

Jawaharlal Nehru Engineering College Aurangabad

Laboratory Manual

DIGITAL IMAGE PROCESSING

For

Third Year IT Students
T.Y(IT)

24, Dec 2009 – Rev 00 –IT– ISO 9000 Tech Document
© Author JNEC, Aurangabad

FORWARD

It is my great pleasure to present this laboratory manual for Third year engineering students for the subject of **Digital Image Processing** keeping in view the vast coverage required for visualization of concepts of Subject with simple language.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

Visual information plays an important role in almost all areas of our life. Today, much of this information is represented and processed digitally. Digital image processing is ubiquitous, with applications ranging from television to tomography, from photography to printing, from robotics to remote sensing. This is a graduate-level introductory course to the fundamentals of digital image processing. It emphasizes general principles of image processing, rather than specific applications.

We expect to cover topics such as ,Basics of digital Image, image Enhancement, segmentation, Compression, color representations

As you may be aware that MGM has already been awarded with ISO 9000 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9000 Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr. H.H Shinde
Principal

LABORATORY MANUAL CONTENTS

This manual is intended for the **Third year** students of **INFORMATION TECHNOLOGY** in the subject of **DIGITAL IMAGE PROCESSING** . This manual typically contains practical/Lab Sessions related DIP covering various aspects related the subject to enhanced understanding.

As per the syllabus, the Practicals are to be conducted in MATLAB. The programs are for image enhancement (Filtering) in special and frequency domain. Segmentation, morphological operations, Color Image processing. Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions.

Ms. Vijaya K. Ahire
Asst. Professor
Information Technology

Prof S. C .Tamane
Head
Information Technology



MGM's
Jawaharlal Nehru Engineering College, Aurangabad

Department of Information Technology

Vision of the department

To develop expertise of budding technocrats by imparting technical knowledge and human value based education.

Mission of the department

- A. Equipping the students with technical skills, soft skills and professional attitude.
- B. Providing the state of art facilities to the students to excel as competent professionals, entrepreneurs and researchers.

Program Specific Outcomes

- PSO1. An ability to design, develop and implement computer programs in the areas related to Algorithms, Multimedia, Website Design, System Software, DBMS and Networking.
- PSO2. Develop software systems that would perform tasks related to Research, Education and Training and/or E governance.
- PSO3. Design, develop, test and maintain application software that would perform tasks related to information management and mobiles by utilizing new technologies to an individual or organizations.

Program Educational Objectives

- PEO1. The graduates will utilize their **expertise** in IT industry and solve industry technological problems.
- PEO2. Graduates should excel in **engineering positions** in industry and other organizations that emphasize design & implementation of IT applications.
- PEO3. Graduates will be **innovators & professionals** in technology development, deployment & system implementation.
- PEO4. Graduates will be pioneers in engineering, engineering management, research and **higher education**.
- PEO5. Graduates will be good citizens & cultured human being with full appreciation of importance of IT **professional ethical & social** responsibilities.

Programme Outcomes (POs):

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change

List of Experiments

1. Introduction to MATLAB.
2. Study of Basic DIP functions in MATLAB
3. Programs for Basic Intensity operations and Histogram .
4. Programs for Image Enhancement using spatial domain filters.
5. Programs for Image Enhancement using frequency domain filters.
6. Programs for Image segmentation.
7. Programs for morphological operations on Image.
8. Programs for color Image processing.

DOs and DONT DOs in Laboratory:

1. Do not handle any equipment before reading the instructions/Instruction manuals
2. Read carefully the power ratings of the equipment before it is switched on whether ratings 230 V/50 Hz or 115V/60 Hz. For Indian equipments, the power ratings are normally 230V/50Hz. If you have equipment with 115/60 Hz ratings, do not insert power plug, as our normal supply is 230V/50 Hz, which will damage the equipment.
3. Observe type of sockets of equipment power to avoid mechanical damage
4. Do not forcefully place connectors to avoid the damage
5. Strictly observe the instructions given by the teacher/Lab Instructor

Instruction for Laboratory Teachers::

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.
2. Students should be taught for taking the printouts under the observation of lab teacher.
3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

1. Introduction to MATLAB

Aim :-Study of working environment of MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment

Typical uses include Math and computation Algorithm development Data acquisition Modeling, simulation, and prototyping Data analysis, exploration, and visualization Scientific and engineering graphics Application development, including graphical user interface building MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The name MATLAB stands for matrix laboratory.

MATLAB was originally written to provide easy access to matrix software in university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology.

Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The entire session will illustrate how to use MATLAB. It will cover following points.

Study of MATLAB Desktop.

- 1) Command Window.
- 2) Current Directory.
- 3) Command History.
- 4) Editor
- 5) Workspace Window.
- 6) Figure window.

Study of general MATLAB commands

Commands	Explanation
who	List current variables
whos	List current variables, long form
clear	Clear variables and functions from memory.

pack	Consolidate workspace memory.
load	Load workspace variables from disk.
save	Save workspace variables to disk.
quit	Quit MATLAB session
exit	Exit from MATLAB.

Study of general MATLAB Functions

functions	Explanation/Example
Sqrt(x)	Sqrt(81)
exp(x)	exp(10)
abs(x)	abs(-67)
log(x)	log(1000) natural logarithm base e
log10(x)	log10(x) natural logarithm base 10
factorial(x)	factorial(5)
sin(x)	sin(5)
round(x)	round(17/5)
fix(x)	fix(13/5)
ceil(x)	ceil(11/5)
floor(x)	floor(-9/4)
rem(x)	rem(13,5)
sign(x)	sign(5)

MATLAB predefined variables

ans	pi	eps	Inf	i,j	NaN
-----	----	-----	-----	-----	-----

Study of Data types, Variables, Arithmetic & Logical Operators.

Data types (classes)

- 1) **double** -- Double precision floating point number array

(this is the traditional MATLAB matrix or array)

- 2) **single** -- Single precision floating point number array
- 3) **logical** -- Logical array
- 4) **char** -- Character array
- 5) **cell** -- Cell array
- 6) **struct** -- Structure array
- 7) **int8** -- 8-bit signed integer array
- 8) **uint8** -- 8-bit unsigned integer array
- 9) **int16** -- 16-bit signed integer array
- 10) **uint16** -- 16-bit unsigned integer array
- 11) **int32** -- 32-bit signed integer array
- 12) **uint32** -- 32-bit unsigned integer array
- 13) **int64** -- 64-bit signed integer array
- 14) **uint64** -- 64-bit unsigned integer array

Matrix Indexing, Array indexing, complex data manipulation, string handling

- Creating one dimensional (Vector): row vector, column vector
- Creating 2D Matrix , Array Addressing
- Build in function for handling arrays

length(A)	size(A)	reshape(A,m,n)	A=diag(a)
------------------	----------------	-----------------------	------------------

Use of MATLAB editor to Create M-Files.

1. Writing Simple MATLAB Program ie **.m file**.
2. Study of syntax of various programming constructs.
e.g if,for,while,break,continue,switch,return
3. Writing **.m** programs using functions

Components of a function M-File

- The Function definition line
- The H1 line
- Help text
- The Function Body
- Comments.

4. Develop some Demo Programs using functions

2. Study of DIP functions in MATLAB.

WARMUP EXERCISES:

- Define pixel ,Digital Image , resolution, size of image.
- How Images are stored?

Aim :- Study of Basic DIP functions in MATLAB

Study and apply following functions on various images .

- Reading Images
- Displaying Images
- Write Image to a Disk
- Image Resizing

imread()	imshow()	imwrite()	size()	pixval()	iminfo	plot()
im2uint8()	im2uint16()	mat2gray	im2double()	Im2bw	subplot()	Pause
imcomplement()	imsubtract()	immultiply()	imdivide()	imabsdiff()	imadd()	imadjust()

- cropping
- Zooming Image
- Brightness Enhancement
- Contrast Manipulations
- Arithmetic Operations :- Adding , Subtracting , Multiplying , Dividing Images

% Addition Of Image:-

```
clear all;
J=imread('rice.png');
[R C]=size(J)
L=imread('cameraman.tif');
for i=1:R
    for j=1:C
        K(i,j)=J(i,j)+ L(i,j);
    end
end
subplot(1,3,1)
imshow(J)
title('image1')
subplot(1,3,2)
imshow(L)
title('image2')
subplot(1,3,3)
imshow(K)
title('added image ')
```

% Subtraction Of Image:-

```
clear all;
J=imread('rice.png');
[R C]=size(J);
L=imread('circles.png');
J1=im2double(J);
L1=im2double(L);
for i=1:R
    for j=1:C
        K(i,j)=J1(i,j)- L1(i,j);
    end
end
subplot(1,3,1)
imshow(J1),title('image1')
subplot(1,3,2)
imshow(L1)
title('image2')
subplot(1,3,3)
imshow(K)
title('subtract image')
```

% Multiplication Of Image:-

% Division Of Image:-

```

clear all;
J=imread('rice.png');
[R C]=size(J);
L=imread('circles.png');
J2=im2double(J);
L2=im2double(L);
for i=1:R
    for j=1:C
        K(i,j)=J2(i,j)* L2(i,j);
    end
end
subplot(1,3,1)
imshow(J), title('image1')
subplot(1,3,2)
imshow(L), title('image2')
subplot(1,3,3)
imshow(K),title('multiplied image')

```

% MATLAB code that performs zooming operation through pixel replication

```

clc;
clear all;
close all;
a=imread('C:\Program Files\MATLAB\R2010a\Images\dove1.jpg');
[m n]=size(a);
p=input('Enter the size you want: ');
for i=1:m % loop to extract every row
    for j=1:n % loop to extract every column
        for k=1:p % loop to control the number of replication
            b(i,(j-1)*p+k)=a(i,j);% replication of pixels in row wise
        end
    end
end
c=b;
[m n]=size(c);
for i=1:n % loop to extract every column
    for j=1:m % loop to extract every row
        for k=1:p % loop to control the number of replication
            b((j-1)*p+k,i)=c(j,i); % replication of pixel in column wise
        end
    end
end
imshow(a),title('original image')
figure,imshow(b),title('Zoomed image')
xlabel(sprintf('Zooming factor is %g',p))

```

3. Programs for Point Processing Operations and Histogram.

WARMUP EXERCISES:

1. What do you mean by Point Processing or intensity operations.
2. Which are the Point Processing operations .
Or
Which are the intensity transformations.

Aim:- Write Program for point processing Operations

Theory:

Image enhancement:

Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image. It is used to enhance medical images , images captured in remote sensing , images from satellite . The transformation function has been given below

$$s = T (r)$$

where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of r to each value of s.

Basic Gray level transformation operations are:

- 1) Image Negatives, Complementing Image
- 2) Log Transformations
- 3) Power Law Transformations
- 4) Piecewise Linear Transformations
 - 4.1) Intensity Level Slicing
 - 4.2) Bit Plane Slicing

% binary Image complement

```
clear all
logo=imread('logo.tif')
figure(1)
imshow(logo)
[R C]=size(logo)
for i=1:R
    for j=1:C
        logo1(i,j)=1-logo(i,j);
    end
end
figure(2)
imshow(logo1)
```

% gray Image complement

```
IM=imread('d:\lenna.jpg');
[R C]=size(IM)
for i=1:R
    for j=1:C
        IM1(i,j)=255-IM(i,j);
    end
end
figure
subplot(1,2,1)
imshow(IM)
subplot(1,2,2)
imshow(IM1)
```

% logarithmic transformation

```
a=imread('c:\crow.jpg');
L=255;
c=L/log10(1+L);
d=c*log10(1+double(a));
subplot(1,2,1);
imshow(a),title('original image')
subplot(1,2,2);
imshow(uint8(d)),title('Log transformation image')
```

%This code performs power law transformation

```
clc
clear all
close all
a=imread('c:\myna.jpg');
a=rgb2gray(a);
gamma=1.1;
d=double(a).^gamma;
subplot(1,2,1);
imshow(a),title('original image')
subplot(1,2,2);
imshow(uint8(d)),title('Powerlaw transformation')
```

Thresholding Of Image:-

```
clear all;
I=imread('rice.png');
[R C]=size(I);
for i=1:R
    for j=1:C
        if (I(i,j)<=128)
            J(i,j)=0;
        else
            J(i,j)=255;
        end
    end
end
subplot(1,2,1)
imshow(I)
title('original image')
subplot(1,2,2)
imshow(J)
title('image after thresholding')
```

Program for displaying Histogram of image and Histogram equalization

Input:- Enter gray image. e.g pout.tif

Definition Histogram of an image with L total possible intensity levels in the range [0,G] is defined as the discrete function

$$h(r_k) = n_k$$

where r_k is the kth intensity level in the interval [0,G] and n_k is the number of pixels in the image whose intensity level is r_k .

Algorithm :-

1. Open any gray image say of resolution MxN e.g 256x256. Total no of pixels will be 65536 Here image will be of class uint8.
2. For an 8bit image the gray levels will be in the range [0,255].
3. Create an array of size 1x65536.
4. Read image intensity values starting from first pixel to last.
5. Use a counter 0 to 255 to keep track how many pixels belongs to that intensity.
6. Use plot function to display intensity levels on X-Axis and Number of Pixels of that intensity on Y Axis give proper title to the corresponding axis.
7. Display Image and corresponding Histogram.
8. Display Equalized histogram and enhanced image.


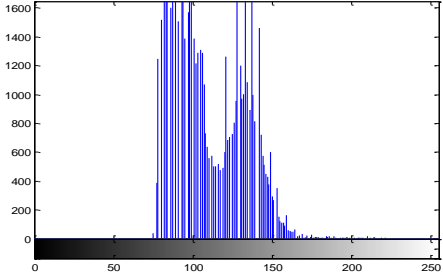

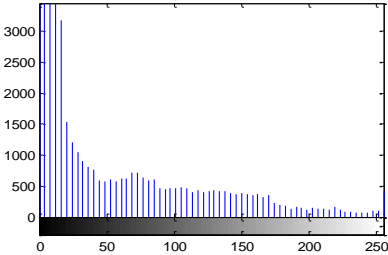

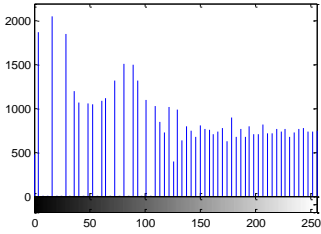
Histogram equalization enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram (uniform distribution by default).

MATLAB Code

```
I = imread('tire.tif');
J = histeq(I);
figure(1); imshow(I),
figure(2); imshow(J)
figure(3); imhist(I,64)
figure(4); imshow(J) imhist(J,64)
```

```
%Program For Histogram Equalization
clc
clear all
close all
a=imread('tire.tif')
% a=imread('C:\deer1.jpg');
%perform histogram equalization
b=histeq(a);
subplot(2,2,1),imshow(a),title('original image'),
subplot(2,2,2),imshow(b),title('After histogram equalization'),
subplot(2,2,3),imhist(a),title('original histogram')
subplot(2,2,4),imhist(b),title('After histogram equalization')
```

Output:

 <p>Input Image pout.tif</p>	 <p>Output Histogram</p>
 <p>Input image tire.tif</p>	 <p>Original Histogram</p>
 <p>Enhanced Image</p>	 <p>Equalized Image</p>

4. Image Enhancement using Spatial Domain filter.

WARMUP EXERCISES:

1. What do you mean by Spatial Domain?
2. Define mask or kernel or Spatial filter.
3. What is neighborhood Processing?
4. What is spatial smoothing? What are uses?
5. Which are the spatial smoothing Filters
6. What is spatial Sharpening? What are uses?
7. Which are the spatial smoothing Filters

Aim :- Program for image enhancement (Smoothing & Sharpening)using spatial domain filters.

Objective :-

The purpose of this assignment is to study image filtering in spatial domain. Spatial filtering is performed by convolving the image with a mask or a kernel. Spatial filters include sharpening, smoothing, edge detection, noise removal, etc.

It consists of four parts the first one discusses the spatial filtering of an image using a spatial mask .3x3, 5x5, and then this mask is used in blurring filter . The second part studies the order statistics filters, specially the median filter.

Part I :- smoothing spatial filter

The output of a smoothing spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.

- These filters are sometimes called *averaging filters* and also *lowpass filters*
- Two types of masks of the spatial filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Steps :-

- 1) Read input Image.
- 2) Add noise using "imnoise()" function.
- 3) Define a (3 x 3) filter.
- 4) Use convolution function conv2() for filtering.

- Order statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in an area covered by the filter
- The best known example in this category is *median filter*
- **median filter**
 - Median filters replace the value of the pixel by the median of the gray levels in the neighborhood of that pixel

Part :-II **sharpening spatial filter**

Spatial domain sharpening filters are also called as High Pass Filters

Laplacian Filters

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Gradient Filter

-1	0	0	-1		
0	1	1	0		
-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Part III :- Study of spatial domain filters .

Study following functions and use on various images with all possible parameters.

<code>fspecial()</code> ,	<code>imfilter()</code>	<code>Ordfilt2()</code>
<code>Medfilt2()</code>	<code>Imnoise()</code>	<code>Median()</code>

fspecial :- Create predefined 2-D filter

Syntax

`h = fspecial(type)`

`h = fspecial(type, parameters)`

Description

h = fspecial(type) creates a two-dimensional filter `h` of the specified type. `fspecial` returns `h` as a correlation kernel,.

Value	Description
'average'	Averaging filter
'gaussian'	Gaussian lowpass filter

'laplacian'	Approximates the two-dimensional Laplacian operator
'prewitt'	Prewitt horizontal edge-emphasizing filter
'sobel'	'Sobel horizontal edge-emphasizing filter
'Unsharp'	"Unsharp contrast enhancement filter

h = fspecial(type, parameters)

accepts the filter specified by type plus additional modifying parameters particular to the type of filter chosen. If you omit these arguments, fspecial uses default values for the parameters. The following list shows the syntax for each filter type. Where applicable,

h = fspecial('average', hsize)

returns an averaging filter h of size hsize. The argument hsize can be a vector specifying the number of rows and columns in h, or it can be a scalar, in which case h is a square matrix. The default value for hsize is [3 3].

h = fspecial('gaussian', hsize, sigma) :-

returns a rotationally symmetric Gaussian lowpass filter of size hsize with standard deviation sigma (positive). hsize can be a vector specifying the number of rows and columns in h, or it can be a scalar, in which case h is a square matrix. The default value for hsize is [3 3]; the default value for sigma is 0.5.

h = fspecial('laplacian', alpha) returns a 3-by-3 filter approximating the shape of the two-dimensional Laplacian operator. The parameter alpha controls the shape of the Laplacian and must be in the range 0.0 to 1.0. The default value for alpha is 0.2.

h = fspecial('log', hsize, sigma) returns a rotationally symmetric Laplacian of Gaussian filter of size hsize with standard deviation sigma (positive). hsize can be a vector specifying the number of rows and columns in h, or it can be a scalar, in which case h is a square matrix. The default value for hsize is [5 5] and 0.5 for sigma.

h = fspecial('prewitt')

h = fspecial('sobel')

Median Filter :- Median filters replace the value of the pixel by the median of the gray levels in the neighborhood of that pixel

- 1) Open /Read an image in a matrix .
- 2) Create a 3x3 matrix B called as mask.
- 3) Read the first 3x3 pixel grid of input image into B.

- 4) Sort the matrix B in ascending order.
- 5) Select the middle value and put that as first pixel value in the output image matrix.
- 6) Repeat the procedure for entire input image by reading the next 3x3 values from the input image and sort using mask B. This way Output image values are calculated.
- 7) Display the input image and Output Image.

Programs for image enhancement using spatial domain filters.

% This program is for Averaging spatial Filter

```

a=imread('D:\DIP Course Material\DIP pract\Images\rose.jpg');
% Addition of noise to the input image
b=imnoise(a,'salt & pepper');
c=imnoise(a,'gaussian');
d=imnoise(a,'speckle');
% Defining 3x3 and 5x5 kernel
h1=1/9*ones(3,3);
h2=1/25*ones(5,5);
% Attempt to recover the image
b1=conv2(b,h1,'same');
b2=conv2(b,h2,'same');
c1=conv2(c,h1,'same');
c2=conv2(c,h2,'same');
d1=conv2(d,h1,'same');
d2=conv2(d,h2,'same');

% displaying the result
figure,subplot(2,2,1),imshow(a),title('Original Image'),
subplot(2,2,2),imshow(b),title('Salt & Pepper noise'),
subplot(2,2,3),imshow(uint8(b1)),title('3 x 3 Averaging filter'),
subplot(2,2,4),imshow(uint8(b2)),title('5 x 5 Averaging filter')
% .....
figure,subplot(2,2,1),imshow(a),title('Original Image'),
subplot(2,2,2),imshow(c),title('Gaussian noise'),
subplot(2,2,3),imshow(uint8(c1)),title('3 x 3 Averaging filter'),
subplot(2,2,4),imshow(uint8(c2)),title('5 x 5 Averaging filter'),
% .....
figure,subplot(2,2,1),imshow(a),title('Original Image'),
subplot(2,2,2),imshow(d),title('Speckle noise'),
subplot(2,2,3),imshow(uint8(d1)),title('3 x 3 Averaging filter'),
subplot(2,2,4),imshow(uint8(d2)),title('5 x 5 Averaging filter'),

```

```
%this program is for comparing averaging & median filter
```

```
clc
```

```
clear all
```

```
close all
```

```
a=imread('D:\DIP Course Material\DIP pract\Images\horse.jpg');
```

```
%Addition of salt and pepper noise
```

```
b=imnoise(a,'salt & pepper',0.1);
```

```
%Defining the box and median filters
```

```
h1=1/9*ones(3,3);
```

```
h2=1/25*ones(5,5);
```

```
c1=conv2(b,h1,'same');
```

```
c2=conv2(b,h2,'same');
```

```
c3=medfilt2(b,[3 3]);
```

```
c4=medfilt2(b,[5 5]);
```

```
subplot(3,2,1),imshow(a),title('Original image')
```

```
subplot(3,2,2),imshow(b),title('Salt & pepper noise')
```

```
subplot(3,2,3),imshow(uint8(c1)),title('3 x 3 smoothing')
```

```
subplot(3,2,4),imshow(uint8(c2)),title('5 x 5 smoothing')
```

```
subplot(3,2,5),imshow(uint8(c3)),title('3x 3 Median filter')
```

```
subplot(3,2,6),imshow(uint8(c4)),title('5 x 5 Median filter')
```

```
%Averaging Filter
```

```
A=ones(200,200);
```

```
A(30:60,30:60)=0;
```

```
A(70:150,50:170)=0
```

```
figure(1)
```

```
subplot(1,2,1)
```

```
imshow(A)
```

```
AM=1/9.*[1 1 1;1 1 1;1 1 1];
```

```
B=conv2(A,AM);
```

```
subplot(1,2,2)
```

```
imshow(B)
```

```
% this program is for sharpening spatial domain filter
```

```
%Sharpening Filters
```

```
A=ones(200,200);
```

```
A(30:60,30:60)=0;
```

```
A(70:150,50:170)=0
```

```
figure(1),subplot(1,2,1)
```

```
imshow(A)
```

```
AM=[1 1 1;1 -8 1;1 1 1];
```

```
B=conv2(A,AM);
```

```
subplot(1,2,2), imshow(B)
```

```
% this program is for sharpening spatial domain filter
```

```
a=imread('D:\horse.jpg');
```

```
%Defining the laplacian filters
```

```
h1=[0 -1 0;-1 4 -1;0 -1 0]
```

```
h2=[-1 -1 -1;-1 8 -1; -1 -1 -1];
```

```
h3=[-1 -1 -1;-1 9 -1; -1 -1 -1];
```

```
c1=conv2(a,h1,'same');
```

```
c2=conv2(a,h2,'same');
```

```
c3=conv2(a,h3,'same');
```

```
subplot(2,2,1),imshow(a),title('Original image')
```

```
subplot(2,2,2),imshow(uint8(c1)),title('Laplacian sharpening 4 at center')
```

```
subplot(2,2,3),imshow(uint8(c2)),title('Laplacian sharpening 8 at center')
```

```
subplot(2,2,4),imshow(uint8(c3)),title('Laplacian sharpening 9 at center')
```

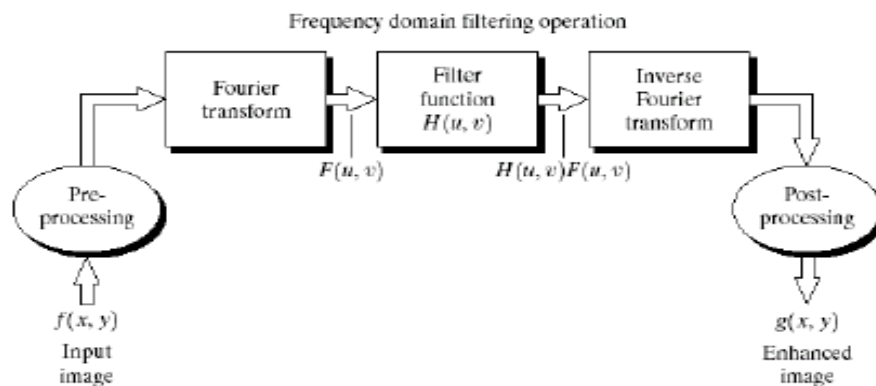
5. Image Enhancement using Frequency Domain Filter.

WARMUP EXERCISES:

1. What do you mean by Frequency Domain?
2. Define Low pass and High pass Frequency Domain filter.
3. What is Frequency Domain smoothing? What are uses?
4. What is Frequency Domain Sharpening? What are uses?
5. Which are the Frequency Domain Filters?

Aim :- Program for image enhancement using frequency domain filters.

Part I :- Study of frequency domain filters .

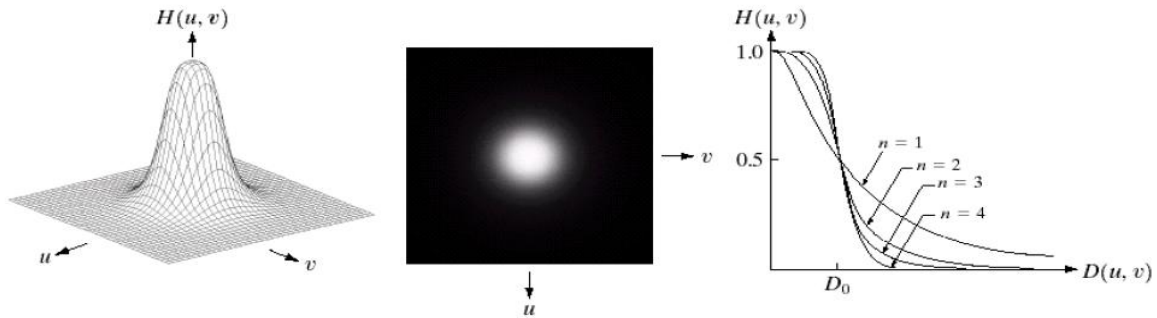


Basic Steps for Filtering in the Frequency Domain:

1. Multiply the input image by $(-1)^{x+y}$ to center the transform.
2. Compute $F(u, v)$, the DFT of the image from (1).
3. Multiply $F(u, v)$ by a filter function $H(u, v)$.
4. Compute the inverse DFT of the result in (3).
5. Obtain the real part of the result in (4).
6. Multiply the result in (5) by $(-1)^{x+y}$.

Butterworth	Gaussian
$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$	$H(u, v) = e^{-D^2(u, v)/2D_0^2}$

Frequency Domain Filtering



a b c

FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

- Low-pass filter: A filter that attenuates high frequencies while passing low frequencies.- used for blurring (smoothing).
- Lowpass filter: A filter that attenuates high frequencies while passing the low frequencies.
- Low frequencies represent the gray-level appearance of an image over smooth areas.

Algorithm:

Butterworth low pass filter:

1. Read the i/p image & its size.
2. Read the cutoff frequency D_0
3. Find center of image
4. Implement the function d
5. Find FFT 2- DFT of i/p image.
6. Shift 2D FFT image
7. Multiply with H
8. Take absolute multiple value of image
9. Display Butterworth low pass image.

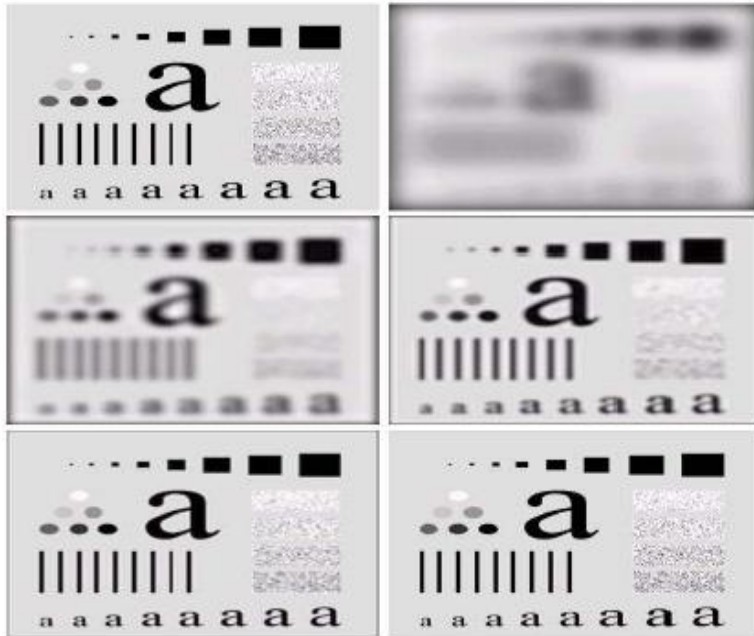


Fig. a) Original Image fig. fig. b) BLPF with cutoff=5,
 Fig. c) cutoff=15, fig. fig. d) LPF with cutoff=30,fig.
 Fig. e) cutoff=80, fig. F fig. f) B LPF with cutoff=230,

Study following functions

fft2()	Abs()	fftshift()	Ifftshift()
--------	-------	------------	-------------

% This code is used to Butterworth lowpass filter

```
close all;
clear all;
clc;
im=imread('pout.tif');
im1=imnoise(im,'salt & pepper',0.05);
figure(1)
imshow(im1);

fc=20;%Cutoff frequency
n=1;
[co,ro] = size(im);
cx = round(co/2); % find the center of the image
cy = round(ro/2);
imf=fftshift(fft2(im1));
```

```

figure(2)
imshow(imf);
H=zeros(co,ro);
for i = 1 : co
    for j =1 : ro
        d = (i-cx).^2 + (j-cy).^ 2;
        H(i,j) = 1/(1+((d/fc/fc).^(2*n)));
    end;
end;
figure(3)
imshow(H)
outf = imf .* H;
out = abs(iff2(outf));
figure(4)
imshow(outf),title('F(u,v)*H(u,v)'),
figure(5)
imshow(im1),title('Original Image'),figure (6),imshow(uint8(out)),title('Lowpass Filterd Image')
%This code is used to gaussian lowpass filter
close all;
clear all;
clc;
im=imread('pout.tif');
fc=10;
imf = fftshift(fft2(im));
[co,ro]=size(im);
out = zeros(co,ro);
cx = round(co/2); % find the center of the image
cy = round (ro/2);
H = zeros(co,ro);
for i = 1 : co
    for j = 1 : ro
        d = (i-cx).^2 + (j-cy).^2;
        H(i,j) = exp(-d/2/fc/fc);
    end;
end;
outf= imf.*H;
out=abs(iff2(outf));
imshow(im),title('Original Image'),figure,imshow(uint8(out)),
title('GaussianLowpass Filterd Image')
figure,imshow(H),title('2D View of H'),figure,surf(H),
title('3D View of H')

```

AIM: Program for sharpen an image using high pass filter in frequency domain

Objective: Student should be able to develop a program by using high pass filter in frequency domain.

Theory:

Sharpening Frequency-Domain Filters

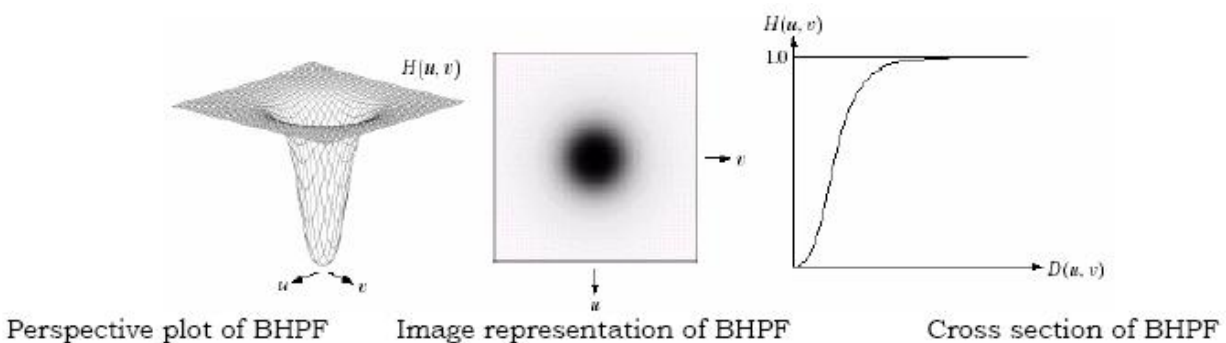
- Butterworth High pass Filter (BHPF): The transfer function of BHPF of order n and with specified cutoff frequency is given by:

Perspective plot of BHPF Image representation of BHPF Cross section of BHPF

$D(u,v)$ is the distance from the origin

D_0 is the cutoff frequency.

n is the order of the filter



$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

High-pass filter: A filter that attenuates low frequencies while passing high frequencies is used for sharpening

High frequencies represent the details such as edges and noise.

Sharpening is achieved by attenuating the LF (low frequency) components of DFT of an image. Where Edges and fine detail in images are associated with high frequency components (high pass filters).

Algorithm:

Butterworth high pass filter --

1. Read the i/p image and size
2. Enter the cutoff frequency d_0
3. Find center of image
4. Implement the function d
5. Find FFT 2- DFT of i/p image .
6. Shift 2D FFT image
7. Multiply with H
8. Take absolute multiple value of image
9. Display Butterworth high pass image

Output:

Butterworth Highpass Filter (BHPF): Smoother results are obtained in BHPF when compared IHPF. There is almost no ringing artifacts.

6. Programs for Image Segmentation.

WARMUP EXERCISES:

1. What do you mean by image segmentation ?
2. What are the applications.
3. What is meant by discontinuity Based segmentation?
4. What is meant by Region Based segmentation?
5. What is meant by Threshold Based segmentation?

Aim :- Write Programs for image segmentation. Program for detecting edges in an image using Roberts cross gradient operator and sobel operator.

Algorithm:

1. Read i/p image & its size
2. Apply Roberts, Prewitt, Sobel & canny edge masks on i/p image
3. Display i/p image & edge detected image.

Edges are those places in an image that correspond to object boundaries. Edges are pixels where image brightness changes abruptly
Prewitt is simpler to implement but canny gives the better result.

```
% This program computes the edges in the image
clear all;
close all;
clc;
a=imread('moon.tif');
b=edge(a,'roberts');
c=edge(a,'sobel');
d=edge(a,'prewitt');
e=edge(a,'log');
f=edge(a,'canny');
imshow(a),title('Original Image')
figure,imshow(b),title('Roberts')
figure,imshow(c),title('Sobel')
figure,imshow(d),title('Prewitt')
figure,imshow(e),title('Log')
figure,imshow(f),title('Canny')
```

7. Programs for Morphological Operations on Image.

WARMUP EXERCISES:

1. What do you mean by morphology ?
2. What are the applications?
3. Define Dilation and erosion.
4. Define Opening and closing.

Aim :- Program for morphological operations on image

Part I :- Study of Logical operations on image.

Objective: Student should be able to develop a program on morphological operation.

Theory :

Morphological operations: including erosion, dilation, opening, closing and Users can arbitrarily choose one of the morphological operations

Structuring Element (SE):

SE size: users can dynamically change the SE size through moving the slider.

SE type: there are 3 types of SE which are implemented in the program, including diamond, square and octagon. Users can dynamically change SE type through the pop-up menu.

User-defined SE: users can arbitrarily change the SE through pressing the mouse button. When users press the specific radio button, MorphPic will show the corresponding resultant morphological operations on the original picture.

Erosion:

The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object (Always try to keep foreground in white). The kernel slides through the image (as in 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero). All the pixels near boundary will be discarded depending upon the size of kernel. So the thickness or size of the foreground object decreases or simply white region decreases in the image. It is useful for removing small white noises detach two connected objects.

Dilation:

It is just opposite of erosion. Here, a pixel element is '1' if atleast one pixel under the kernel is '1'. So it increases the white region in the image or size of foreground object increases. Normally, in cases like noise removal, erosion is followed by dilation. Because, erosion removes white noises, but it also shrinks our object. So we dilate it. Since noise is gone, they won't come back, but our object area increases. It is also useful in joining broken parts of an object.

Set Operation	Expressions For binary Image	Name
$A \cap B$	$A \& B$	AND
$A \cup B$	$A B$	OR
A^c	$\sim A$	NOT
$A - B$	$A \& \sim B$	DIFFERENCE

Part II :- Study following functions parameters.

imdilate()	strel()	imerode()	imopen()	bwmorph()
------------	---------	-----------	----------	-----------

```
% program for Dilation and Erosion
close all;
clear all;
clc;
a=imread('C:\morph1.bmp');
b=[1 1 1;1 1 1;1 1 1];
a1=imdilate(a,b);
a2=imerode(a,b);
imshow(a),title('Original image')
figure,imshow(a1),title('Dilated image')
figure,imshow(a2),title('Eroded image')
```

```
% Program for opening and closing
close all;
clear all;
clc;
a=imread('C:\morph2.bmp');
b=[1 1 1;1 1 1;1 1 1];
a1=imopen(a,b);
a2=imclose(a,b);
imshow(a),title('Original image')
figure,imshow(a1),title('Opened image')
figure,imshow(a2),title('Closed image')
```

Conclusion:

It is a tool for extracting image components that are useful in representation & description region shape. **Dilation** :It is also useful in joining broken parts of an object.**Erosion** : It is useful for removing small white noises detach two connected objects. **Opening** is useful in removing noise. **Closing** is useful in closing small holes inside the foreground objects,

8. Color Image Processing

Aim :- Program for illustrating color image processing.

Study following functions and use on various images with all possible parameters.

rgb2gray()	ind2gray()	ntsc2rgb()	rgb2ind()
rgb2ntsc()	Dither()	Grayslice()	Gray2ind()

Part I:- Displaying RGB Components of color Image and combing to form RGB Image.

%Program for RGB plane

```
RGB=imread('C:\Program Files\MATLAB\R2010a\toolbox\images\imdemos\peppers.png');
R=RGB;
G=RGB;
B=RGB;
R(:,2)=0;
R(:,3)=0;
G(:,1)=0;
G(:,3)=0;
B(:,1)=0;
B(:,2)=0;
subplot(2,2,1),imshow(RGB),title('original image')
subplot(2,2,2),imshow(R),title('Red Component')
subplot(2,2,3),imshow(G),title('Green Component')
subplot(2,2,4),imshow(B),title('Blue Component')
```

%Program for RGB plane removal

```
clc
clear all
close all
a=imread('C:\Program Files\MATLAB\R2010a\Images\fl1.bmp');
a1=a;
b1=a;
c1=a;
a1(:,1)=0;
b1(:,2)=0;
c1(:,3)=0;
imshow(a),title('original image')
figure,imshow(a1),title('Red Missing!')
figure,imshow(b1),title('Green Missing!')
```



```
figure,imshow(c1),title('Blue Missing!')
```

```
%This program performs median filtering of the colour image
```

```
clc  
clear all  
close all  
a=imread('C:\Program Files\MATLAB\R2010a\Images\f1.bmp');  
b=imnoise(a,'salt & pepper',0.2);  
c(:,:,1)=medfilt2(b(:,:,1));  
c(:,:,2)=medfilt2(b(:,:,2));  
c(:,:,3)=medfilt2(b(:,:,3));  
imshow(a),title('original image')  
figure,imshow(b),title('corrupted image')  
figure,imshow(c),title('Median filtered image')
```

```
%Color Image segmentation
```

```
clc  
clear all  
close all  
a=imread('C:\Program Files\MATLAB\R2010a\Images\Tomato2.jpg');  
%Conversion of RGB to YCbCr  
b=rgb2ycbcr(a);  
%Threshold is applied only to Cb component  
mask=b(:,:,2)>120;  
imshow(a),title('original image')  
figure,imshow(mask),title('Segmented image')
```

11. Quiz on the subject:

Quiz should be conducted on tips in the laboratory, recent trends and subject knowledge of the subject. The quiz questions should be formulated such that questions are normally are from the scope outside of the books. However twisted questions and self formulated questions by the faculty can be asked but correctness of it is necessarily to be thoroughly checked before the conduction of the quiz.

12. Conduction of Viva-Voce Examinations:

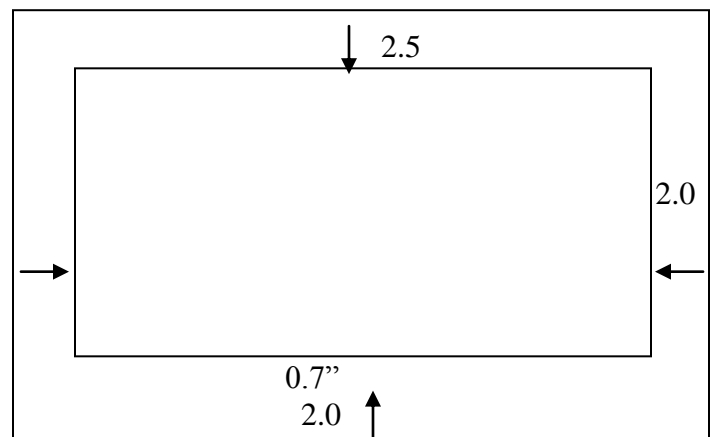
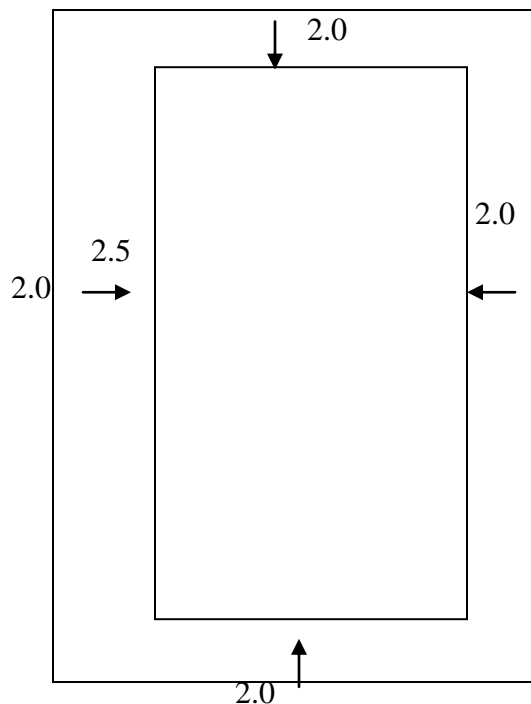
Teacher should oral exams of the students with full preparation. Normally, the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the students in the subject is tested Oral examinations are to be conducted in co-cordial environment amongst the teachers taking the examination. Teachers taking such examinations should not have ill thoughts about each other and courtesies should be offered to each other in case of difference of opinion, which should be critically suppressed in front of the students.

13. Submission:

Document Standard:

- | | |
|--|----------------|
| A] Page Size | A4 Size |
| B] Running text | Justified text |
| C] Spacing | 1 Line |
| D] Page Layout and Margins (Dimensions in Cms) | |
- Normal Page

Horizontal



Desription	Font	Size	Boldnes s	Italics	Underline	Capitalize
College Name	Arial	24	-----	-----	Yes	-----
Document Title	Tahoma	22	-----	-----	-----	-----
Document Subject	Century Gothic	14	-----	-----	-----	Capital
Class	Bookman Slyle	old 12	-----	-----	-----	-----
Document No	Bookman Slyle	old 10	-----	-----	-----	-----
Copy write inf	Bookman Slyle	old 9	-----	-----	-----	-----
Forward heading	Bookman Slyle	old 12	-----	-----	Yes	Capital
Forward matter	Bookman Slyle	old 12	-----	-----	-----	-----
Lab man Contents title	Bookman Slyle	old 12	-----	-----	Yes	Capital
Index title	Bookman Slyle	old 12	Yes	-----	Yes	Capital
Index contents	Bookman Slyle	old 12	-----	-----	-----	-----
Heading	Tahoma	14	Yes	Yes	Yes	-----
Running Matter	Comic Sans MS	10	-----	-----	-----	-----

14. Evaluation and marking system:

Basic honesty in the evaluation and marking system is absolutely essential and in the process impartial nature of the evaluator is required in the examination system to become popular amongst the students. It is a wrong approach or concept to award the students by way of easy marking to get cheap popularity among the students to which they do not deserve. It is a primary responsibility of the teacher that right students who are really putting up lot of hard work with right kind of intelligence are correctly awarded.

The marking patterns should be justifiable to the students without any ambiguity and teacher should see that students are faced with unjust circumstances.