LAB FILE

Digital Image Processing- BCSE0131

Submitted to:

Ankush Agarwal

Submitted by:

Surendra Tomar

Uni. Roll no:

191500831Section: J2

Roll no: 40

Experiment- 01

Create command to familiarize with MATLAB & Create the matrix & perform the various operations on them.

```
Variable = 2;
%% Displaying the variable using disp() function. %%
disp(Variable);

%% Creating an 3x3 Matrix %%
Matrix = [4 5 2; 3 5 7; 4 5 7]; %% Here Space Seprated Columns and
Semicolons Seprated Rows. %%

%% Finding the size of the matrix using size function. %%
size(Matrix) %% Output: 3x3 %%

%% Finding the number of rows in the matrix using length funcion. %%
length(Matrix) %% Output: 3 %%

%% Sorting the matrix using sort funcion. %%
sort(Matrix)
%% Sorting the rows %%
sortrows(Matrix)

whos('Matrix') %% "whos" function is used to show the details of the
matrix in a tabular form. %%

Matrix(1,:) %% Output: 4 5 2 %%
Matrix(1:2,:) %% Output: 4 5 2 ; 4 5 7

%% Generating a rsndom matrix using rand() function. %%
rand(5,7) %% All the elements will be between 0 and 1 inclusive. %%

 "clc" %% is a teminal command used to clear the screen. %%

abs(Matrix) %% Output: 'Returns absolute value.' %%
sin(Matrix) %% Output: 'Sine values matrix form.' %%
floor(4.3) %% Output: 4 %%
ceil(4.3) %% Output: 5 %%
min() %% Output: Minimum value of each column in the form of matrix. %%
max() %% Output: Maximum value of each column in the form of matrix. %%

imread() %% is used to read the image. %%
imshow() %% is used to display the image. %%
```

Experiment- 02

Understanding Image Basics "image Resize, image type conversion, extraction of color band, creating a synthesic image, psdeocolor image.

```
%% Reading, resizing of the image. %%

a = imread ('C:\Users\CL235\Desktop\A\Sunflower.jfif');
imshow(a) %% Output: Showing Image. %%
length(a) %% Output: 303 %%

mygray_image = rgb2gray(a); %% rgb2gray() function is used to convert RGB
image to grayscale image. %%
mypic = im2bw(a); %% im2bw() is used to convert image to black and white
image. %%

myred_image = a(:, :, 1) %% Output: this will give red channel of the
image. %%
mygreen_image = a(:, :, 2) %% Output: this will give green channel of the
image. %%
myblue_image = a(:, :, 3) %% Output: this will give blue channel of the
image. %%

O = cat(3, myred_image, mygreen_image, myblue_image); %% cat() is used to
concatenate the three channels of the image and output the Original image.
%%

%% Ploting the images into matrix %%

subplot(2,2,1);
imshow(a);
title('Original Coloured Image');

subplot(2,2,2);
imshow(mygray_image);
title('Gray Image');

%% Resizing the image using imresize() function. %%

resized_image = imresize(a,[256,256]);
imshow(resized_image)
```

Experiment- 03

Perform various Arithmetic Operation (Image Addition & Complement) & Logical Operation (NOT, OR and XOR) on images.

```
%%%Perform the Arithimetic and Logical Operations on the images.%%%

%First Image
a = imread('C:\Users\CL235\Desktop\A\Sunflower.jfif');

%Second Image
b = imread('C:\Users\CL235\Desktop\loin.jpg');


%Resizing Image to the same size
resizeA = imresize(a,[256,256]);
resizeB = imresize(b,[256,256]);

%Addition Of two Images
additionofab = imadd(resizeA, resizeB);

%Showing the result Image
imshow(additionofab)

%Reducing intensity of an image applying by

%%%%"Arithimetic Operations"%%%

%Addition(+)
BrightA = imadd(a, 50);
imshow(BrightA)

%Subtraction(-)
DarkA = imsubtract(a, 50);
imshow(DarkA)

%Multiplication(*)
BrighterA = immultiply(a, 2);
imshow(BrighterA)

%Division(/)
DarkerA = imdivide(a,2);
imshow(DarkerA)

%%%Ploting of Images%%%
subplot(4,4,1);
imshow(a);
title('Original Image')

subplot(4,4,2);
imshow(b);
title('Second Original Image')
```

```
subplot(4,4,3);
imshow(additionofab);
title('Result of a + b')

%%%Implementing Logical Operations%%%

%Black And White(im2bw)
Bw = im2bw(a);
subplot(4,4,4);
imshow(Bw);
title('Black And White')

%Complement(~)
comp = imcomplement(a);
subplot(4,4,5);
imshow(comp);
title('Complemented Image of image a')

%AND(&)
operation1 =(a&b);
subplot(4,4,6);
imshow(operation1);
title('AND Image')

%OR(|)
operation2 = (a|b);
subplot(4,4,7);
imshow(operation2);
title('OR Image')

%NOT
operation3 = not(a,b);
subplot(4,4,8);
imshow(opration3);
title('NOT Image')
```

Experiment- 04

Perform various Histogram Operations histogram piot, histogram Equalization, Contrast Streatching & gamma correction on images & piot histogram without using imhist function.

```
A=imread('peppers.png');
%%% Convert RGB 2 GRAY %%%
gA=rgb2gray(A)
%%% Get histogram of the image %%%
imhist(gA);
subplot(4,2,1);
title('Histogram')

subplot(4,2,2);
imshow(gA);
title('Histogram Image')

%%% Contrast Streching %%%
adj=imadjust(A, [.2, .6], [0, 1])
gadj=rgb2gray(adj);

%%% Streched %%%
subplot(4,2,3);
imhist(gadj);
title('Streched Histogram')

subplot(4,2,4);
imshow(gadj);
title('Streched Histogram image')

%%% Second Way StrechLimit %%%
s=stretchlim(A);
badj=imadjust(A,s);
gbadj=rgb2gray(badj);

%%% Streched %%%
subplot(4,2,5);
imhist(gbadj);
title('Streched Histogram using Strechlim')

subplot(4,2,6);
imshow(gbadj);
title('Streched Histogram image using Strechlim')

%% Histogram Equalization %%
equalisedImage=histeq(gA);
subplot(4,2,7);
imhist(equalisedImage);
title('Streched Histogram using Strechlim')

subplot(4,2,8);
```

```
imshow(equalisedImage);
title('Streched Histogram image using Strechlim')
```

Experiment- 05

Perform smoothing using linear and order statistics filters min, max & med of verifying sizes and Sharpen an image using Laplacian filter.

```
%%%% Sharpening image %%%

i1 = imread('C:\Users\Ghanshyam Verma\Downloads\mountains.jpg');
%imshow(i1)
i1 = rgb2gray(i1);
%imshow(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); %%%Sharpening%%%
    end
end
s(i,j) = sum;
    end
end
st = i1-s;

subplot(4 ,4,1);
imshow(uint8(i1));
title('Image1');

subplot(4 ,4,2);
imshow(uint8(s));
title(' Image2');

subplot(4 ,4,3);
imshow(uint8(st));
title('Image3');


%%% Smoothing image using Linear filters. %%%

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);%%%
Smoothing the image 'i2'
            end
        end
        i2(i,j) = sum / (size^2);
    end
end

subplot(2,2,1);
imshow((i1));
title('Org. Gray Image');

subplot(2,2,2);
imshow((i2));
title('Smooth Image');
```

Experiment- 06

Perform various Fast Fourier Transform (FFT) and frequency domain filtering on image using MATLAB.

```
%%% Fourier transformation using zeros mask. %%%


f = imread('C:\Users\CL235\Desktop\Cat1.tif');
%f = rgb2gray(f);
%imshow(F)
[m,n] = size(f);

%(1)
mask = zeros(m,n);
for i = 150:180
    for j = 210:240
        mask(i,j) = 1;
    end
end

%(2)
b = fft2(f);%fourier transform
c = fftshift(mask);%lowpass filter
%(3)
d = b.*c;
e = abs(ifft2(d));

subplot(2,2,1);
imshow(f);
title('Original');

subplot(2,2,2);
imshow(e);
title('Inversed');

subplot(2,2,3);
imshow(mask);
title('Mask');

subplot(2,2,4);
imshow(c);
title('Lowpass Filter');


%%% Frequency Domain Filter %%%

F = imread('C:\Users\CL235\Desktop\Cat.jfif');
%imshow(F)
%(1)
ff = fft2(F);
Cff  = log(1 + abs(ff));
```

```
%(2)
sff = fftshift(ff);
lsff = log(1 + abs(sff));

%(3)
Or = real(ifft2(ff));

subplot(2,2,1);
 imshow(F);
 title('Original image');

subplot(2,2,2);
 imshow(Cff,[]);
 title('Fourier image');

 subplot(2,2,3);
 imshow(lsff,[]);
 title('lsff');

 subplot(2,2,4);
 imshow(Or,[]);
 title('Original');
```

Experiment- 07

Perform various Morphological operation dilation, erosion, internal & external boundary Extraction, Thinning, Thickening & Skeletionziation of Image & Perform Dilation, erosion, boundary Extraction without using direct direct function.

```
% Morphology is used to find the structure of the image.
% Function used in morphology are:
% (1) Thinning
% Syntax: thinmorph = bwmorph(name_of_image,'thin');

% (2) Thickening
% Syntax: thickmorph = bemorph(name_of_image,'thicken';

% (3) Skeletion
% Syntax: imgskel = bwmorph(name_of_image, 'skel', iteration)

%%% Note: Morphology always apply only on Black & White Images.
```

```
myorigimg = imread('C:\Users\CL235\Desktop\Atlas.png');
myorigimg = im2bw(myorigimg);

%Thinning
thinmorph = bwmorph(myorigimg,'thin');

%Thickening
thickmorph = bwmorph(myorigimg,'thicken');

%Skeletion
imgskel = bwmorph(myorigimg, 'skel',100);

subplot(3,3,1);
imshow(myorigimg);
title('Original image');

subplot(3,3,2);
imshow(thinmorph);
title('Thin image');

subplot(3,3,3);
imshow(thickmorph);
title('Thick image');

subplot(3,3,4);
imshow(imgskel);
title('Skeleton image');


% Perform Dilation operation using imerode command
% Read the test Image
% Convert the image to binary image
myorigimg = imread('C:\Users\CL235\Desktop\Atlas.png');
myorigimg = im2bw(rgb2gray(myorigimg));

subplot(3,3,1);
imshow(myorigimg);
title('Original image');


% Dilation Operation

% Create Structuring Element
se = strel('disk', 6);
mydilatedimg = imdilate(myorigimg, se);

subplot(3,3,2);
imshow(mydilatedimg);
title('Dilated image');

%Erosion operation
```

```
% Create Structuring Element
se1 = strel('disk', 6);
myerotedimg = imerode(myorigimg, se1);

subplot(3,3,3);
imshow(myerotedimg);
title('Eroted image');

% Subtracting Original Image from Dilated Image
sub = imsubtract(mydilatedimg,myorigimg);

% Subtracting Eroted Image from Original Image
sub1 = imsubtract(myorigimg, myerotedimg);

subplot(3,3,4);
imshow(sub);
title('Dilated image - Original image');

subplot(3,3,5);
imshow(sub1);
title('Original image - Eroted image')
```

<u>Experiment- 08</u>

Perform various thresholding semgmentation (Simple, Multiple, Adaptive & Optimal thresholding).

```
%% Segmentation using thresholding %%
%% There are 4 types of thresholding:
% (1)Simple thresholding
% (2)Multiple thresholding
% (3)Adaptive thresholding
% (4)Optimal thresholding

% (1) Simple thresholding(Done)


% (2) Multiple thresholding(Done)

a = rgb2gray(imread('C:\Users\CL235\Desktop\Exp.png'));

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(3,3,1);
imshow(a);
title('Original Image');

subplot(3,3,2);
Segimg = im2bw(tmp,0);
imshow(Segimg);
title('SegImg');
```

Experiment- 09

Perform the various Edge Detection Operations (Ordinary, Robert, Prewitts and Sobel Operator).

```
% Finding an edge in image using Roberts, Prewitt and Sobel Mask.
k=imread('C:\Users\CL235\Desktop\Me.png');

% Define the Laplacian filter.
Robertx =[1 0; 0 -1];
Roberty =[0 1; -1 0];

Prewittx=[-1 0 1; -1 0 1; -1 0 1];
Prewitty=[1 1 1; 0 0 0; -1 -1 -1];

Sobelx=[-1 -2 -1; 0 0 0; 1 2 1];
Sobely=[-1 0 1; 2 0 2; -1 0 1];

% imfilter the image using Laplacian Filter
k2=imfilter(k, Robertx);

k3=imfilter(k,Prewittx);

k4=imfilter(k,Sobelx);

k5=imfilter(k, Roberty);

k6=imfilter(k,Prewitty);
```

```
k7=imfilter(k,Sobely);

k8=imfilter(k, Robertx + Roberty);

k9=imfilter(k,Prewittx + Prewitty);

k10=imfilter(k,Sobelx+Sobely);

% Display the image.
subplot(3,4,1);
imshow(k, []);
title('Original Image');

subplot(3,4,2);
imshow(k2, []);
title('Robertx Image');

subplot(3,4,3);
imshow(k5, []);
title('Roberty Image');

subplot(3,4,4);
imshow(k8, []);
title('Robertx + Roberty Image');

subplot(3,4,5);
imshow(k3, []);
title('Prewittx Image');

subplot(3,4,6);
imshow(k6, []);
title('Prewitty Image');

subplot(3,4,7);
imshow(k9, []);
title('Prewittx + Prewitty Image');

subplot(3,4,8);
imshow(k4, []);
title('Sobelx Image');

subplot(3,4,9);
imshow(k7, []);
title('Sobely Image');

subplot(3,4,10);
imshow(k10, []);
title('Sobelx + Sobely Image');
```

Experiment-10

Perform the Extraction of image features and Specification in MATLAB

Case Study: Calculate the total No. of coins & diameter of coin in image.

```
%%%%%%%%%%%%%%%%
Amount = 0;
% Read the image
Coins = imread('Coins4.jpeg');
%Coins = imresize(Coins,0.3);
CoinsBinary = im2bw(Coins,0.15);
se = strel('disk',20);
open=imopen(CoinsBinary,se);
fill = imfill(open,'holes');
clearing = imclearborder(fill);
figure
imshow(clearing)

%*** Automatic measuring of the diameter ****
diameter = regionprops(clearing,'MinorAxisLength');
A_cell = struct2cell(diameter);
for j = 1:length(diameter)
r(j) = A_cell{j};
end
radius =r/2;
rMin = int32(min(radius));
rMax = int32(max(radius));
[centers, radii] = imfindcircles(clearing,[rMin
rMax],'ObjectPolarity','bright','Sensitivity',0.98,'EdgeThreshold',0.85);
numCircles = length(centers)

imshow(Coins)
h = viscircles(centers,radii);

% Just for checking if it works to calculate the sum like this....
for i = 1:numCircles
if radii(i) > rMin && radii(i) < 140
    Amount = Amount + 0.1;
elseif radii(i) > 140 && radii(i) < 150
    Amount = Amount + 0.2;
elseif radii(i) > 150 && radii(i) < 160
    Amount = Amount + 1;
elseif radii(i) > 160 && radii(i) < 170
    Amount = Amount + 0.5;
elseif radii(i) > 170 && radii(i) < rMax
    Amount = Amount + 2;
end
end
fprintf('Total amount of coins: %.2f €', Amount);
```