LAB MANUAL ON

# IMAGE PROCESSING USING MATLAB

PREPARED BY
PICKON MAJUMDAR

**Program -1: Reading and Displaying Image, adding & removing Noise, Histogram, Mean & Standard Deviation on Image**

```
%% Reading and Displaying Image
i=imread('onion.png');  %% Image at current folder
imshow(i);
figure();
imshow(i(:,:,1));
figure();
imshow(i(:,:,2));
figure();
imshow(i(:,:,3));

%% Adding Noise to an Image
i1=imnoise(i,'salt & pepper',0.1);
figure();
imshow(i1);

%% Removing Noise
i2(:,:,1)=medfilt2(i1(:,:,1),[3 3]);
i2(:,:,2)=medfilt2(i1(:,:,2),[3 3]);
i2(:,:,3)=medfilt2(i1(:,:,3),[3 3]);
figure();
imshow(i2);

%% Histogram
i3=rgb2gray(i);
figure()
imhist(i3)

%% Mean And Standard Deviation
Mean=mean2(i3);
SD=std2(i3);

%% Black n white image
i4=im2bw(i3);
figure()
imshow(i3)

%% Edge Detection
i5=edge(i3,'sobel');
figure(); imshow(i5)
```

**Program -2: Reading an image, converting to Gray Scale, Edge Detection, Rotation and Translation of an Image**

```
x=imread('onion.png');
figure('Name','Original Image','Numbertitle','off');
imshow(x);
x=imread('onion.png');

%Gray scale image from RGB image
y=rgb2gray(x);
figure('Name','gray scale image','Numbertitle','off');
imshow(y);

%Edge Detection : canny /sobel/prewitt
w=edge(y,'canny');
figure('Name','Edge detection','Numbertitle','off');
imshow(w);

% Rotation of an Image
I(:,:,1)=imrotate(x(:,:,1),45);
I(:,:,2)=imrotate(x(:,:,2),45);
I(:,:,3)=imrotate(x(:,:,3),45);
figure('Name','Rotation of image','Numbertitle','off');
imshow(I);

% Translation of an Image
I1=imtranslate(x,[2.5 5.5]);
figure('Name','Translate image','Numbertitle','off');
imshow(I1);
figure('Name','Histogram image','Numbertitle','off');
imhist(y);
```

**Program-3: Reading an image, converting to HDR image**

```
hdr_image = hdrread('office.hdr');
imshow(hdr_image)
rgb = tonemap(hdr_image);
imshow(rgb)
subplot(1,2,1),imshow(hdr_image)
subplot(1,2,2),imshow(rgb)
```

**Program-4: Reading an image, converting to HDR image**

```
I=imread('autumn.tif'); %Read in image

Ihsv=rgb2hsv(I); %Convert original to HSV image, I2
imshow(Ihsv);

V=histeq(Ihsv(:,:,3)); %Histogram equalise V (3rd) channel of I2
Ihsv(:,:,3)=V; %Copy equalized V plane into (3rd) channel I2

Iout=hsv2rgb(Ihsv); %Convert I2 back to RGB form
subplot(1,2,1), imshow(I);
subplot(1,2,2), imshow(Iout);
```

**Program-5: Mixing noise with an image and filtering of an image**

```
clc;
clear all;
close all;
I1=imread('peppers.png');

% Original Image Without Noise
figure('Name','Original Image Without
Noise','Numbertitle','on');
imshow(I1);

%Mixing Noise With Image
I=imnoise(I1,'salt & pepper',0.09);
figure('Name','Mixing Noise With
Image','Numbertitle','on');
imshow(I);

% Filtering image
x(:,:,1)=medfilt2(I1(:,:,1),[3 3]);
x(:,:,2)=medfilt2(I1(:,:,2),[3 3]);
x(:,:,3)=medfilt2(I1(:,:,3),[3 3]);
figure('Name','filt','Numbertitle','on');
imshow(x);
```

**Program-6: Mixing noise with an image and filtering of an image**

```
I= imread('coins.png');
% Create filter.
h = ones(5,5) / 25;
% Apply filter to image using imfilter.
I2 = imfilter(I,h);
% Display original image and filtered image for
comparison.
imshow(I), title('Original Image');
figure, imshow(I2), title('Filtered Image')
% subplot(1,2,1),imshow(I)
% subplot(1,2,2),imshow(I2)
```

**Program-7: Simulation and Display of an Image, Negative of an Image (Binary & Gray Scale)**

```
i=imread('onion.png');
subplot(3,2,1); imshow(i); title('Original Image');
r=i(:,:,1);
subplot(3,2,2); imshow(r);title('Red Component');
%Green Component
g=i(:,:,2);
subplot(3,2,3); imshow(g); title('Green Component');
%Blue Component
b=i(:,:,3);
subplot(3,2,4); imshow(b); title('Blue Component');
%Color to Gray Image
rg=rgb2gray(i);
subplot(3,2,5); imshow(rg); title('Gray Image');
```

**% Display color Image, find its complement and convert to gray scale**

```
I=imread('onion.png');
subplot(2,2,1); imshow(I);
subimage(I); title('Color Image');
c=imcomplement(I); subplot(2,2,2); imshow(c); subimage(c);
title('Complement of color Image'); r=rgb2gray(I);
subplot(2,2,3); imshow(r);
subimage(r); title('Gray scale of color Image');
%Complement of Gray Image
b=imcomplement(r); subplot(2,2,4); imshow(b); subimage(b);
title('Complement of Gray Image');
```

```
%Simulation of an Image( Arithmetic & Logic Operation)
a=ones(40);
b=zeros(40);
c=[a b;b a];
d=[b b;a a];
A=10*(c+d); M=c.*d; S=c-d; D=c/4;
figure; subplot(3,2,1);
imshow(c); subplot(3,2,2); imshow(d);
subplot(3,2,3); imshow(A);
subplot(3,2,4); imshow(M);
subplot(3,2,5); imshow(S);
subplot(3,2,6); imshow(D);
```

**Program-8: Implementation of Transformations of an Image**
**%Scaling & Rotation**

```
% Scaling (Resize)
I=imread('onion.png');
subplot(2,2,1); subimage(I); title('Original Image');
s=input('Enter Scaling Factor');
j=imresize(I,s); subplot(2,2,2); subimage(j);
title('Scaled Image');

% Rotation
K=imrotate(j,60);
subplot(2,2,3); imshow(K); title('Rotated Image 60deg');
R=imrotate(j,45);
subplot(2,2,4); imshow(R); title('Rotated Image 45deg');

%Display the color image and its Resized images by
different methods

%Display the color image
I=imread('embryo.jpg');
figure, subplot(2,2,1);
subimage(I); title('Original Image');

%Display Resized image by Bilinear method
B=imresize(I,5); subplot(2,2,2); subimage(B);
title('Bilinear Image');
```

```
%Display Resized image by Nearest method
C=imresize(I,5,'nearest'); subplot(2,2,3); subimage(C);
title('Nearest Image');

%Display Resized image by Bicubic method
D=imresize(I,5,'Bicubic');
subplot(2,2,4); subimage(D); title('Bicubic Image');
```

**Problem -9: Contrast stretching of a low contrast image, Histogram, and Histogram Equalization**

```
% Image Enhancement
I=imread('onion.png');
subplot(4,2,1); imshow(I); title('Original Image');
g=rgb2gray(I);
subplot(4,2,5); imshow(g); title('Gray Image');
J=imadjust(g,[0.3 0.7],[]);
subplot(4,2,3); imshow(J); title('Enhanced Image');
D= imadjust(I,[0.2 0.3 0; 0.6 0.7 1],[]);
subplot(4,2,4);imshow(D);title('Enhanced Image 2');

% Histogram and Histogram Equalization
subplot(4,2,7);
imhist(g); title('Histogram of Gray Image');
m=histeq(g);
subplot(4,2,6); imshow(m); title('Equalized Image');
subplot(4,2,8); imhist(m); title('Histogram of Equalized
Image');
```

**Problem -10: Display of FFT (1-D & 2-D) of an image**

```
l=im2double(imread('onion.png'));
f1=fft(l); f2=fftshift(f1);
subplot(2,2,1); imshow(abs(f1));
title('Frequency Spectrum');
subplot(2,2,2); imshow(abs(f2));
title('Centered Spectrum');
f3=log(1+abs(f2));
subplot(2,2,3); imshow(f3); title('log(1+abs(f2))');
l=fft2(f1); l1=real(l);
subplot(2,2,4); imshow(l1);title(' 2-D FFT');
```

**Problem -11: Computation of mean, Standard Deviation, Correlation coefficient of the given Image**

```
i=imread('onion.png');
subplot(2,2,1); imshow(i);title('Original Image');
g=rgb2gray(i); subplot(2,2,2); imshow(g);title('Gray
Image');
c=imcrop(g); subplot(2,2,3); imshow(c);title('Cropped
Image');

m=mean2(c);disp('m'); disp(m);
s=std2(c); disp('s'); disp(s);
figure, k=(checkerboard>0.8); subplot(2,1,1); imshow(k);
title('Image1');
k1=(checkerboard>0.5);
subplot(2,1,2); imshow(k1); title('Image2');
r=corr2(k,k1); disp('r');disp(r);
```

**Problem -12: Implementation of Image Intensity slicing technique for image enhancement**

```
i=imread('onion.png');
subplot(3,2,1);imshow(i); title('Original Image');
l=im2double(i);
level=graythresh(l);
BW = im2bw(l,level);
subplot(3,2,2); imshow(BW); title('Image graythresh');
level1=0.2*BW;
subplot(3,2,3); imshow(level1); title('0.2 Slice');
level2=0.4*BW;
subplot(3,2,4); imshow(level2);title('0.4 Slice');
level3=0.6*BW;
subplot(3,2,5); imshow(level3);title('0.6 Slice');
level4=0.8*BW;
subplot(3,2,6); imshow(level4); title('0.8 Slice');
```

**Problem -13: Implementation of Image Smoothening Filters(Mean and Median filtering of an Image)**

```
% Median Filters I=imread('onion.png');
K = rgb2gray(I);
J= imnoise(K ,'salt & pepper',0.05);
f= medfilt2(J,[3,3]);
f1=medfilt2(J,[10,10]);
subplot(3,2,1); imshow(I); title('Original Image');
subplot(3,2,2); imshow(K); title('Gray Image');
subplot(3,2,3); imshow(J); title('Noise added Image');
subplot(3,2,4); imshow(f); title('3x3 Image');
subplot(3,2,5); imshow(f1); title('10x10 Image');

%Mean Filter and Average Filter figure;
i=imread('onion.png'); g=rgb2gray(i);
g1=fspecial('average',[3 3]);
b1 = imfilter(g,g1);
subplot(2,2,1); imshow(i); title('Original Image');
subplot(2,2,2); imshow(g); title('Gray Image');
subplot(2,2,3); imshow(b1); title('3x3 Image');
g2= fspecial('average',[10 10]);
b2=imfilter(g,g2);
subplot(2,2,4); imshow(b2); title('10x10 Image');

%Implementation of filter using Convolution
figure;
I= imread('onion.png');
I=I(:,:,1); subplot(2,2,1);
imshow(I); title('Original Image');
a=[0.001 0.001 0.001; 0.001 0.001 0.001; 0.001 0.001 0.001];
R=conv2(a,I); subplot(2,2,2); imshow(R);
title('Filtered Image');
b=[0.005 0.005 0.005; 0.005 0.005 0.005; 0.005 0.005 0.005];
R1=conv2(b,I); subplot(2,2,3); imshow(R1);
title('Filtered Image 2');
```

**Problem -14: Implementation of image sharpening filters and Edge Detection using Gradient Filters**

```
i=imread('onion.png');
subplot(4,2,1); imshow(i);
title('Original Image');
g=rgb2gray(i);
subplot(4,2,2); imshow(g); title('Gray Image');
f=fspecial('laplacian',0.05);
im=imfilter(g,f);
subplot(4,2,3); imshow(im); title('Laplacian ');
s=edge(g, 'sobel');
subplot(4,2,4); imshow(s); title('Sobel');
p=edge(g, 'prewitt');
subplot(4,2,5); imshow(p); title('Prewitt');
r=edge(g, 'roberts');
subplot(4,2,6); imshow(r); title('Roberts');
[BW,thresh,gv,gh]=edge(g,'sobel',[],'horizontal');
[BW1,thresh1,gv1,gh1]=edge(g,'sobel',[],'vertical');
subplot(4,2,7); imshow(BW); title('Sobel Horizontal');
subplot(4,2,8);
imshow(BW); title('Sobel Vertical');
```

**Problem 15:  Canny edge detection Algorithm**

```
i= imread('onion.png');
g=rgb2gray(i);
subplot(2,2,1); imshow(i); title('Original Image');
subplot(2,2,2); imshow(g); title('Gray Image');
c=edge(g,'canny');
subplot(2,2,3); imshow(c); title('Canny output');
```