

**department of Computer Engineering & Applications**

**Institute of Engineering & Technology**

Lab Manual

**Subject Name & Code: Digital Image Processing Lab (CSE481)**

**Course: B. Tech.**

**Year: IV Semester: VII**

|  |  |  |
| --- | --- | --- |
| Ex.No. | Program | Page No. |
| 1 | Create command to familiarize with MATLAB & Create the matrices & perform the various operations. on them | 3 |
| 2 | Understanding Image Basic “image Resize, image type conversion,extraction of color band, creating a synthesic image, psedocolor image” | 7 |
| .3 | Perform Various Arithmetic Operation(Image Addition & complement) & Logical Operation( NOT,OR and XOR) on images | 11 |
| 4 | Perform various Histogram Operations histogram plot, histogram Equalization, Contrast Stretching & gamma correction on images & plot histogram without using imhist function | 15 |
| 5 | Perform smoothing using linear and order statistics filters min, max & med of varying sizes and Sharpen an image using Laplacian filter. | 19 |
| 6 | Perform various Fast Fourier transform (FFT) and frequency domain filtering on images using MATLAB. | 23 |
| 7 | Perform various Morphological operation dilation, erosion, internal & external boundary Extraction, Thinning , Thickening & Skeletonziation of image & Perform Dilation, erosion, boundary Extraction without using direct function | 26 |
| 8 | Perform various thresholding segmentation (Simple,  Multiple, Adaptive & Optimal thresholding) | 30 |
| 9 | Perform the various Edge Detection Operators (Ordinary, Roberts, Prewitts & Sobel Operator) | 33 |
| 10 | Perform the Extraction of image features and Specification in MATLAB  Case Study: Calculate the total No. of Coins & Diameter of Coin in image. | 36 |

**EXPERIMENT NO: 1**

**Environment:** Windows

**Tools/ Language :** MATLAB

**Experiment Objective**The objective of this lab session is to understand and implement the following tasks in MATLAB.

1. Familiarization with MATLAB
2. Making Arrays
3. Making Matrices
4. Matrix Manipulation functions
5. Accessing Elements
6. Extracting a Sub-matrix
7. Matrix Element wise operations
8. Elementary Math functions
9. Perform Matrix all arithmetic, diagonal matrix.
10. Obtain the Plot with different function

**MATLAB Code:**

* 1. **Familiarization with MATLAB**
  2. MATLAB Desktop
  3. Toolboxes
  4. MATLAB Relational operators
  5. MATLAB Logical Operators
  6. MATLAB as calculator
     + % This is a comment
     + >> ((1+2)\*3 - 2^2 - 1)/2 ans: 2
     + % Use ; to suppress output (scripts and functions)
     + >> ((1+2)\*3 - 2^2 - 1)/2;
     + % You need to use the ... operator to wrap lines
     + >> 1 + 2 + 3 + 4 + 5 ...

+ 6 + 7 + 8 + 9 ans: 45

**1.2 Making Arrays**

1. % A simple array
2. >> [1 2 3 4 5] ans: 1 2 3 4 5
3. >> [1,2,3,4,5] ans: 1 2 3 4 5
4. >> 1:5 ans: 1 2 3 4 5
5. >> 1:2:5 ans: 1 3 5
6. >> 5:-2:1 ans: 5 3 1
   1. **Making Matrices**
   2. % All the following are equivalent
   3. >> [1 2 3; 4 5 6; 7 8 9]
   4. >> [1,2,3; 4,5,6; 7,8,9]
   5. >> [[1 2; 4 5; 7 8] [3; 6; 9]]
   6. >> [[1 2 3; 4 5 6]; [7 8 9]]
   7. **Matrix Manipulation functions**
   8. zeros : creates an array of all zeros, Ex: x = zeros(3,2)
   9. ones : creates an array of all ones, Ex: x = ones(2)
   10. eye : creates an identity matrix, Ex: x = eye(3)
   11. rand : generates uniformly distributed random numbers in [0,1]
   12. diag : Diagonal matrices and diagonal of a matrix
   13. size : returns array dimensions
   14. length : returns length of a vector (row or column)
   15. det : Matrix determinant
   16. inv : matrix inverse
   17. eig : evaluates eigenvalues and eigenvectors
   18. rank : rank of a matrix
   19. find : searches for the given values in an array/matrix.
   20. **Accessing Elements**
   21. % Make a matrix
       1. >> A = [1 2 3; 4 5 6; 7 8 9]
       2. ans: 1 2 3
       3. 4 5 6
       4. 7 8 9
   22. % Access Individual Elements
       1. >> A(2,3) ans: 6
   23. % Access 2nd column ( : means all elements)
       1. >> A(:,2) ans: 2 5 8
   24. % Access Individual Elements
       1. >> A([1, 3, 5]) ans: 1 7 5
   25. >> A( [1,3], 2:end ) %1 & 3 rows, column 2 to end
       1. ans: 2 3
       2. 8 9
   26. **Extracting a Sub-matrix** sub\_matrix = matrix ( r1 : r2 , c1 : c2 ) ;

where r1 and r2 specify the beginning and ending rows and c1 and c2 specify the beginning and ending columns to be extracted to make the new matrix.

* 1. >> X = [1,2,3;4,5,6;7,8,9]
     1. X = 1 2 3
     2. 4 5 6
  2. 7 8 9
  3. >> X13 = X(3,1:3) X13 = 7 8 9
  4. >> X22 = X(1:2 , 2:3)
     1. X22 = 2 3
        1. 5 6
  5. >> X21 = X(1:2,1) X21 = 1 4
  6. **Matrix Element wise operations**
  7. >> a = [1,2;1,3];
  8. >> b = [2,2;2,1];
  9. Element wise division
     1. >> c = a./b
     2. c = 0.5 1
     3. 0.5 3
  10. Element wise multiplication
      1. >> c = a.\*b
      2. c = 2 4 3
  11. Element wise power operation
      1. >> c = a.^2
      2. c = 1 4
         1. 9
      3. >> c = a.^b
      4. c =1 4
         1. 3
  12. **Elementary Math functions**
  13. abs - finds absolute value of all elements in the matrix
  14. sign - signum function
  15. sin,cos,… - Trignometric functions
  16. asin,acos… - Inverse trignometric functions
  17. exp - Exponential
  18. log,log10 - natural logarithm, logarithm (base 10)
  19. ceil,floor - round towards +infinity, -infinity respectively
  20. round - round towards nearest integer
  21. real,imag - real and imaginary part of a complex matrix
  22. sort - sort elements in ascending order
  23. sum,prod - summation and product of elements
  24. max,min - maximum and minimum of arrays
  25. mean,median – average and median of arrays
  26. std,var - Standard deviation and variance
  27. **Perform Matrix all arithmetic, diagonal matrix,**

1. Create the following matrices & perform the following operations on them.

A = [1 2 3;3 4 5; 6 7 8] B = [-1 3 10;-9 5 25; 0 14 2]

S = [-1 8 5] T = [7;0;11]

1. A + B
2. S - T %will give error as matrix dimensions do not agree. So if we want to find the difference between the elements of S & T then find S - T'
3. Verify AA-1 = I
4. Multiply elements of S with themselves.
5. 2nd to 3rd Isolate the submatrix that consists of the rows of the matrix A.
6. Find the maximum and minimum elements in the matrix A.
7. Use commands *triu* & *tril* on A
8. Sort the values of the vector S.
   1. **Obtain the Plot with different function**
   2. Plot sin(x) and cos(x) over [0,2pi ], on the same plot with different colours.  
      Use the commands: linspace, hold on, hold off, plot, xlabel, ylabel, title.

clc

clear all

x = linspace(0,2\*pi,50);

s = sin(x);

c = cos(x);

hold on;

plot(x, s,'o--b');

plot(x, c, '^-g');

xlabel Points;

ylabel Sin/Cos;

title SamplePlot;

legend ('sin','cos');

hold off;

* 1. Obtain a plot of the functions f (x) = x2, g (x) = x3 for x = –1, …,1 on the same axis. Label the x and y axes and create a legend indicating which graph is which.

**Post Experiment Question:**

**Ques1:** Define MATLAB?

**Ques2:** Define Matrices?

**Ques3:** What is the use of legend function?

**Ques4:** Explain the syntax of plot

**Ques5:** how could you set xlabel and ylabel?

**EXPERIMENT NO: 2**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understanding image Basic and implements the following tasks in MATLAB.

**2.1** Image resizing

Image type conversion

**2.2** Extraction of colour bands

Creation of a synthetic image

Pixelization and false contouring

Display of a psedocolour image

**MATLAB syntax:**

* The MATLAB command imresize() is used to resize the image. The syntax is

imresize(Image,{Parameters})

for example: command imresize(Imagehandle,[256,256]) resizes the given image to an image of size 256 x 256 and imresize(Imagehandle,k) scales the image by a factor of k.

* The colour image can be converted to a grey scale image using the MATLAB command rgb2gray(Imagehandle) and to binary image using the command im2bw(Imagehandle).
* The MATLAB command subimage(), similar to subplot(), is used to display multiple images. This command is used when the images to be displayed use different colour maps.
* Mycolourimage(:,:,1) extracts the red band. In this line of code, by changing 1 to 2 we can extract the green band and the change of 1 to 3 would extact the green band.
* Imhist() can be used to plot the histogram
* A synthetic image can be generated with the command phantom(). Two parameters are used, one for creating a medical p

**MATLAB Code:**

**2.1 Resizing image & image Type Conversion**

clc

clear all

close all

**%read the colour image**

myimage = imread('pot.jpg');

**%resize the colour image to 256 x 256 image**

mycolorimage=imresize(myimage,[256,256],'nearest');

**%convert the colour image to a greylevel image**

mygrayimage = rgb2gray(mycolorimage);

%convert the colour image to a greylevel image

mybinimage = im2bw(mycolorimage);

**%Display the Original Image in a grid of 2 x 2 images**

subplot(2, 2, 1);

imshow(mycolorimage); title('Original Colour Image');

**%Display the Grey level Image in a grid of 2 x 2 images**

subplot(2, 2, 2);

imshow(mygrayimage); title('Grey Image');

**%Display the Original Image in a grid of 2 x 2 images**

subplot(2, 2, 3);

imshow(mybinimage); title('Binary Image');

**%Display the Line Profile of a line drawn with coordinates**

%(10,45) and (50,100)

**%Original Image in a grid of 2 x 2 images**

subplot(2, 2, 4);

improfile(mygrayimage,[10,50],[45,100]);

**%Display x axis, y-axis and Title of the Line profile graph**

ylabel('Pixel Value');

xlabel('Distance');

title('Intensity profile of the given line');

**2.2 Perform Extraction of colour bands, Creation of a synthetic image, Pixelization and false contouring, Display of a psedocolour image**

clc

close all

**% Read the colour image and resize it to 256 x 256**

myimage = imread('zoo1.jpg');

mycolourimage = imresize(myimage,[256,256],'nearest');

**%Display the Colour Image**

subplot(2, 4, 1);

subimage(mycolourimage), title('Original colour image');

axis off; axis equal;

**% Extract the Red band from the colour image**

myredimage = mycolourimage(:, :, 1);

subplot(2, 4, 2);

subimage(myredimage); title('Red channel');

axis off; axis equal;

**% Display the histogram of the Red channel**

subplot(2, 4, 3);

imhist(myredimage); title('Red channel');

axis off;

**%Convert the colour image to Grey level image**

mygrayimage = rgb2gray(mycolourimage);

subplot(2, 4, 4);

subimage(uint8(mygrayimage)), title('Grey image-8 bits');

axis off; axis equal;

**% Resolution demo**

**% let us convert mygrayimage to 8-bit image for resolution demo**

**% Reduce the resolution to 8 X 8**

myreducedimg = uint8(mygrayimage);

myreducedresolutionimg = imresize(imresize(myreducedimg,1/16),16);

subplot(2, 4,5);

subimage(myreducedresolutionimg), title('resolution now 16 x 16');

axis off; axis equal;

**% Quantization demo**

**% let us convert mygrayimage to 8-bit image for resolution demo**

**% Reduce the resolution to 8 X 8**

subplot(2,4,6);

subimage(grayslice(myreducedimg,2),gray(2));

title('Quantized image with 4 bits');

axis off; axis equal;

**% generate a syntheic Image**

**% Matlab command phantom is used to generate a 64 x 64 image**

subplot(2, 4, 7);

p = phantom('Modified Shepp-Logan',64);

subimage(p), title('Synthetic image');

axis off; axis equal;

**%Display the grey level image using Pesudocolour using** colormap(Jet(256)).

**% This is the default colour map used to display the pesudocolour image**

subplot(2, 4, 8);

subimage(mygrayimage,colormap(jet(256))), title('Pseudocolour image');

axis off; axis equal;

**Post Experiment Question:**

**Ques1:** How to resize the image?

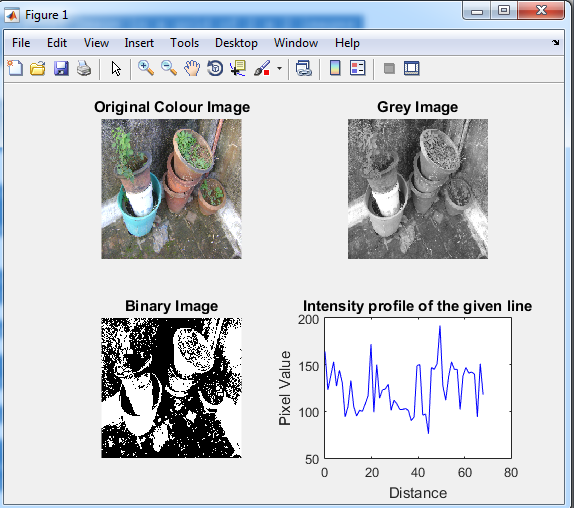
**Ques2:** what is equation of convert RGB image into Grey Image?

**Ques3:** What is the syntax of Phantom function?

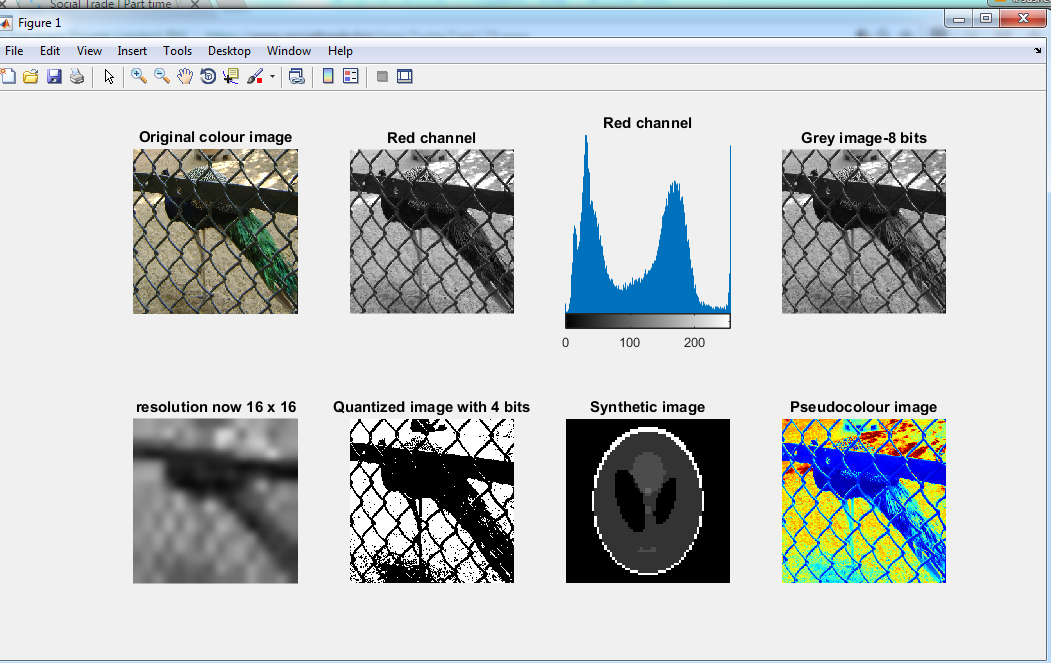
**Ques4:** what is the use of colormap(jet(256))

**Ques5:** how to quantize the image?

2.1 OUTPUT:



2.2 OUTPUT:



**EXPERIMENT NO: 3**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand the image Arithmetic Operations & image Logical Operation and implement the following tasks in MATLAB.

* 1. Image addition

Image complement

* 1. Logical operations such as NOT, OR, AND, and XOR on images

Understand the usefulness of logical operations

**MATLAB Syntax**

* imadd() is used to add a scalar to the image or to add two given images.

imadd(image, scalar)

Similarly for command arthimetic operation imsubtract(), immultiply(), imdivide(), imcomplement(), imabsdiff().

* Logical operation such as XOR(image1, image2) , etc.

**MATLAB Code:**

**3.1. Perform Arithmetic operation (images Additions, Image Complements)**

clc

close all

**% Read the Original Image and resize it to 256 x 256**

**% Let it be Image A**

**% Display the Original Image**

mygrayimg = imread('dog.tif');

scaledimageA = imresize(mygrayimg,[256,256]);

subplot(2, 2, 1);

imshow(scaledimageA,[]); title('Original Image A');

**% Read the Original Image and resize it to 256 x 256**

**% Let it be Image B**

**% Display the Original Image**

subplot(2, 2, 2);

mygrayimg1 = imread('cb.jpg');

scaledimageB = imresize(mygrayimg1,[256,256]);

imshow(scaledimageB,[]); title('Original Image B');

**% Take the complent of Image B and display it**

**% imcomplement() is used to take complement of the Image**

d = imcomplement(scaledimageB);

subplot(2, 2, 3);

imshow(d,[]); title('Image complement');

**% Add the value 50 to all the pixels of the Image A and display it**

**% imadd() is used to add a scalar with the image**

modbrightimg = imadd(scaledimageA,50);

subplot(2, 2, 4);

imshow(modbrightimg); title('Add Brightness of the Image');

**3.2 Perform Logical Operation on Images (NOT,OR,AND and XOR)**

clc

close all

**% Read Image A that is an image of an ellipse**

**% Convert Image A to a binary image**

myimageA=imread('ellipse.jpg');

mybinaryimageA = im2bw(myimageA);

**% Read Test image B**

**% Convert Image B to a binary image**

myimageB=imread('test1.jpg');

myimageadjustB =imresize(myimageB,[256,256]);

mybinaryimageB = im2bw(myimageadjustB);

**% Display the Original Image A**

subplot(4,2,1)

imshow(mybinaryimageA),title('Binary Image - Image A ');

**% Display the Original Image B**

subplot(4,2,2)

imshow(mybinaryimageB),title('Binary Image - Image B');

**% Take a complement of Image A and Display it**

subplot(4,2,3)

resultor= ~mybinaryimageA ;

imshow(mat2gray(resultor)), title('Complement of Image A');

**% Take a Ex-OR of Image A and Image B and Display it**

subplot(4,2,4)

resultxor= xor(mybinaryimageA,mybinaryimageB);

imshow(mat2gray(resultxor)), title('Image A XOR Image B');

**% Take OR of Image A and Image B and Display it**

subplot(4,2,5)

resultor= mybinaryimageA | mybinaryimageB;

imshow(mat2gray(resultor)), title('Image A OR Image B ');

**% Take AND of Image A and Image B and Display it**

subplot(4,2,6)

resultand= mybinaryimageA & mybinaryimageB;

imshow(mat2gray(resultand)), title('Image A AND Image B ');

**% Read a colour image and convert it to 8 bit grey level image**

subplot(4,2,7)

mycolourimg = imread('zoo1.jpg');

mygrayimg = rgb2gray(mycolourimg);

subimage(mygrayimg),title('Original Grey Image');

axis off; axis equal;

**% Extract seventh plane and display it**

subplot(4,2,8)

planeimg = double(mygrayimg);

k = 128;

mysevengreyimg = mod(floor(planeimg/k),2);

imshow(mysevengreyimg),title('Seventh Plane');

**Post Experiment Question:**

**Ques1:** How to add image, what is syntax?

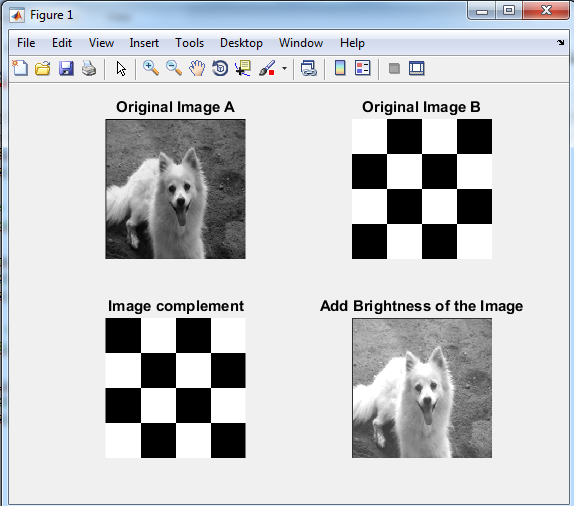
**Ques2:** list the different type of Logical operation on image?

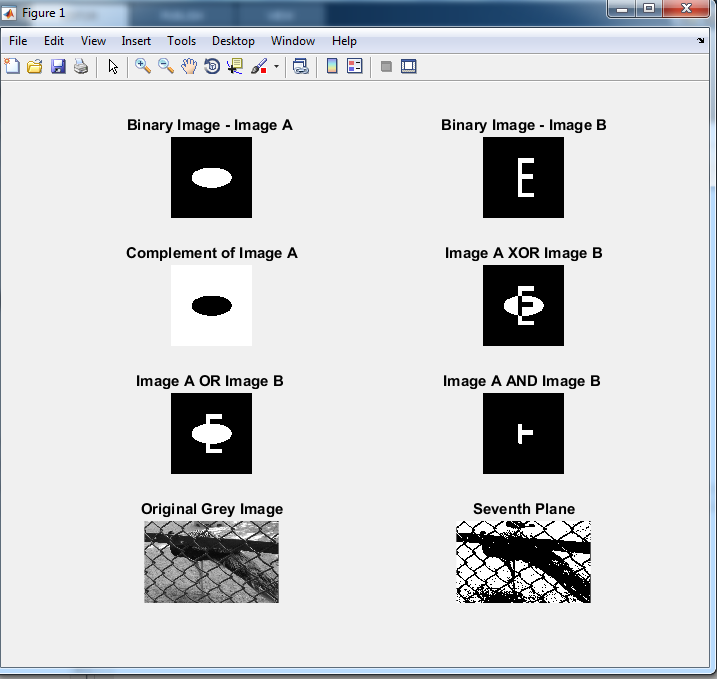
**Ques3:** What is the syntax of subplot?

**Ques4:** How to extract the seventh plane in image?

**Ques5:** what is used of axis off?

**3.1 OUTPUT:**



**3.2. OUTPUT:** 

**EXPERIMENT NO: 4**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand Image Enhancement Operations and implement the following tasks in MATLAB.

4.1 Histogram operations

Contrast stretching and gamma correction on images

4.2 Plot histogram without using imhist function

**MATLAB Requirements:**

* Imhist(image name,[Parameters]) is used to display the histogram of an image. When no parameters are given , MATLAB by default assumes 256 bins of displaying the histogram
* Imadjust(image,(oldmin,oldmax),(newmin,newmax)) maps the old intensity range in the image to a new intensity range to improve the visual quality of the image.
* Imhisteq(image,a) is used to enhance the constrast of the image using histogram equalization
* Normalize the Histogram the maximum value to be plotted will always be 1.
* bar (h) is use to plot the histogram.

**MATLAB Code:**

**4.1 Performs various Histogram Operations (histogram plot, histogram Equalization, Contrast Stretching & gamma correction on images)**

clc

close all

mycolourimage = imread('pot.jpg');

mygrayimage = rgb2gray(mycolourimage);

mygrayimage = imsubtract(mygrayimage,60);

subplot(2,3,1);

imshow(mygrayimage);

title('Original Image');

**% Find histogram of the Image**

**%imhist works with only 8 bit images**

**% Hence convert the image to unsigned 8 bit image and plot the histogram**

myimage = uint8(mygrayimage);

subplot(2,3,2);

imhist(myimage);

axis off, axis tight;

title('Histogram of the Image');

**% Adjust the intensity of the Image**

subplot(2,3,3);

myadjustedimage = imadjust(myimage,stretchlim(myimage),[]);

imshow(myadjustedimage);

title('Contrast adjusted Image');

**% Adjust the Gamma to 0.8**

subplot(2,3,4);

myadjustedgammaimage = imadjust(myimage,[],[],0.8);

imshow(myadjustedgammaimage);

title('Gamma Adjusted Image to 0.8');

**% Histogram equalise**

subplot(2,3,5);

[equalisedimage, T] = histeq(myimage);

imhist(equalisedimage);

axis off, axis tight;

title('Histogram of the Equalized Image');

**% Histogram equalise**

subplot(2,3,6);

imshow(equalisedimage);

title('Histogram-Equalized final Image');

**4.2 plot histogram without using imhist function**

clear all

clc

a=imread('pot.jpg');

a=double(a);

[row col]=size(a);

h=zeros(1,300);

for n=1:1:row

for m=1:1:col

if a(n,m)==0 %%To ensure that the values of a are not zero

a(n,m)=1;

end

end

end

for n=1:1:row

for m=1:1:col

t=a(n,m); %% takes the value of the pixel.ex. value 12

h(t)=h(t)+1; %% incrementing each counter h(12)

end

end

figure(1)

imshow(uint8(a));

figure(2)

bar(h)

**Post Experiment Question:**

**Ques1:** what is Histogram?

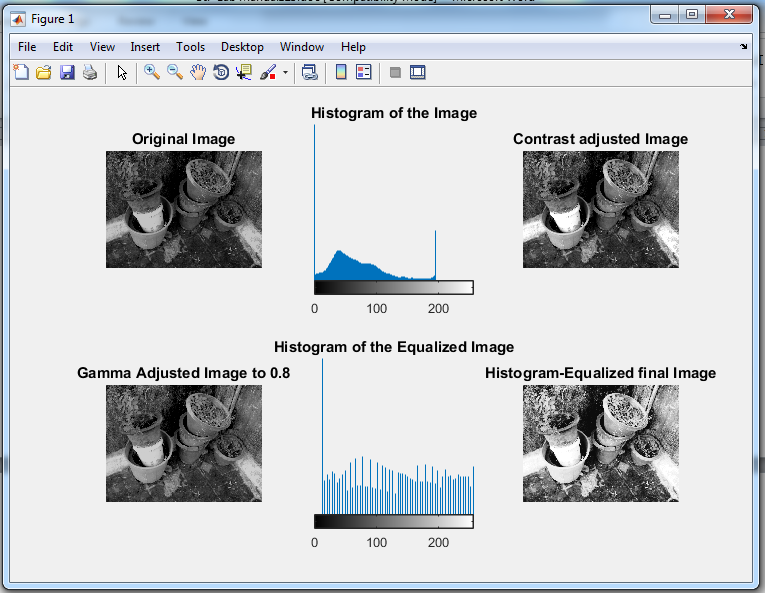
**Ques2:** which command is used to plot the histogram?

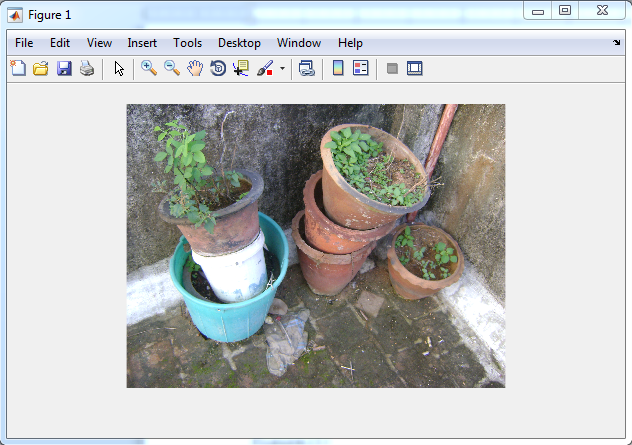
**Ques3:** What is the syntax of subplot?

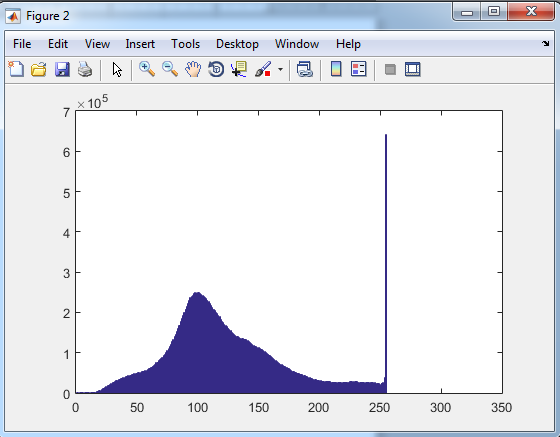
**Ques4:** How to extract the seventh plane in image?

**Ques5:** what is used of axis off?

**4.1. OUTPUT:**



**4.2.OUTPUT:**

****

**EXPERIMENT NO: 5**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand and implement the following tasks in MATLAB

5.1 Perform smoothing using linear and order statistics filters of varying sizes.

5.2 Perform smoothing image using Max Min & Med filtering

5.3 Sharpen an image using Laplacian filter.

**MATLAB Requirements**

* 1. take the odd mask filter size n
* sum of product with image & mask filter
* display the process image
  1. The image (a) applying max (a) find the max, min & med intensity values of image.
  2. Sharpen the image by laplacian filter

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

**MATLAB Code:**

**5.1 Perform smoothing using linear and order statistics filters of varying sizes.**

% Linear smoothing

clear all

clc

i1 = imread('peppers.png');

i1 = rgb2gray(i1);

[m n] = size(i1);

i1 = double(i1);

size = input('size of filter (odd number): ');

f = ones(size);

c = (size + 1)/2;

i2 = i1;

for i = c:m-c+1

for j = c:n-c+1

sum = 0;

for k = 1:size

for l = 1:size

sum = sum + i1(i-c+k, j-c+l)\*f(k,l);

end

end

i2(i,j)= sum / (size ^2);

end

end

figure(1), subplot(1,2,1), imshow(uint8(i1);

figure(1), subplot(1,2,2), imshow(uint8(i2));

**5.2 Perform smoothing image using Max Min & Med filtering**

clear all

clc

i1 = imread('peppers.png');

i1 = rgb2gray(i1);

[m n] = size(i1);

i1 = double(i1);

size = input('size of filter (odd number): ');

f = ones(size);

c = (size + 1)/2;

maxi = i1;

mini = i1;

medi = i1;

for i = c:m-c+1

for j = c:n-c+1

maxi(i,j) = max(max(i1(i-c+1:i-c+size,j-c+1: j-c+size)));

mini(i,j) = min(min(i1(i-c+1:i-c+size,j-c+1: j-c+size)));

medi(i,j) = median(median(i1(i-c+1:i-c+size,j-c+1: j-c+size)));

end

end

figure(1), subplot(2,2,1), imshow(uint8(i1)), title('Original');

figure(1), subplot(2,2,2), imshow(uint8(maxi)), title('Max Filtered');

figure(1), subplot(2,2,3), imshow(uint8(mini)), title('Min Filtered');

figure(1), subplot(2,2,4), imshow(uint8(medi)), title('Median Filtered');

**5.3 Sharpen an image using Laplacian filter.**

%Laplacian

clear all

clc

i1 = imread('peppers.png');

i1 = rgb2gray(i1);

[m n] = size(i1);

i1 = double(i1);

f = [0 1 0;1 -4 1; 0 1 0];

s = i1;

for i = 2:m-1

for j = 2:n-1

sum = 0;

for k = 1:3

for l = 1:3

sum = sum + i1(i-2+k, j-2+l)\*f(k,l);

end

end

s(i,j) = sum;

end

end

sm = i1 - s;

figure(1), subplot(1,3,1), imshow(uint8(i1)), title('Original');

figure(1), subplot(1,3,2), imshow(uint8(s)), title('Sharpened');

figure(1), subplot(1,3,3), imshow(uint8(sm)), title('Subtracted');

**Post Experiment Question:**

**Ques1:** what is Order Statically filter?

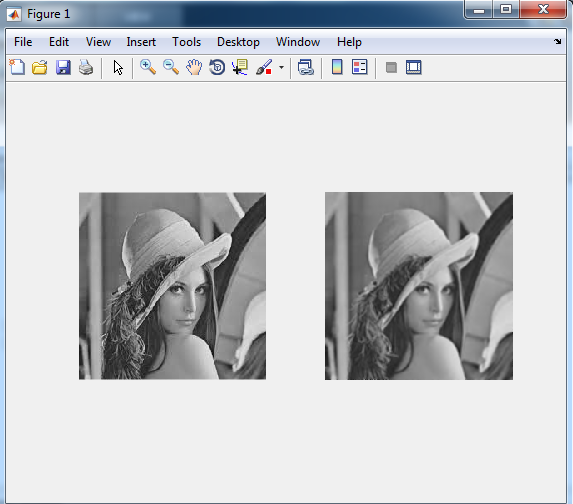
**Ques2:** what is laplacian filter marks?

**Ques3:** what is command of min, max filter?

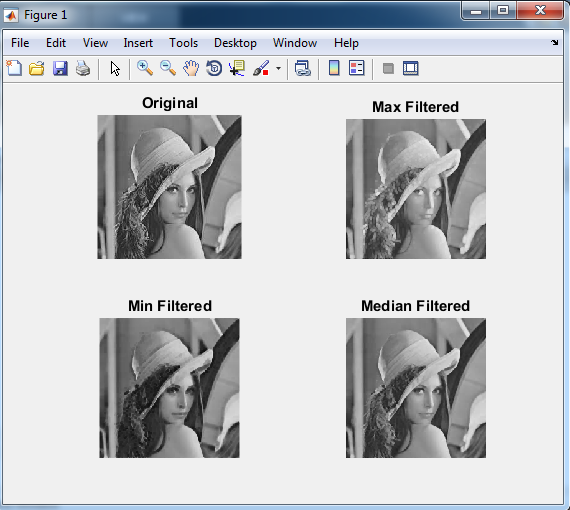
**Ques4:** how to sharpen the image?

**Ques5:** which filter is used to smoothing the image?

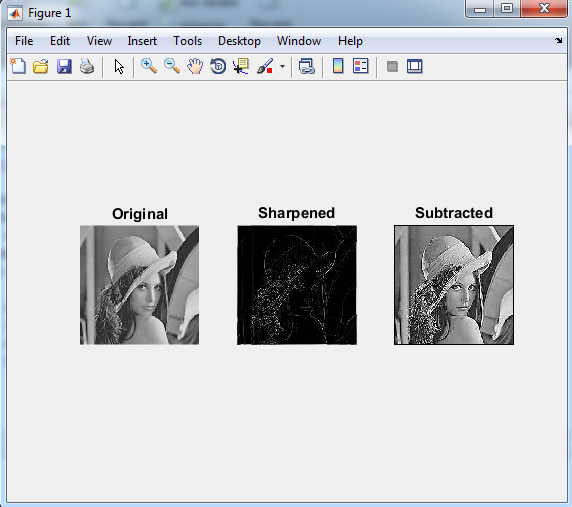
4.1. Output:



4.2. Output:



4.3 Output:



**EXPERIMENT NO: 6**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand and implement the Fast Fourier transform (FFT) and frequency domain filtering on images using MATLAB.

FFT on image

Low- pass filtered Image

Band-pass filtered image

**MATLAB Requirements**

* The command fft2() is used to apply FFT on a two-dimensional image
* The command ffshift() is used to shift the spectrum to the centre.
* The filter mask can be designed by comparing the radial distance with the cufoff frequency
* Ifft2() is used to apply the inverse FFT.

**MATLAB Code:**

clc

close all

clear all

**% Read the image, resize it to 256 x 256**

**% Convert it to grey image and display it**

mygrayimg = imread('dog.tif');

mygrayimg = imresize(rgb2gray(mygrayimg),[256 256]);

subplot(2,2,1);

imshow(mygrayimg),title('Original Image');

**% Find FFT**

**% Use the command fft2() to get FFT of the image**

**% The log scale of FFT image is displayed**

myfftimage = fft2(mygrayimg);

**% Take logarithmic scale for display**

tmp = abs(myfftimage);

mylogimg = log(1+tmp);

subplot(2,2,2);

imshow(mat2gray(mylogimg));

title('FFT Image');

**% Find size**

[M,N] = size(myfftimage);

**% Create Filter array**

**% The cut off frequency 20 is used here**

low = 62;

band1 = 15;

band2 = 60;

**% create ideal high pass filter mask**

**% Create matrix of size equals original matrix**

mylowpassmask = ones(M,N);

mybandpassmask = ones(M,N);

**% Generate values for ideal high pass mask**

for u = 1:M

for v = 1:N

tmp = ((u-(M+1))/2)^2 +(v-(N+1)/2)^2;

raddist = round((sqrt(tmp)));

disp(raddist)

if raddist > low

mylowpassmask(u,v) = 0;

end

if raddist > band2 || raddist < band1;

mybandpassmask(u,v) = 0;

end

end

end

**% Shift the spectrum to the centre**

f1 = fftshift(mylowpassmask);

f3 = fftshift(mybandpassmask);

**% Apply the filter H to the FFT of the Image**

resimage1 = myfftimage.\*f1;

resimage3 = myfftimage.\*f3;

**% Apply the Inverse FFT to the filtered image**

**% Display the low pass filtered image**

r1 = abs(ifft2(resimage1));

subplot(2,2,3);

imshow(r1,[]),title('Low Pass filtered image');

**% Display the band pass filtered image**

r3 = abs(ifft2(resimage3));

subplot(2,2,4);

imshow(r3,[]),title('Band Pass filtered image');

**Post Experiment Question:**

**Ques1:** what is Fourier transform?

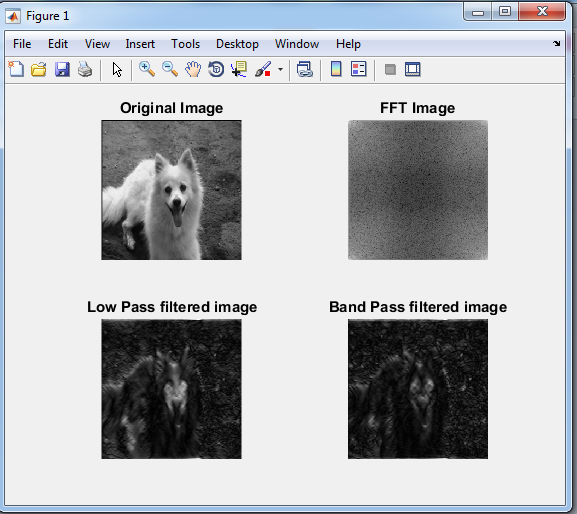
**Ques2:** which command is used to Fourier transform on image?

**Ques3:** what is equation of ideal low pass filter?

**Ques4:** what is equation of band pass filter?

**Ques5:** which command is used to apply inverse Fourier transform?

**Output:**

****

**EXPERIMENT NO: 7**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand and implement the morphological operators and their applications.

7.1 Dilation image

Erosion image

Internal Boundary Extraction

External Boundary Extraction

Morphological gradient

Thinning of the image

Thickening of the image

Skeletonziation of image

7.2 perform Dilation, erosion, boundary Extraction without using direct function

**MATLAB Requirements**

* The command imdilate(image,Structuring element) is used to dilate an image.
* The command strel() is used to create structure element masks
* The command imeorde() and imdilate() is use to perform erosion and dilation respectively on images
* bwmorph() is another useful command for morphological operations such as thinning, thickening and skeletonzation.
* Dilation operation means find the max intensity of an given image
* Erosion operation means find the min intensity of an given iamge
* Boundary Extraction (A)=(A-(A erosion B))

**MATLAB Code:**

**7.1 Perform various Morphological operation on their applications**

clc;

close all;

clear all;

**% Read the test Image**

**% Convert the image to binary image**

myorigimg = imread('test.jpg');

myorigimg = im2bw(rgb2gray(myorigimg));

subplot(3, 3, 1);

imshow(myorigimg);title('Originalimage');

**% Create Structuring Element**

se = strel('disk', 9);

**% Perform dilation operation using imdilate command**

**% Display the dilated image**

mydilatedimg = imdilate(myorigimg, se);

subplot(3, 3, 2);

imshow(mydilatedimg);title('Dilated image');

**% Perform Erosion operation using imerode command**

**% Display the Eroded image**

myerodedimg = imerode(myorigimg, se);

subplot(3, 3, 3);

imshow(myerodedimg);title('Eroded image');

**% Find Internal Boundary**

**% Internal Boundary = Dilated Image AND Not of Eroded Image**

**% Display Internal Boundary**

internalboundimg = mydilatedimg & ~ myerodedimg;

subplot(3, 3, 4);

imshow(internalboundimg,[]);title('Internal Boundary');

**% Find External Boundary**

**% External Boundary = Dilated Image AND Not of Eroded Image**

**% Display External Boundary**

externalboundimg = mydilatedimg & ~myorigimg;

subplot(3, 3, 5);

imshow(externalboundimg,[]);title('External Boundary');

**% Find Morphological Gradient**

**% Morphological Gradient = Dilated Image AND Not of Eroded Image**

**% Display External Boundary**

mymorphgradimg = imsubtract(myorigimg,myerodedimg);

subplot(3, 3, 6);

imshow(mymorphgradimg,[]);title('Morphological Gradient');

**% Perform Thinning operation using bwmorph() command**

**% Display the dilated image**

thinf = bwmorph(myorigimg,'thin');

subplot(3,3,7);

imshow(thinf);title('Thinning of the Image');

**% Perform Thickening operation using bwmorph()command**

**% Display the dilated image**

thickf = bwmorph(myorigimg,'thicken');

subplot(3,3,8);

imshow(thickf);title('Thickening of the Image');

**% Perform Skeletonozation operation using bwmorph()command**

**% with 8 iterations and display the dilated image**

skelf100 = bwmorph(myorigimg,'skel',9);

subplot(3,3,9);

imshow(skelf100);title('Skeletonization - 9 iterations');

**7.2 perform Dilation, erosion, boundary Extraction without using direct function**

clear all

clc

a=imread('test1.jpg');

p=size(a);

w=[1 1 1; 1 1 1; 1 1 1];

for x=2:1:p(1)-1

for y=2:1:p(2)-1

a1=[w(1)\*a(x-1,y-1) w(2)\*a(x-1,y) w(3)\*a(x-1,y+1) w(4)\*a(x,y-1) w(5)\*a(x,y) w(6)\*a(x,y+1) w(7)\*a(x+1,y-1) w(8)\*a(x+1,y) w(9)\*a(x+1,y+1)];

A(x,y)=min(a1);%Erosion

B(x,y)=max(a1);%dilation

Sharp(x,y)=a(x,y)-A(x,y);

end

end

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

subplot(2,2,2),imshow(A),title('erosion');

subplot(2,2,3),imshow(B),title('Dilation');

subplot(2,2,4),imshow(Sharp),title('boundary extracted');

**Post Experiment Question:**

**Ques1:** what is Image Morphology?

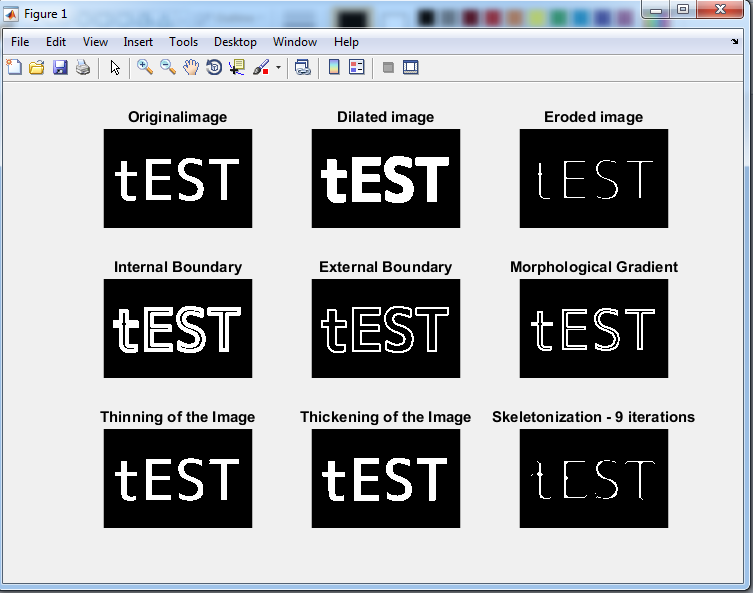
**Ques2:** which command is used to thinning an image?

**Ques3:** what is hit or miss transformation?

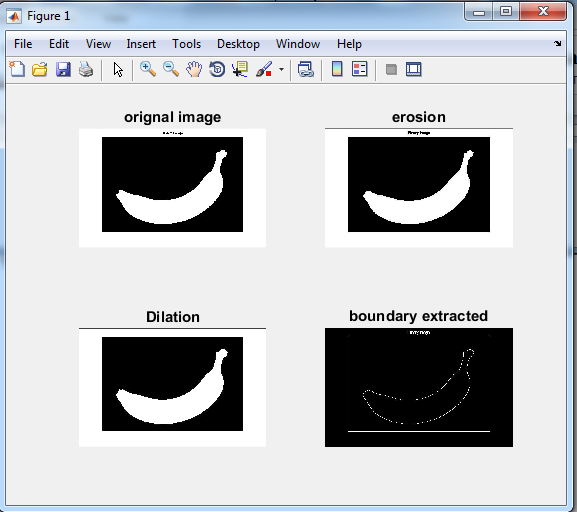
**Ques4:** what is equation of skeletonization?

**Ques5:** how to fill the region in image?

**7.1. OUTPUT:**

****

**7.2. OUTPUT:**



**EXPERIMENT NO: 8**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand Segmentation and implement the following tasks in MATLAB

Simple thresholding

Multiple threshoding

Adaptive thresholding

Optimal thresholding

**MATLAB Requirements**

* The command greythresh() can be used to find optimal threshold value required to segment and image
* Simple thresholding can implemented using the command im2bw(image, label). This command can segment images of any data type with the threshold level specified in the command
* The command find() can be used to identify the coordinates with specific values. This command can be used to implement the multiple thresholding
* The threshold for adaptive algorithms can be either mean+constant or median+constant

**MATLAB Code:**

clc;

close all;

clear all;

a = imread('grayflower256.jpg');

a = rgb2gray(a);

subplot(3,3,1);

imshow(a); title('Original Image');

**% Simple thresholding at 0.3**

**% This is equivalent to saying threshoding at pixel value**

**% 0.3 x 255 = 76.5 , approximatesly 77**

**% imlabel() is a Matlab command that can threshold any matrix**

level = 0.3;

**%% Display the threshold image**

subplot(3,3,2);

segimage1 = im2bw(a,level);

imshow(segimage1); title('Simple Thresholding at 0.3');

**% Simple thresholding at 0.6**

**% So threshold value is 0.6 x 255 = 153**

**% Single Thresholding can be done like this also**

**%% Display the threshold image**

subplot(3,3,3);

imshow(a > 153); title('Simple Thresholding at 0.6');

**% Multiple thresholding Algorithm**

**% Let us assume that the output should be zero if pixel value is**

**% if <= 0.1 x 255 = 25.5 = 26, pixel output 204 if pixel value is <= 0.9 x**

**% 255 = 230 and 0 if pixel value is above 230.**

**%% Create a temporary matrix g**

tmp = a;

[m n]= find(a<26);

for j = 1: length(m)

tmp(m(j),n(j))=0;

end

[m n]= find(a>26 & a <= 230);

for j = 1: length(m)

tmp(m(j),n(j))=0.8;

end

[m n]= find(a>230);

for j = 1: length(m)

tmp(m(j),n(j))=0;

end

subplot(3,3,4);

segimage2 = im2bw(tmp,0);

imshow(segimage2); title('Multiple threshoding(Between 27-230)');

**%% Find the threshold Value using Otsu Algorithm**

level = graythresh(a);

**%% Display the threshold image**

subplot(3,3,5);

segimage = im2bw(a,level);

imshow(segimage); title('Otsu - Optimal Segmented Image');

**%% Display Blured Image**

b = imread('bluredtxt.jpg');

subplot(3,3,6);

imshow(b); title('Badly illuminated Image');

level = graythresh(b);

subplot(3,3,7);

segimage = im2bw(b,level);

imshow(segimage); title('Otsu - Segmentation for bad illuminated Image');

b = imread('bluredtxt.jpg');

b = rgb2gray(b);

**%Create an average Image**

avgfilt = ones(13,13);

adaptfiltmask = avgfilt/sum(avgfilt);

im = imfilter(b,adaptfiltmask,'replicate');

**%Create an median image**

im1 = medfilt2(b,[20 20]);

**%Adaptive threshold algorithm use**

**% threshold = mean + constant (Here 18)**

thresh = im+18;

adaptthreshimg = b - thresh;

subplot(3,3,8);

imshow(adaptthreshimg > 0);

**%Adaptive threshold algorithm used threshold = mean + constant (Here 2)**

thresh1 = im1 + 2;

adaptthreshimg = b - thresh1;

subplot(3,3,9);

imshow(adaptthreshimg > 0);

**Post Experiment Question:**

**Ques1:** what is thresholding?

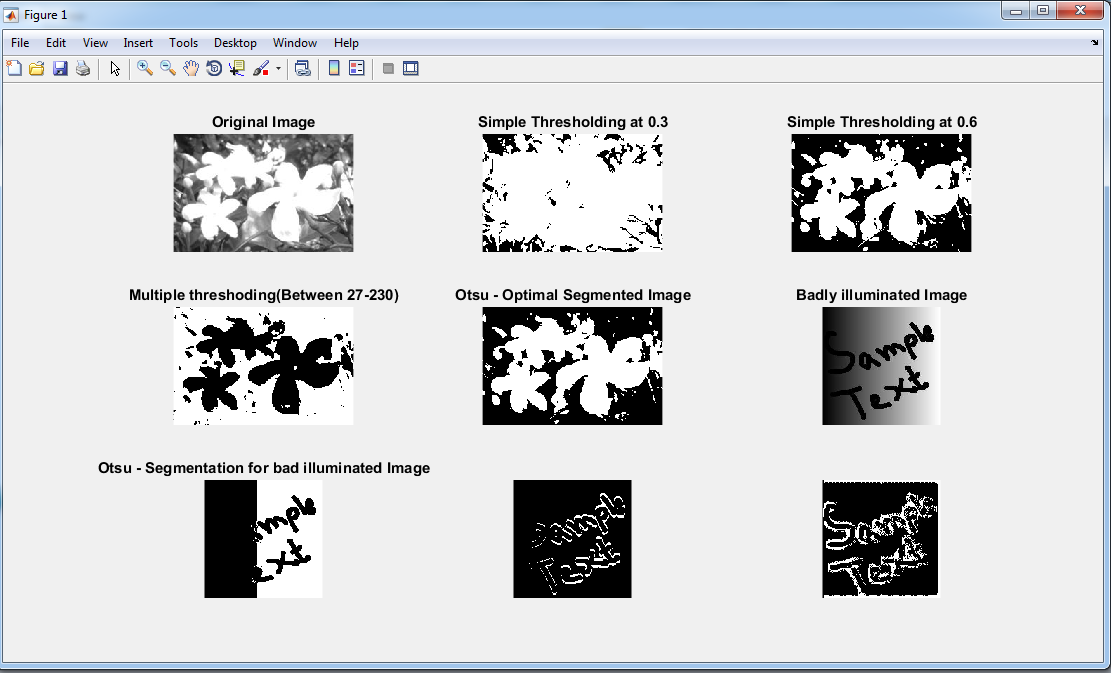
**Ques2:** what is adaptive thresholding?

**Ques3:** what is equation of multiple thresholding?

**Ques4:** what is different between multiple and adaptive thresholding?

**Ques5:** how to splitting & merging the object in an image?

**OUTPUT:**

****

**EXPERIMENT NO: 9**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand edge detection operators in MATLAB.

Ordinary operator

Roberts’s operator

Prewitts operator

Sobel operator

**MATLAB Requirements**

* Convolve the original image with mask1. This gives us the gradient along the x-direction
* Convolve the original image with mask2. This gives us the gradient along the y-direction
* Add the result of 1 and 2
* Convolve the new mask with the original image
* **Ordinary operator mask**

Mask1 [1 0 -1 0] mask2 [1 -1 0 0]

**Matlab Code:**

clear all

clc

a=imread('shape456.png');

%a=double(aa);

[row col]=size(a);

**% ordinary operator**

w1=[1 0; -1 0];

w2=[1 -1; 0 0];

**% Roberts operator**

w3=[1 0; 0 -1];

w4=[0 1; -1 0];

**% prewitts operator**

w5=[-1 0 -1;-1 0 1;-1 0 1];

w6=[-1 -1 -1;0 0 0;1 1 1];

**%sobel operator**

w7=[-1 -2 -1;0 0 0; 1 2 1];

w8=[-1 0 1;-2 0 2; -1 0 1];

for x=2:1:row-1

for y=2:1:col-1

a1(x,y)=w1(1)\*a(x,y)+w1(2)\*a(x,y+1)+w1(3)\*a(x+1,y)+w1(4)\*a(x+1,y+1);

a2(x,y)=w2(1)\*a(x,y)+w2(2)\*a(x,y+1)+w2(3)\*a(x+1,y)+w2(4)\*a(x+1,y+1);

a3(x,y)=w3(1)\*a(x,y)+w3(2)\*a(x,y+1)+w3(3)\*a(x+1,y)+w3(4)\*a(x+1,y+1);

a4(x,y)=w4(1)\*a(x,y)+w4(2)\*a(x,y+1)+w4(3)\*a(x+1,y)+w4(4)\*a(x+1,y+1);

a5(x,y)=w5(1)\*a(x-1,y-1)+w5(2)\*a(x-1,y)+w5(3)\*a(x-1,y+1)+w5(4)\*a(x,y-1)+w5(5)\*a(x,y)+w5(6)\*a(x,y+1)+w5(7)\*a(x+1,y-1)+w5(8)\*a(x+1,y)+w5(9)\*a(x+1,y+1);

a6(x,y)=w6(1)\*a(x-1,y-1)+w6(2)\*a(x-1,y)+w6(3)\*a(x-1,y+1)+w6(4)\*a(x,y-1)+w6(5)\*a(x,y)+w6(6)\*a(x,y+1)+w6(7)\*a(x+1,y-1)+w6(8)\*a(x+1,y)+w6(9)\*a(x+1,y+1);

a7(x,y)=w7(1)\*a(x-1,y-1)+w7(2)\*a(x-1,y)+w7(3)\*a(x-1,y+1)+w7(4)\*a(x,y-1)+w7(5)\*a(x,y)+w7(6)\*a(x,y+1)+w7(7)\*a(x+1,y-1)+w7(8)\*a(x+1,y)+w7(9)\*a(x+1,y+1);

a8(x,y)=w8(1)\*a(x-1,y-1)+w8(2)\*a(x-1,y)+w8(3)\*a(x-1,y+1)+w8(4)\*a(x,y-1)+w8(5)\*a(x,y)+w8(6)\*a(x,y+1)+w8(7)\*a(x+1,y-1)+w8(8)\*a(x+1,y)+w8(9)\*a(x+1,y+1);

end

end

A1=a1+a2;

A2=a3+a4;

A3=a5+a6;

A4=a7+a7;

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

figure(1),subplot(2,3,2),imshow(uint8(A1)),title('ordinary operator');

figure(1),subplot(2,3,3),imshow(uint8(A2)),title('Roberts Operator');

figure(1),subplot(2,3,4),imshow(uint8(A3)),title('Prewitts Operator');

figure(1),subplot(2,3,5),imshow(uint8(A4)),title('Sobel Operatos');

**Post Experiment Question:**

**Ques1:** what is sobel operator & application?

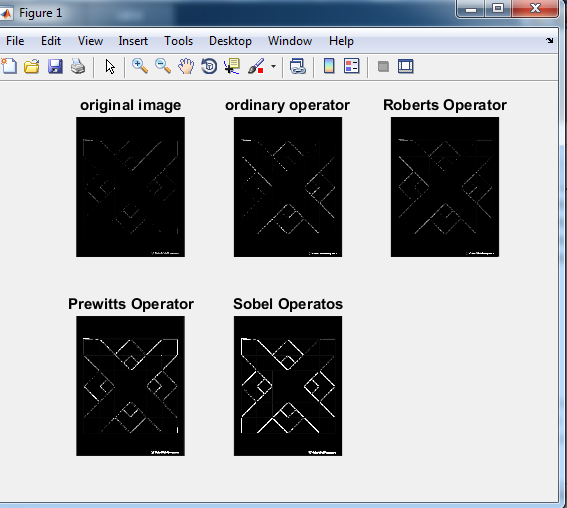
**Ques2:** what is prewitts operator & application?

**Ques3:** what is canny edge detector?

**Ques4:** which operator is best for detecting edge in an image?

**Ques5:** what is command of sobel operator in MATLAB?

**Output:**

****

**EXPERIMENT NO: 10**

**Environment:** Windows

**Tools/ Language:** MATLAB

**Experiment Objective** The objective of this lab session is to understand the Extraction of image features and Specification in MATLAB.

10.2 Counting Number of coins in given image

**MATLAB Requirements**

* The command regionprops() is used to extract all the features
* To reduce the time, the image is cropped to a lesser size
* The command datastats() is used to extract the statistical features.

**MATLAB Code:**

clc;

clear all;

**% Read the Original Image**

**% Convert it to Grey Image, rescale it to 10 x 10 to reduce**

**% running time of this program and display it**

mycolourimg = imread('colorleaf.jpg');

mygrayimg = rgb2gray(mycolourimg);

subplot(1,2,1);

imshow(mygrayimg);

title('My original Image');

croppedimg = imresize(mygrayimg,[10 10]);

subplot(1,2,2);

imshow(croppedimg);

title('Cropped Image of size 10 x 10');

**% Label the image for extraction of features**

labelimage = croppedimg;

labelimage = bwlabel(labelimage);

**% Obtain features of the image using regionprops() command**

**% The parameter all indicates that all the supported features**

**% to be displayed on screen**

featuresimg = regionprops(labelimage,'Area','MajorAxisLength','MinorAxisLength');

**% The individual features can be extracted like this**

fprintf('Thre area is %6.2f \n',featuresimg.Area);

fprintf('The MajorAxisLength is %6.2f \n',featuresimg.MajorAxisLength);

fprintf('The MinorAxisLength is %6.2f \n',featuresimg.MinorAxisLength);

**% Calculate eccenticity**

myecc = featuresimg.MajorAxisLength/featuresimg.MinorAxisLength;

fprintf('The eccentricity is %6.2f \n',myecc);

**% Calculate pixel statistics**

**% Convert the matrix to a single vector**

newcropimg = croppedimg(:);

pixstat=datastats(double(newcropimg));

disp(pixstat)

**10.2 Counting Number of coins in given image**

function ret = CountCoins(img)

img=imread('coins123.png');

subplot(2,2,1);

imshow(img);

subplot(2,2,2);

imgBW = im2bw(img);

imshow(imgBW);

subplot(2,2,3);

imhist(img);

subplot(2,2,4);

imgZ = zeros(size(img));

imgZ(img > 100) = 1;

imshow(imgZ);

ret = round(sum(imgBW(:)) / 2100);

imgConn = bwconncomp(imgZ);

ret = imgConn.NumObjects;

end

**Post Experiment Question:**

**Ques1:** what is Feature extraction?

**Ques2:** what is used of regionprops command?

**Ques3:** how to Calculate eccentricity?

**Ques4:** how to find diameter of coin?

**Ques5:** which command is used to crop the part of image?

**10.1 Output:**

**Thre area is 100.00**

**The MajorAxisLength is 11.55**

**The MinorAxisLength is 11.55**

**The eccentricity is 1.00**

**num: 100**

**max: 255**

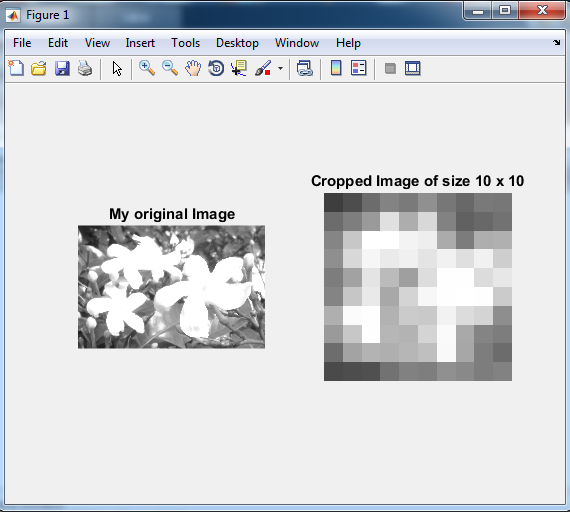
**min: 62**

**mean: 176.5400**

**median: 175.5000**

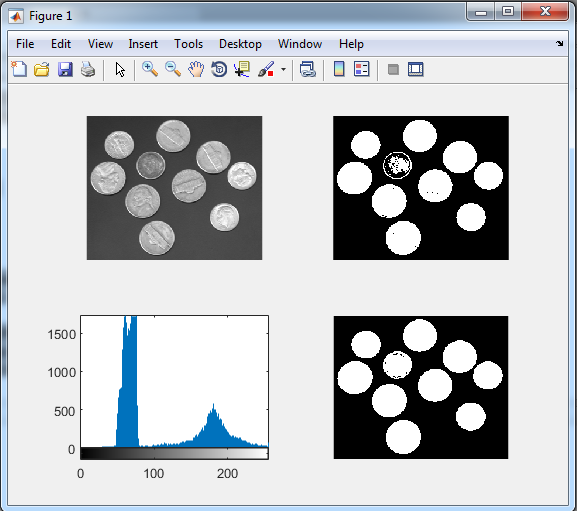
**range: 193**

**std: 54.2714**

****

**10.2 Output**

**ans = 10**

****