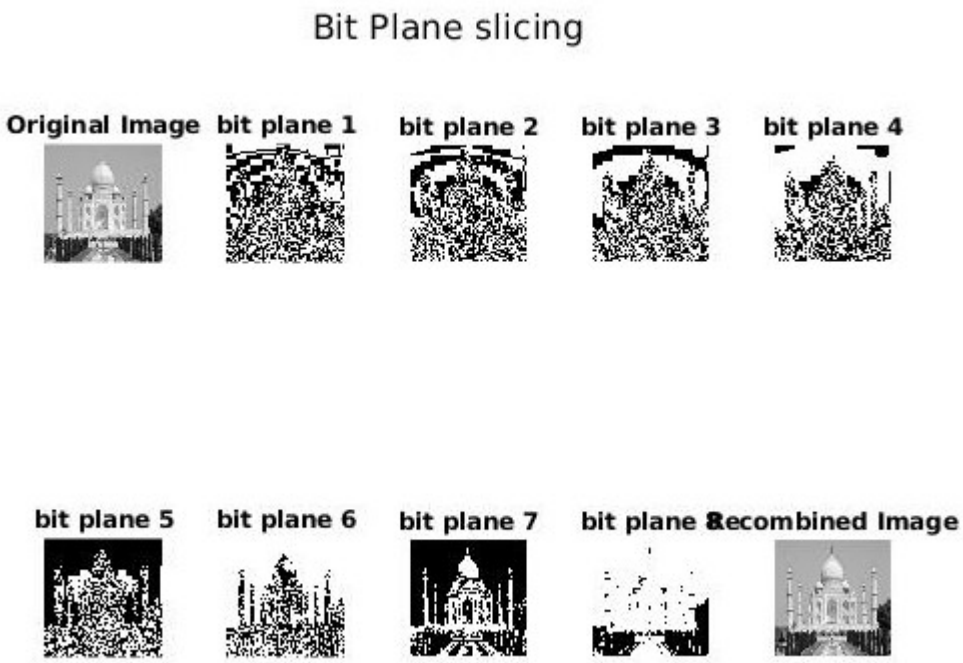
Result:-Applying a low-pass blurring filter smooths edges and removes noise from an image. Blurring is often used as a first step before we perform Thresholding, Edge Detection, or before we find the Contours of an image.Hence,we have successfully executed image blurring.

9. Bit Plane Slicing

Theory:-

It is nothing but to highlight the contribution made to the total image appearance by specific bits.

- Assuming that each pixel is represented bu 8 bits, the image is composed of 8:1 bit planes.
- Plane 1 contains the least significant bit and plane 8 contains the most significant bit.
- Useful for analyzing the relative importance played by each bit of the image.
- Only the higher order bits contain visually significance data. The other bit planes contribute the more subtle details.
- Plane 8 corresponds exactly with an image threshold at gray level 128.



Code:

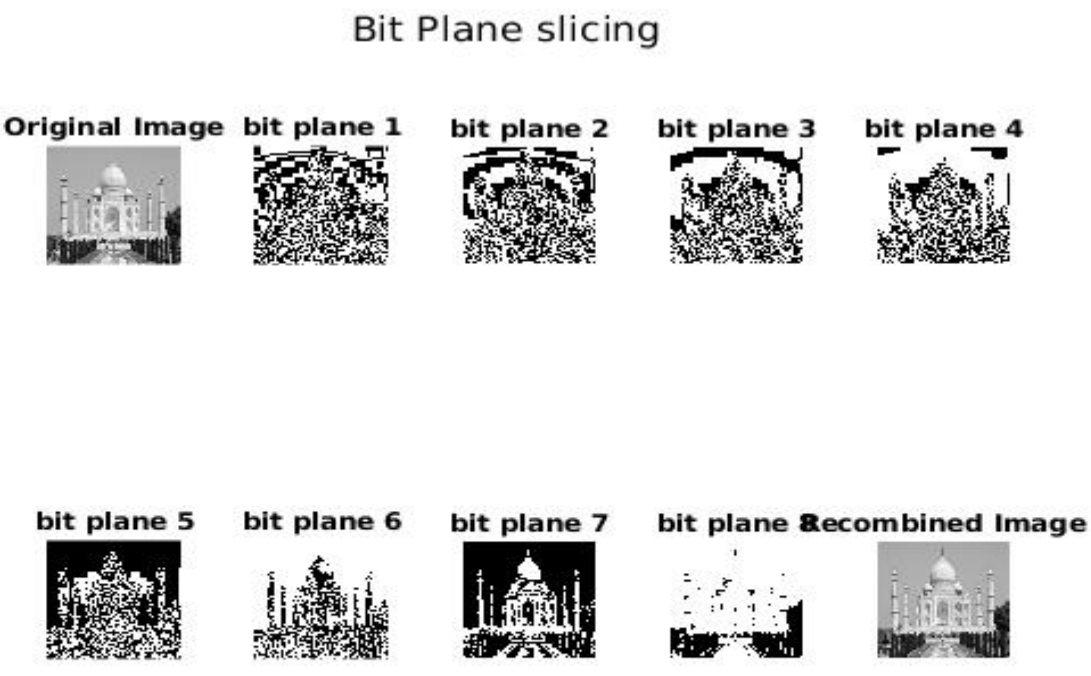
```
% clearing the output screen
clc;
clear all;
close all;
% reading image's pixel in c
c = imread('taj.jpeg');
c1=rgb2gray(c);
adjustc=imresize(c1,[256,256]);
% storing image information in cd
cd = double(adjustc);
% extracting all bit one by one
% from 1st to 8th in variable
% from c1 to c8 respectively
c1 = mod(cd, 2); c2 = mod(floor(cd/2), 2);
c3 = mod(floor(cd/4), 2); c4 = mod(floor(cd/8), 2);
c5 = mod(floor(cd/16), 2); c6 = mod(floor(cd/32), 2);
c7 = mod(floor(cd/64), 2); c8 = mod(floor(cd/128), 2);
% combining image again to form equivalent to original grayscale image
```

```
cc =  (2*(2*(2* (2*(2 * (2 * (2 * c8 + c7) + c6) +c5)+c4)+c3)+c2)+c1) ;
figure;
sgtitle("Bit Plane slicing");
subplot(2, 5, 1); imshow(adjustc); title('Original Image');
% plotting binary image having extracted bit from 1st to 8th
% in subplot from 2nd to 9th
subplot(2,5,2);imshow(c1); title('bit plane 1');
subplot(2,5,3);imshow(c2); title('bit plane 2');
subplot(2,5,4);imshow(c3);title('bit plane 3');
subplot(2,5,5);imshow(c4); title('bit plane 4');
subplot(2,5,6);imshow(c5); title('bit plane 5');
subplot(2,5,7);imshow(c6); title('bit plane 6');
subplot(2,5,8);imshow(c7); title('bit plane 7');
subplot(2,5,9);imshow(c8); title('bit plane 8');
% plotting recombined image in 10th subplot
subplot(2, 5, 10); imshow(uint8(cc)); title('Recombined Image');
```

Code Explanation:

clc, clear all, close all are the statements which will clear command window, workspace and figures respectively. A image is read into the variable image and resized. After that, By using some inbuilt functions we wrote the code".

Output:-

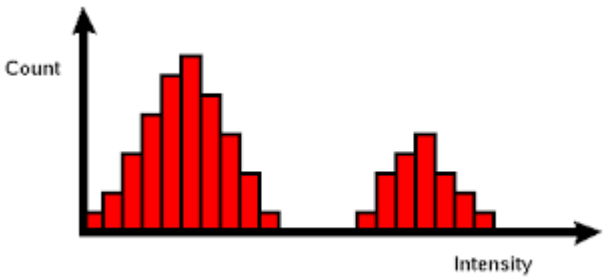


Result: Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, implying, it determines the adequacy of numbers of bits used to quantize each pixel , useful for image compression.Hence,we have successfully executed bit-plane slicing.

10.Histograms of different contrast images

Theory:-

A histogram is a graphical representation that organizes a group of data points into user-specified ranges. Similar in appearance to a bar graph, the histogram condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges or bins.



In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image.An image histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image.It plots the number of pixels for each value.
In digital image processing, the histogram is used for graphical representation of a digital image. A graph is a plot by the number of pixels for each tonal value. Nowadays, image histogram is present in digital cameras. Photographers use them to see the distribution of tones captured.

Applications of Histograms:

1. In digital image processing, histograms are used for simple calculations in software.
2. It is used to analyze an image. Properties of an image can be predicted by the detailed study of the histogram.
3. The brightness of the image can be adjusted by having the details of its histogram.
4. The contrast of the image can be adjusted according to the need by having details of the x-axis of a histogram.
5. It is used for image equalization. Gray level intensities are expanded along the x-axis to produce a high contrast image.
6. Histograms are used in thresholding as it improves the appearance of the image.
7. If we have input and output histogram of an image, we can determine which type of transformation is applied in the algorithm.

Contrast:

The term contrast refers to the amount of color or grayscale differentiation that exists between various image features in both analog and digital images. Images having a higher contrast level generally display a greater degree of color or grayscale variation than those of lower contrast.

Contrast refers to the amount of differentiation that is there between the various image features. Images having a higher contrast level generally display a greater degree of color or gray-scale variation than those of lower contrast.

Low contrast:

A low contrast image blends light and dark areas, creating a more flat or soft photo. There are hardly any highlights and shadows and the images are composed mostly in shades of gray. This dullness in the composition of lights and darks will mute the colors in the image.

High contrast:

The image histogram should span the entire dynamic range as shown above by the right histogram. ... Global means increasing the contrast of the whole image, While in local we divide the image into small regions and performcontrast enhancement on these regions independently.

. A low-contrast (left), medium-contrast (center), and high-contrast (right) version of the same photograph

The image in the center is direct from the scanned negative, and the only modification I made to create the other images was increasing or decreasing contrast.I could create high- and low-contrast versions that are taken to the extreme—i.e., one with contrast increased to the point where almost all of the pixels are completely black or completely white and the other with contrast so low that everything is washed out grayness. However, contrast doesn't work that way when we're trying to produce a realistic, visually pleasing image.

As contrast increases, the pixel values shift farther toward the left or right side of the histogram. In an image like this one, which emphasizes dark and light tones with relatively few midtone pixels, increasing the contrast makes the distribution more bimodal: the two peaks are moved farther apart and the "valley" in between the peaks becomes more pronounced.

Disadvantage :

The biggest disadvantage of this method is it does not preserve brightness of an image. The brightness get changed after histogram equalization. Hence preserving the original brightness and enhancing contrast are essential to avoid other side effects.

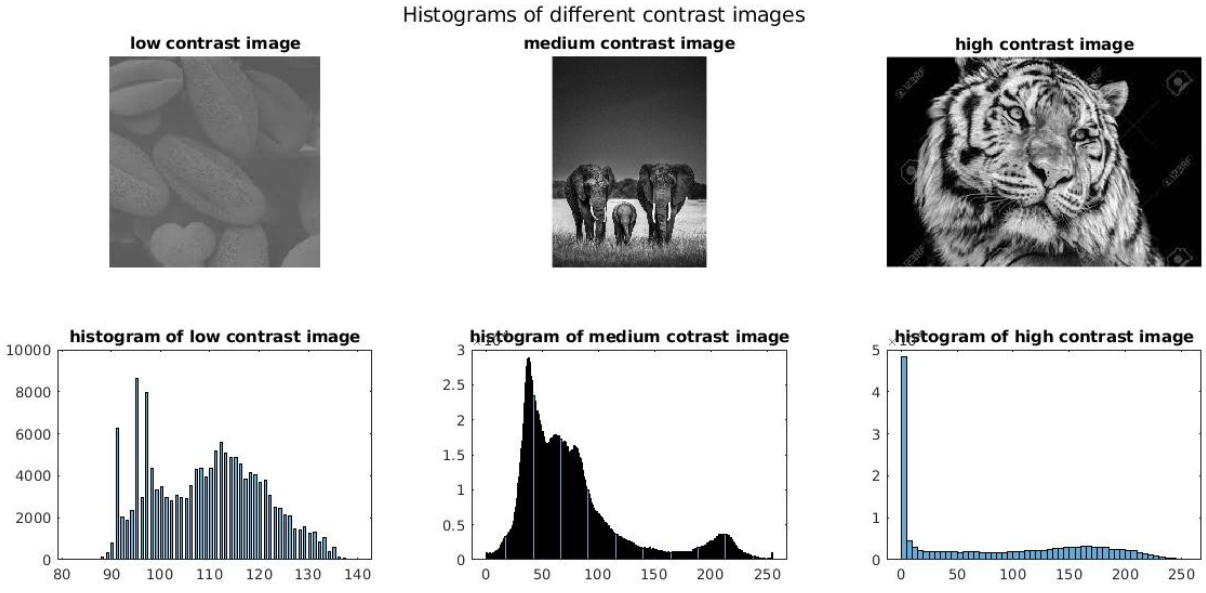
Code:-

```
clc;
close all;
clear all;
im1=imread("lc.jpeg");
im2=imread("mc.jpeg");
im3=imread("hc.jpeg");
sgtitle("Histograms of different contrast images");
subplot(2,3,1);imshow(im1);title("low contrast image");
subplot(2,3,2);imshow(im2);title("medium contrast image");
subplot(2,3,3);imshow(im3);title("high contrast image");
subplot(2,3,4);histogram(im1);title("histogram of low contrast image");
subplot(2,3,5);histogram(im2);title("histogram of medium cotrast image");
subplot(2,3,6);histogram(im3);title("histogram of high contrast image");
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.Three images of low,medium and high contrast images are read and their histograms are plotted.

Output:-



Result:--Histogram is used for graphical representation of a digital image.

It is used to analyze an image. Properties of an image can be predicted by the detailed study of the histogram.Hence,we have successfully executed plotting histograms of different contrast images.

11.Histogram Equalization

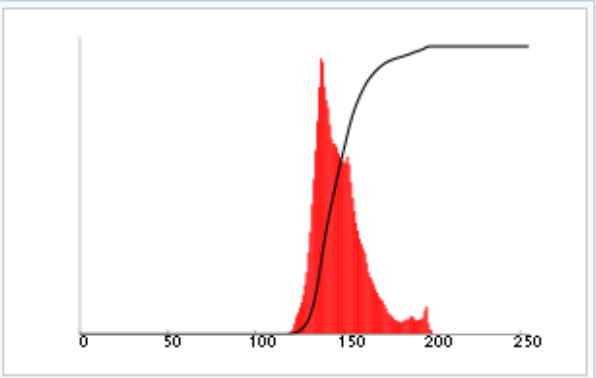
Theory:-

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. Histogram equalization is used to enhance contrast. It is not necessary that contrast will always be increase in this. There may be some cases were histogram equalization can be worse. In that cases the contrast is decreased.To enhance the image's contrast, it spreads out the most frequent pixel intensity values or stretches out the intensity range of the image. By accomplishing this, histogram equalization allows the image's areas with lower contrast to gain a higher contrast.Histogram Linearisation is also known as Histogram Equalisation.

Why Do You Use Histogram Equalization?

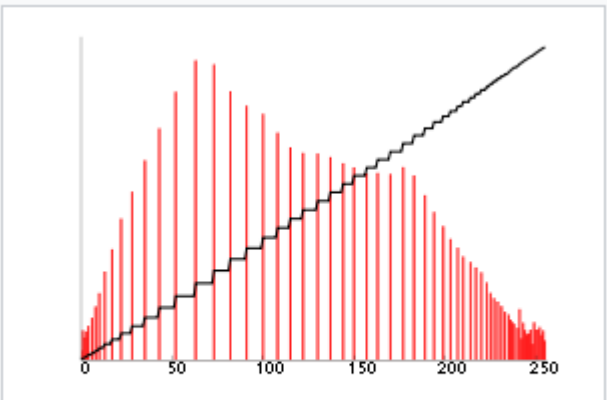
Histogram Equalization can be used when you have images that look washed out because they do not have sufficient contrast. In such photographs, the light and dark areas blend together creating a flatter image that lacks highlights and shadows.

Before Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)

After Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)

If you compare the two images above, you will find that the histogram equalized image has better contrast. It has areas that are darker as well as brighter than the original image.

Applications:

Image-processing technique often used to achieve better quality images in black and white color scales in medical applications such as digital X-rays, MRIs, and CT scans.

Disadvantages:

Histogram equalization cannot be applied separately to the Red, Green and Blue components of the image as it leads to dramatic changes in the image's color balance. A disadvantage of the method is that it is indiscriminate. It can generate only one type of output image.

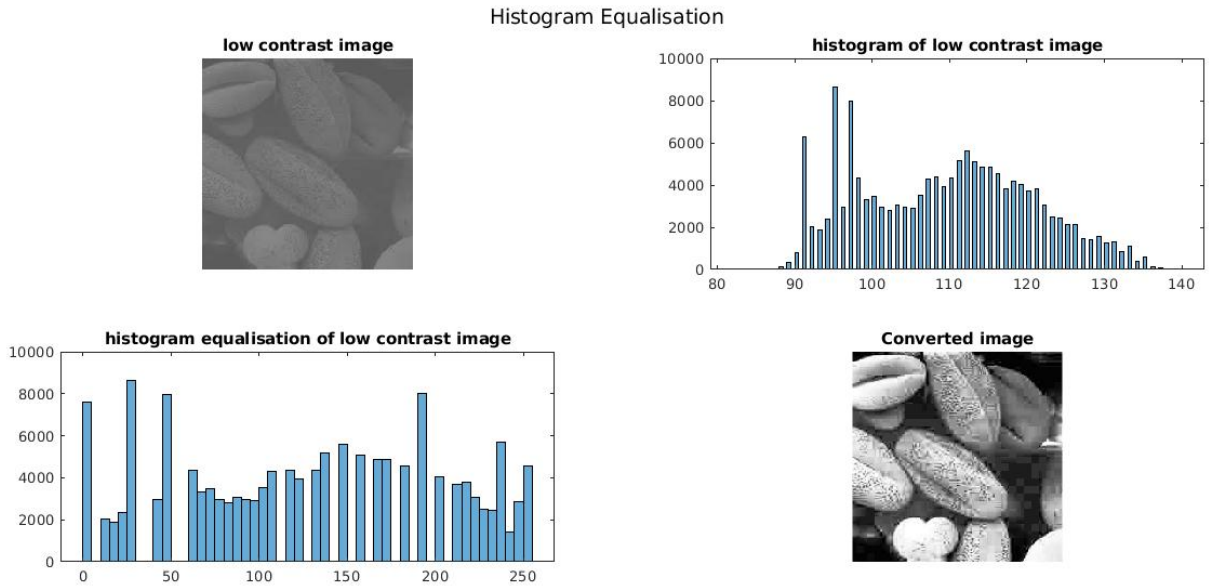
Code:

```
clc;close all;

clear all;

im1=imread("lc.jpeg");

J=histeq(im1);

sgtitle("Histogram Equalisation");

subplot(2,2,1);imshow(im1);title("low contrast image");

subplot(2,2,2);histogram(im1);title("histogram of low contrast image");

subplot(2,2,3);histogram(J);title("histogram equalisation of low contrast image");

subplot(2,2,4);imshow(J);title("Converted image");
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.A low contrast image is taken and its histogram is plotted and the histogram of input image is equalised using function and the equivalent image for output equalised histogram is plotted.

Output:-



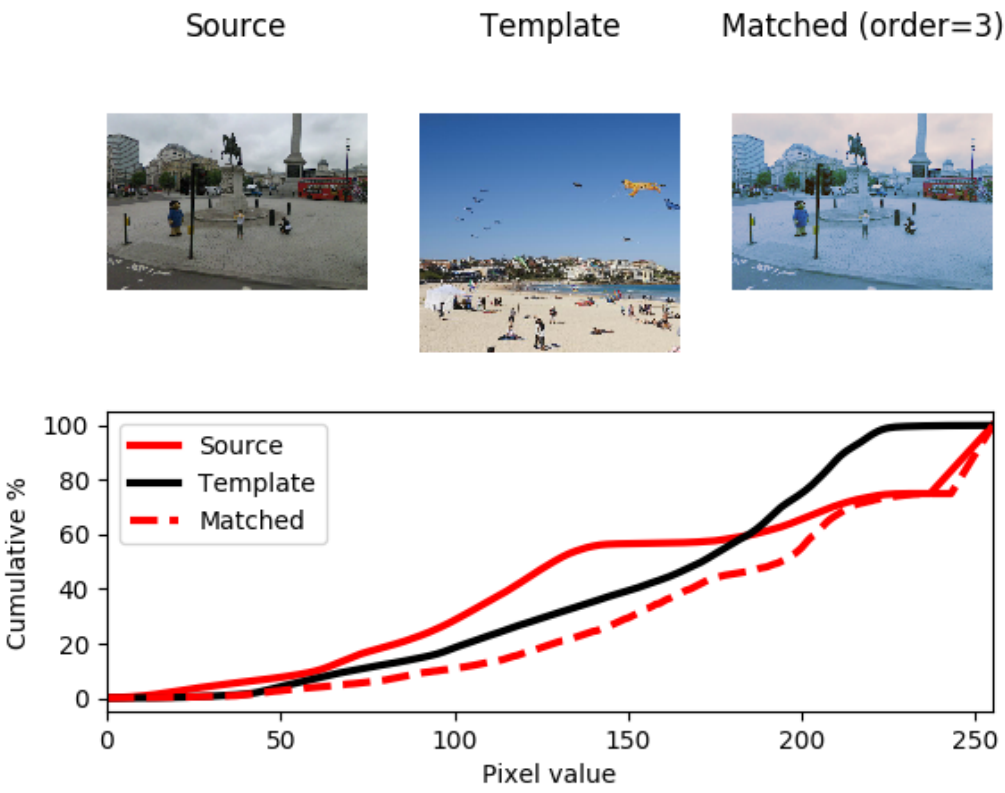
Result:-The brightness of the image can be adjusted by having the details of its histogram.The contrast of the image can be adjusted according to the need by having details of the x-axis of a histogram.It is used for image equalization. Histograms are used in thresholding as it improves the appearance of the image.Hence,we have successfully executed histogram equalization.

12.Histogram Matching (or) Histogram *Specification*

Theory:-

In image processing, histogram matching or histogram specification is the transformation of an image so that its histogram matches a specified histogram. The well-known histogram equalization method is a special case in which the specified histogram is uniformly distributed.

While the goal of histogram equalization is to produce an output image that has a flattened histogram, the goal of histogram matching is to take an input image and generate an output image that is based upon the shape of a specific (or reference) histogram.Let us suppose we have two images, an input image and a specified image. We want to use histogram matching to force the input image to have a histogram that is the shape of the histogram of the specified image.The process of Histogram Matching takes in an input image and produces an output image that is based upon a specified histogram. ... The transformation function is then applied to the input image to produce an output image by remapping the pixel intensities.In order to match the histogram of images A and B, we need to first equalize the histogram of both images. Then, we need to map each pixel of A to B using the equalized histograms. Then we modify each pixel of A based on B.



DISADVANTAGES:

A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

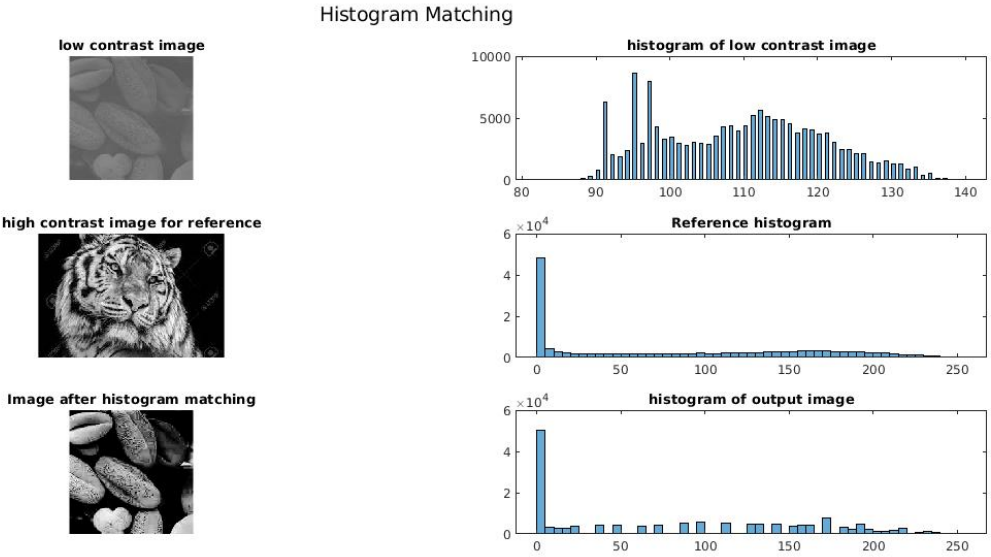
CODE:-

```
clc;close all;clear all;
im1=imread("lc.jpeg");
ref=imread("hc.jpeg");
J=imhistmatch(im1,ref);
sgtitle("Histogram Matching");
subplot(3,2,1);imshow(im1);title("low contrast image");
subplot(3,2,2);histogram(im1);title("histogram of low contrast image");
subplot(3,2,3);imshow(ref);title("high contrast image for reference");
subplot(3,2,4);histogram(ref);title("Reference histogram");
subplot(3,2,5);imshow(J);title("Image after histogram matching");
subplot(3,2,6);histogram(J);title("histogram of output image");
```

Code Explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.An image is read and a reference image is given.Histogram of reference image will be matched to the histogram of output image.Then the output images are plotted.

Output:-



Result:-The goal of histogram matching is to take an input image and generate an output image that is based upon the shape of a specific (or reference) histogram.Hence,we have successfully executed histogram matching concept.

13. RGB Components of an Image

ColorModel:

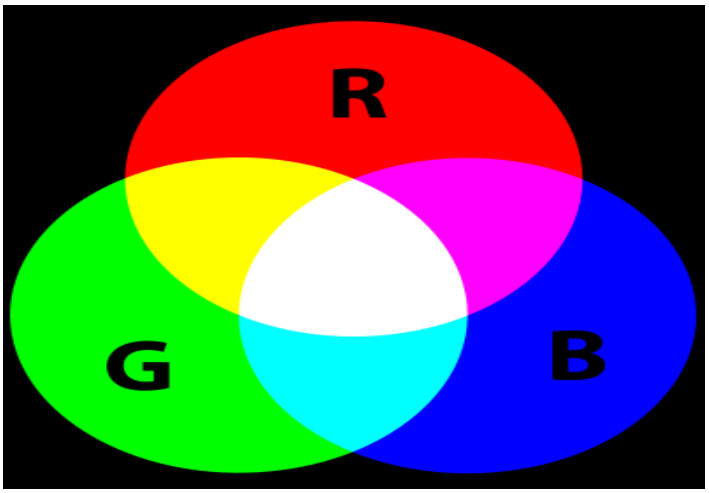
A color model is simply a way to define color. A model describes how color will appear on the computer screen or on paper. Three popular color models are:

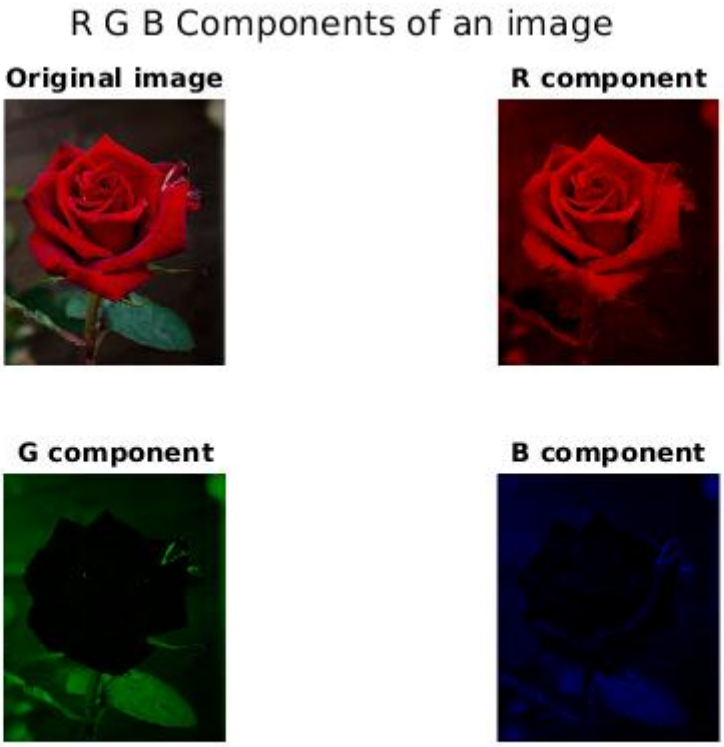
- **CMY and CMYK (Cyan, Magenta, Yellow, Black)**
- **RGB (Red, Green, Blue)**
- **HSI Color Model**

RGB Color Model:

The RGB model is used when working with screen based designs. A value between 0 and 255 is assigned to each of the light colors, Red, Green and Blue. So for example, if you

wanted to create a purely blue color, Red would have a value of 0, Green would have a value of 0 and Blue would have a value of 255 (pure blue). To create black, Red, Green and Blue would each have a value of 0 and to create white, each would have a value of 255. RGB is known as an “additive” model and is the opposite of the subtractive color model.





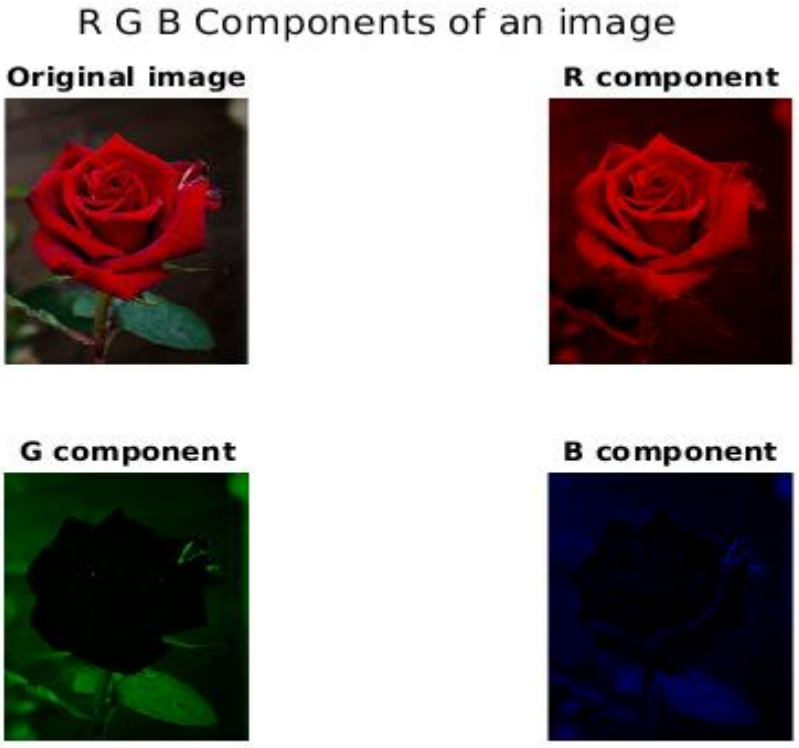
Code:-

```
clc;
clear all;
close all;
a=imread('rose.jpg');
[r g b]=imsplit(a);
sgtitle("R G B Components of an image");
subplot(2,2,1);imshow(a);title("Original image");
r(:,:,1)=r;
r(:,:,2)=0;
r(:,:,3)=0;
subplot(2,2,2);imshow(r);title("R component");
gr(:,:,2)=g;
gr(:,:,1)=0;
gr(:,:,3)=0;
subplot(2,2,3);imshow(gr);title("G component")
bl(:,:,3)=b;
bl(:,:,2)=0;
bl(:,:,1)=0;
subplot(2,2,4);imshow(bl);title("B component");
```

Code Explanation:

clc, clear all, close all are the statements which will clear command window, workspace and figures respectively. A image is read into the variable image and resized. After that, By using some inbuilt functions we wrote the code”.

Output:-



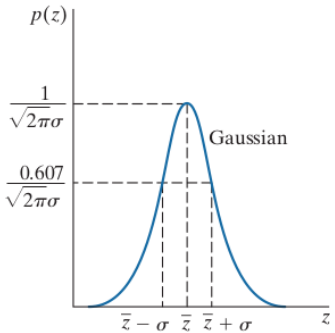
Result: Here,we split an RGB image into three components R G B i.e Red , Blue,Green components of images.Hence we have successfully splitted the RGB image into its individual components.

14.Noise in images and its reduction

Theory:-

The principal sources of noise in digital images arise during image acquisition and/or transmission. The performance of imaging sensors is affected by a variety of environmental factors during image acquisition, and by the quality of the sensing elements themselves. For instance, in acquiring images with a CCD camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are corrupted during transmission principally by interference in the transmission channel. For example, an image transmitted using a wireless network might be corrupted by lightning or other atmospheric disturbance.

Gaussian Noise:-The PDF of a Gaussian random variable, z, is defined by the following familiar expression:

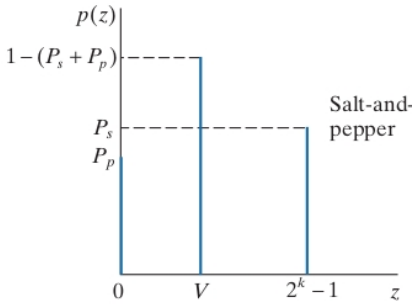


$$p(z)=\frac{1}{\sqrt{2\pi\sigma}}e^{\frac{-(z-\bar{z})}{2\sigma^2}} \quad -\infty<z<\infty$$

where z represents intensity, z is the mean (average) value of z, and s is its standard deviation.

Salt-and-Pepper Noise:-If k represents the number of bits used to represent the intensity values in a digital image, then the range of possible intensity values for that image is [0 , 2 k − 1] (e.g.,[0 , 255] for an 8-bit image). The PDF of salt-and-pepper noise is given by

$$p(z)=\begin{cases}P_s & \text{for } z=2^k-1\\ P_p & \text{for } z=0\\ 1-(P_s+P_p) & \text{for } z=V\end{cases}$$



where V is any integer value in the range 0 < V < 2^k − 1 .
The mean of salt-and-pepper noise is given by
$\bar{z}$ = (0) P_p + K (1 − P_s − P_p) + (2^k − 1) P_s
and the variance by
s^2 = (0 − $\bar{z}$) 2 P_p + (K − $\bar{z}$) 2 (1 − P_s − P_p) + (2^k − 1) 2 P_s

Median Filter:-The best-known order-statistic filter in image processing is the median filter, which,as its name implies, replaces the value of a pixel by the median of the intensity levels in a predefined neighborhood of that pixel:
$\hat{f}$(x , y) = median { g (r , c) } (r , c) ∈ S xy

where, as before, S xy is a subimage (neighborhood) centered on point (x , y). The val-ue of the pixel at (x , y) is included in the computation of the median.
Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise.This filter is used to remove salt and pepper noise

Wiener filtering:-

There is a technique known as Wiener filtering that is used in image restoration. This technique assumes that if noise is present in the system, then it is considered to be **additive white Gaussian noise** (AWGN).

Code:-

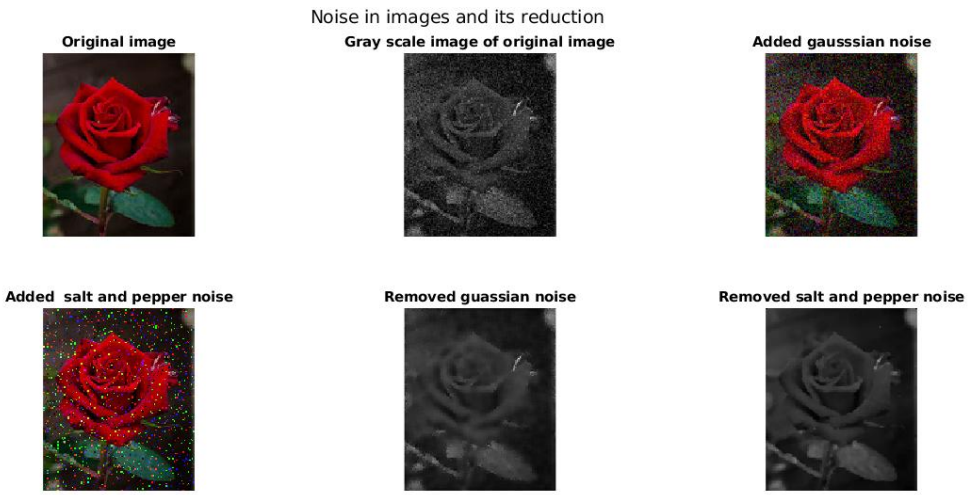
```
clc;close all;
clear all;
im=imread("rose.jpg");
n1=imnoise(im,'gaussian');
n2=imnoise(im,'salt & pepper');
```

```
e=rgb2gray(n1);
f=rgb2gray(n2);
c1=wiener2(e,[5 5]);
c2=medfilt2(f,[3 3]);
subplot(2,3,1)
imshow(im);
title("Original image");
sgtitle("Noise in images and its reduction");
subplot(2,3,2);mshow(e);title("Gray scale image of original image")
subplot(2,3,3);imshow(n1);title("Added gausssian noise");
subplot(2,3,4);imshow(n2);title("Added  salt and pepper noise");
subplot(2,3,5);imshow(c1);title("Removed guassian noise")
subplot(2,3,6);imshow(c2);title("Removed salt and pepper noise");
```

Code explanation:-

clc,clear all , close all are the statements which will clear command window,workspace and figures respectively.A image is read and stored in a varaiable.Next we have applied guassian noise function and the salt and pepper noise function to the input image.Next we have converted the noise images and passed through weiner filter and median filter and we get output images and then we display output images.

Output:-



Result :-Noise reduction in images are used to remove visibility of noise in a noisy image and try to make it as original image by removing noise.Hence we have successfully applied and reduced noise in images.

15.Composite Images

Theory:-

The image fusion process is defined as gathering all the important information from multiple images, and their inclusion into fewer images, usually a single one. This single image is more informative and accurate than any single source image, and it consists of all the necessary information. The purpose of image fusion is not only to reduce the amount of data but also to construct images that are more appropriate and understandable for the human and machine perception.Image fusion refers to the process of combining two or more images into one composite image, which integrates the information contained within the individual images . The result is an image that has a higher information content compared to any of the input images.The purpose of image fusion is to combine information from multiple images of the same scene into a single image that ideally contains all the important features from each of the original images. The resulting fused image will be thus more suitable for human and machine perception or for further image processing tasks.

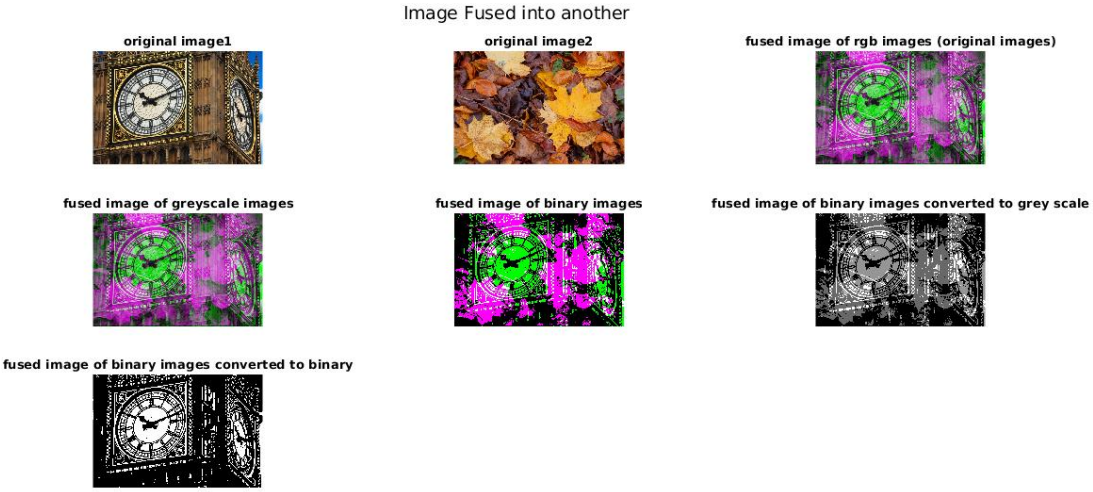
code:-

```
clc;
clear all;
close all;
i1 = imread('clock.jpg');
i2=imread('leaves.jpg');
i3=rgb2gray(i1);
i4=rgb2gray(i2);
i5=im2bw(i1);
i6=im2bw(i2);
c=imfuse(i1,i2);
d=imfuse(i3,i4);
e=imfuse(i5,i6);
f=rgb2gray(e);
g=im2bw(e);
sgtitle("Image Fused into another");
subplot(3,3,1);imshow(i1);title("original image1");
subplot(3,3,2);imshow(i2);title("original image2");
subplot(3,3,3);imshow(c);title("fused image of rgb images (original images)");
subplot(3,3,4);imshow(d);title("fused image of greyscale images");
subplot(3,3,5);imshow(e);title("fused image of binary images");
subplot(3,3,6);imshow(f);title("fused image of binary images converted to grey scale");
subplot(3,3,7);imshow(g);title("fused image of binary images converted to binary");
```

code explanation:-

Here we take two images and we read them.Next we convert rgb images into gray and gray images to binary ones.We fuse both images in three domains rgb,gray and binary.We display the images.

output:-



Result :-Image fusion techniques are widely used in various applications such as remote sensing,medical imagnig,military and astronomy.Hence we have successfully fused two images.

16.Edge Detection

Definition : Edge Detection is an image processing technique for finding boundaries of objects within image. It works by detecting discontinuity in brightness.
Basic Edge Detecting techniques are:
Sobel Edge Detection
Canny Edge Detection
Prewitt Edge Detection

Sobel Edge Detection: Find edges at those points where the gradient of the image is maximum, using Sobel approximation to the derivative.

Canny Edge Detection: Finds edges by looking for local maxima of the gradient of the image.The edge functions calculates the gradient using the derivative of a Gaussian filter. This method uses two threshold to detect strong and weak edges including weak edges in the output if they are connected to strong edges

Prewitt Edge Detection: use to detect edges based applying a horizontal and verticle filter in sequence. Both filters are applied to the image and summed to form the final result. The two filters are basic convolution filters of the form: Horizontal Filter.

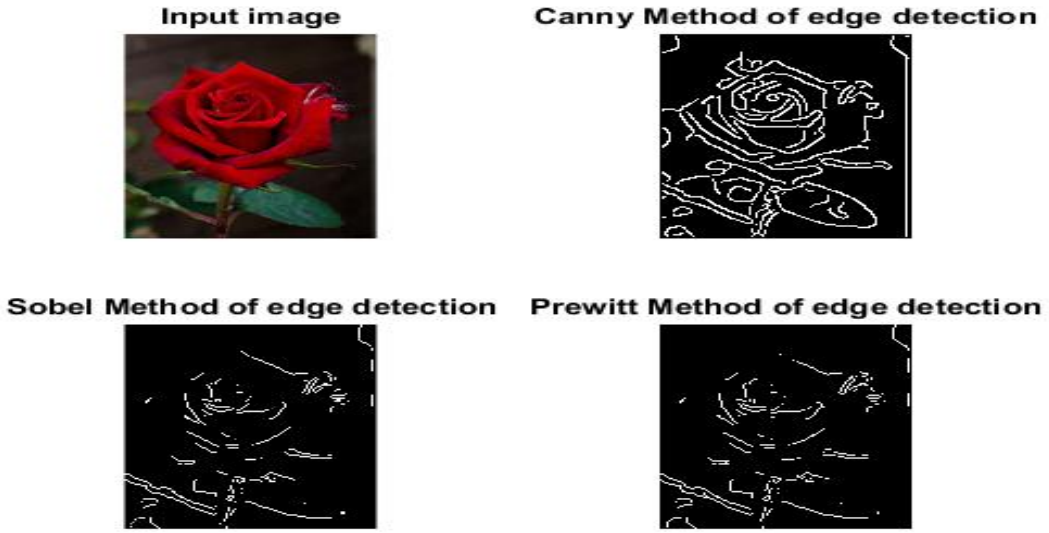
Code:

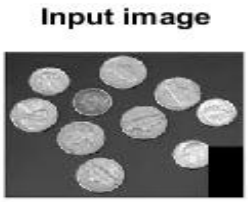
```
clc;
clear all;
close all;
a=imread('rose.jpg');
e=rgb2gray(a);
b=edge(e,'canny');
c=edge(e,'sobel');
d=edge(e,'prewitt');
sgtitle("Edge detection");
subplot(2,2,1);imshow(a);title('Input image');
subplot(2,2,2);imshow(b);title('Canny Method of edge detection');
subplot(2,2,3);imshow(c);title('Sobel Method of edge detection');
subplot(2,2,4);imshow(d);title('Prewitt Method of edge detection');
```

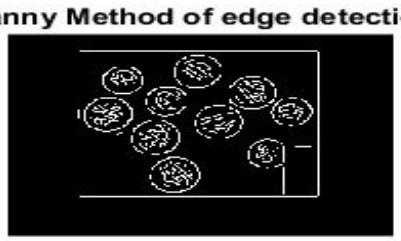
Code Explanation:-

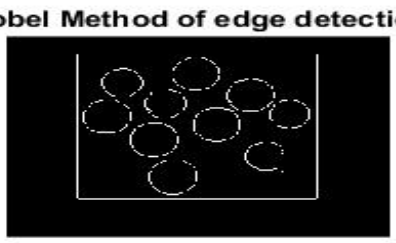
clc, clear all , close all are the statements which will clear command window, workspace and figures respectively. A image is read into the variable ‘a’ then we have applies edge filters – canny filter, sobel filter , prewitt filter for image and then the outputs are displayed using figures.

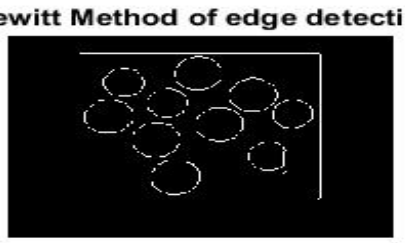
Output:











Result: Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision**.** Hence Edge detection is performed successfully.

17.Morphological Image Processing

Theory:

Morphology is a broad set of image processing operations that process images based on shapes.

Morphological operations apply a structuring elements to an input image , creating an output image of same size

Morphological operations in an image:

The basic morphological operations are Dilation and Erosion

Dilation adds pixels to the boundaries of the object in an image , while Erosion removes the pixels on image boundaries.

The number of pixels added or removed depends on the size and shape of the structure element used to process the image.

Code:

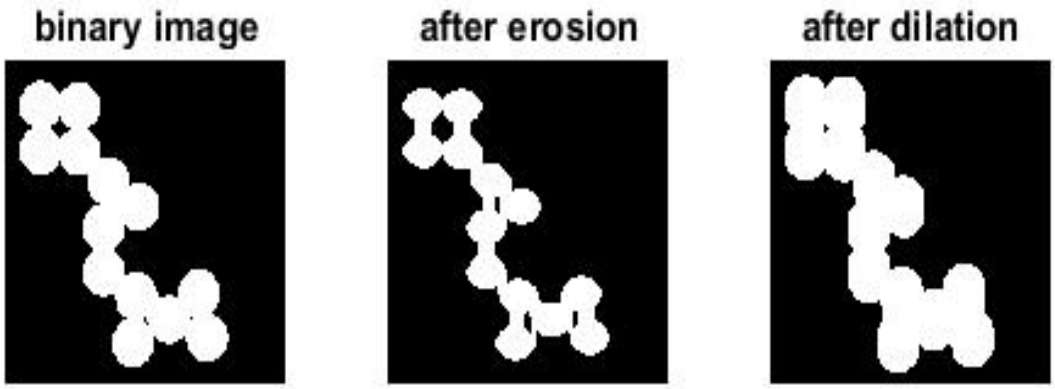
```
clc;
clear all;
close all;
x=imread("dots.png");
y=im2bw(x);
```

```
se=strel('line',11,90);
z=imerode(y,se);
d=imdilate(y,se);
sgtitle(" Morphological Image Processing :-erosion and dilation");
subplot(1,3,1)
imshow(y);
title("binary image");
subplot(1,3,2)
imshow(z);
title("after erosion");
subplot(1,3,3)
imshow(d);
title('after dilation')
```

code explanation:

clc, clear all , close all are the statements which will clear command window, workspace and figures respectively. A image is read into the variable ‘x’and assigned to ‘y’ then we have applies morphological filters of erosion and dilation filter for image and then the outputs are displayed.

Output:-



Result: Some basic aplications of morphology - Dilation adds pixels to the boundaries of the object in an image , while Erosion removes the pixels on image boundaries is executed. Hence Morphological image processing performed successfully

18.Image Compression

Theory:

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It is the process of reducing the amount of data required to represent a given quantity of information.if b and b’ denotesnthe number of bits in two representation of same information. The relative data redundancy,R of the representation with b bits is

R=1-1/C,

where C is compression ratio defined as C=b/b’

Bit-per-pixel ratio, specified as the comma-separated pair consisting of 'bpp' and a scalar. The ratio must be greater than 0 and less than or equal to 8 (grayscale) or 24 compress a truecolor image using the set partitioning in hierarchical trees - 3D ('spiht_3D') compression method.
Load, compress, and store the image in a file. Plot the original and compressed images. Display the compression ratio ('cratio') and the bits-per-pixel ('bpp'), which indicate the quality of the compression.

Code:

```
clc;
clc;
clear all;
close all;
i = imread('clock.jpg');
X=imresize(i,[256,256]);
imwrite(X,"test.jpg")
[cr,bpp] = wcompress('c',X,'wpeppers.wtc','spiht','maxloop',12);
Xc = wcompress('u','wpeppers.wtc');
delete('wpeppers.wtc')
imwrite(Xc,"testc.jpg");
% Display the original and the compressed image
sgtitle("Image Compression");
subplot(1,2,1); imshow(X); title('Original image'); axis square
subplot(1,2,2); imshow(Xc); title('Compressed image'); axis square
```

```
info1=imfinfo('test.jpg');
file_size_of_original_image= info1.FileSize
info2=imfinfo('testc.jpg');
file_size_of_compressed_image=   info2.FileSize
```

Code Explanation:

clc, clear all , close all are the statements which will clear command window, workspace and figures respectively. A image is read into the variable ‘i’ and by the function used above for compression for image makes the input image size reduced according to ‘cr’ and bpp the outputs are displayed.

Output:

Result:

The method of image compression makes to reduce the image pixels qyantity and makes memory free. Hence, the image compression for reducing the size of the file and the image performed successfully.

19.Outcome of project

For many applications of image processing, color information doesn't help us identify important edges or other features. There are exceptions.. Binary images are important when we wish to reduce the amount of information in the image and focus only on regions of the image we need.So we need gray scale images sometimes and sometimes we need rgb image and sometimes binary image.Hence,We have successfully converted RGB images to gray scale images and gray scale images to binary images.

Image subtraction can be used to detect differences between two or more images of the same scene or object.Image arithmetic is used to analyse an image in a convineint way these helps in human interpretation.When we add a costant to an image then image brightens according the value,when we subtract a constant from an image then image brightness,contrast decreases,Image compliment is helpful to analyse the images related to diseases.etc.Hence we have successfully completed some of image arithmetic concepts.

The affine transformation technique is typically used to correct for geometric distortions or deformations that occur with non-ideal camera angles.Geometric transformations in image like rotation ,translation,scalig are useful in setting the images that are defected or deformed at image aquisition state itself.Hence,we have successfully executed some of geometric transformations in an image.

Smoothing is used to reduce noise or to produce a less pixelated image. Most smoothing methods are based on low-pass filters.sharpening filters to accentuate the edges of structures.Hence,we have successfully executed image smoothening and image sharpening.

Applying a low-pass blurring filters mooths edges and removes noise from an image. Blurring is often used as a first step before we perform Thresholding, Edge Detection, or before we find the Contours of an image.Hence,we have successfully executed image blurring.

Histogram is used for graphical representation of a digital image.
It is used to analyze an image. Properties of an image can be predicted by the detailed study of the histogram.Hence,we have successfully executed plotting histograms of different contrast images

The brightness of the image can be adjusted by having the details of its histogram.The contrast of the image can be adjusted according to the need by having details of the x-axis of a histogram.It is used for image equalization. Histograms are used in thresholding as it improves the appearance of the image.Hence,we have successfully executed histogram equalization.

The goal of histogram matching is to take an input image and generate an output image that is based upon the shape of a specific (or reference) histogram.Hence,we have successfully executed histogram matching concept.

Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, implying, it determines the adequacy of numbers of bits used to quantize each pixel , useful for image compression.Hence,we have successfully executed bit-plane slicing.

Here,we split an RGB image into three components R G B i.e Red , Blue,Green components of images.Hence we have successfully splitted the RGB image into its individual components.

Noise reduction in images are used to remove visibility of noise in a noisy image and try to make it as original image by removing noise.Hence we have successfully applied and reduced noise in images

Image fusion techniques are widely used in various applications such as remote sensing,medical imagig,military and astronomy.Hence we have successfully fused two images.

The purpose of edge detection is to discover the information about the shapes and the reflectance or transmittance in an image.It helps in finding the boundaries of edges.Hence we have successfully detected edges of an image.

Morphological operations in an image are useful tools in surface meterology and dimensional metrology.These are used in many applicatios like license plate recognition,character recognition,..etc.Hence we have performed erosion and dilation operarions on images

Image compression reduces the amount of data required to store an image.There are many areas where image compression is used some of them are tv broadcasting,remote sensing via satellite,for military communication systems through radars..etc.We have successfully compressed an image.

20.References

https://in.mathworks.com/help/matlab/workspace.html

https://in.mathworks.com/help/matlab/ref/commandhistory.html
https://in.mathworks.com/help/matlab/matlab_prog/variable-names.html
https://in.mathworks.com/help/matlab/ref/doc.html
https://in.mathworks.com/help/matlab/ref/help.html
https://www.quora.com/In-image-processing-applications-why-do-we-convert-from-RGB-to-Grayscale
https://in.mathworks.com/discovery/affine-transformation.html
www.google.co.in
https://www.javatpoint.com/dip-histograms
https://www.tutorialspoint.com/dip/concept_of_blurring.html

https://www.slideshare.net/hiiampallavi15/smoothing-in-digital-image-processing.html

https://www.quora.com/What-is-image-conversion

https://www.google.com/search?q=image+sharpening+in+image+processing&source=hp&ei=miyKYZ26OYDS1sQP24aNwAE&iflsig=ALs-wAMAAAAAYYo6qs6X-LXA93jGTyd0aE9FCCtCLSMt&oq=image+shar&gs_lcp=Cgdnd3Mtd2l6EAEYAzIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQ6CAgAEIAEELEDOgsILhCABBDHARCvAToLCAAQgAQQsQMQgwFQHVidHGCTMWgAcAB4AIABhwuIAYggkgENMC4zLjQtMS4zLjAuMZgBAKABAQ&sclient=gws-wiz

https://www.google.com/search?q=image+blurring+in+image+processing&bih=576&biw=1366&hl=en&ei=7J2MYb-8HJ6M4-EP6fiVsAk&oq=image+blurring+&gs_lcp=Cgdnd3Mtd2l6EAEYADIFCAAQkQIyBQgAEJECMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQ6BAgAEEM6BwgAELEDEEM6CAgAEIAEELEDOgsILhCABBDHARCvAUoECEEYAFDcAljPHWCfKGgBcAJ4AIAB4AGIAcYTkgEGMC4xMy4ymAEAoAEBsAEAwAEB&sclient=gws-wiz

https://www.mygreatlearning.com/blog/introduction-to-image-processing-what-is-image-processing/#Applications%20in%20Image%20Processing%20%E2%80%93%20A%20Case%20Study

https://www.google.com/search?q=image+conversion+using+matlab&hl=en&ei=eZSMYd79BrrH4-EPtMuVCA&oq=image+conversion+using+matlab&gs_lcp=Cgdnd3Mtd2l6EAMyBggAEBYQHjoECAAQRzoFCAAQkQI6BQguEJECOgsIABCABBCxAxCDAToRCC4QgAQQsQMQgwEQxwEQ0QM6BQguEIAEOgUIABCABDoICC4QsQMQgwE6CAgAELEDEJECOggIABCABBCxAzoOCC4QsQMQgwEQxwEQrwE6CAguEIAEELEDOgQIABBDOgsILhCABBDHARCvAToICAAQFhAKEB46BggAEA0QHjoICAAQDRAFEB46CAgAEAgQDRAeOgUIIRCgAToICCEQFhAdEB5KBAhBGABQtwlY_XdgvXloAnADeACAAdUBiAGsJ5IBBjAuMjUuNZgBAKABAbABAMgBCMABAQ&sclient=gws-wiz&ved=0ahUKEwienpSktY_0AhW64zgGHbRlBQEQ4dUDCA4&uact=5