Name :- Surendra Pratap Toman Section - J2 Roll no - 40

Section Roll No :- 40 Uni Roll :- 19150083 1

Lab - File Digital Image processing practical file.

B.Tech (III rd) Year → Student.

Objective : Introduction with the matlab and matlab commonds :

Theory :

MATLAB is an abbreviation for matric Laboratory, while other programing languages work with numbers on at a time MATLAB operates on whole matrices and array language fundamentals include basic operation such as creating variables, array etc.

clc - clear Commond window.

clear all - delete variable from workspace.

quit - Stop MATLAB

who - List current variables.

function - purpose

char - convert to char array.

1-B - Answer. 19.04.22

Basic Operations :

Input A = 50 , B = 2

operation	output
A - B	48
A + B	52
A * B	100
A / B	25
A ^ B	2500

Matrix

```
a = [1,2,3; 4,5,6; 7,8,9]
```

output

```
a =
   1   2   3
   4   5   6
   7   8   9.
```

for Loop:-

```
A = [3 7 8 5]

for i=1: length(A)
    disp(A(i));
end
```

output:

```
3
7
8
5
```

Transpose of a matrix

```
A = [1,2,3; 4,5,6; 7,8,9].
B = A'
```

```
A =
   1   2   3
   4   5   6
   7   8   9

B =
   1   4   7
   2   5   8
   3   6   9
```

Inverse of a Matrix:-

```
a = [1,2,3; 4,5,6; 7,8,9]
b = inv(a)
```

```
a =
   1   2   3
   4   5   6
   7   8   9.

b =
  -0.4504   0.9007  -0.4504
   0.9007  -1.8014   0.9007
  -0.4504   0.9007  -0.4504
```

Determinant:

```
a = [1,2,3; 4,5,6; 7,8,9]
b = det(a)
```

output

```
a =
   1   2   3
   4   5   6
   7   8   9

b = -2.
```

Practical: 2

Roll no - 40

Objective - 2 Use of plot and sub.plot functions by using the linspace commond.

Code + theory

plot: It creates a 2-D line plot of data in Y-axis the corresponding values in X

Subplot (m,n,p):- It divide the current figure into mxn grid and creates axis in the position specified by P.

Linespace:- It is similar to the Calon operator, but give direct control over the number of points and always includes the end points.

plot →

```
x = linspace (-2*pi, 2*pi);
y = sin(x);
plot (x,y);
x = linspace (-2*pi, 2*pi);
y1 = sin(x);
y2 = cos(x);
figure.
plot (x, x1, x, y2)
```

Subplot :→

```
subplot (2, 1, 1);
x = linspace (0, 1, 0);
y1 = sin (x);
plot (x, y1);
subplot (2, 1, 2);
y2 = sin (5*x);
plot (x, y2);
```

Practical-3

Roll no 40

Objective-3 Read a image, extract the bands and perform the TCC and FCC

Code:-

```
img = imread ('D:\img.jpeg');
imshow (img);
b1 = img (:,:,1);
b2 = img (:,:,2);
b3 = img (:,:,3);
imshow (b1);
imshow (b2);
imshow (b3);
FCC = cat (3, b3, b2, b1);
imshow (FCC);
```

Objective-4: (i) Alternate pixel is zero (read the image)

(ii) Alternate row and column is zero

Code: (i) Alternative pixel to 0:

```
im = imread ('D:\img.jpeg');
for i=1: size (im,1):
    for j=1 : size (im,2):
        if ( mod ((i+j),2) ==0)
            im (i, j, :) = 0;
        end
    end
end
imshow (im);
```

(ii) Alternative column and row to 0 →

```
im = imread ('D:\img.jpeg');
for i=1 : size (im,1)
    for j=1 : size (im,2)
    if ( mod ((i,2)) == 0 || mod (j,2) == 0)
            im ( i, j, :) = 0;
    end
  end
end
imshow (im);
```

Practical-5

Object:-5: Implement the checker board effect by using.

(i) Imresize function

(ii) Without using Imresize function.

Code-Solution:- (i) Implementing by using imresize() function:

```
im = imread ('D:\img.jpeg');
X = imresize (im, [100 NAN]);
Subplot (1,2,1);
imshow (im);
Subplot [1,2,2];
imshow (X);
```

(ii) Without using Imresize function:-

```
a = imread ['D:\img.JPEG'];
scale = [10,10];
alds = size(a);
rows = max [floor [scale.* alds (1:2)), 1);
cal = min [round ((( 1: news(2)) -0.5) . /scale(2)+0.5), alds (2));
rall = min [round (((1: newll (1)) -0.5). /scale(1)+0.5), alds (1)];
z = a [row, cal, :);
subplot [1,2,1);
imshow (a);
Subplot [1,2,2);
imshow (z);
```

Practical :- 6

Roll no-40

Objective : To flip the image in vertical and horizontal direction by using :-

(i) flip command (ii) without using flip command

Code :

(i) :

```
a = imread ('D:\img.jpeg');
V = flip (a,1);
H = flip (a,2);
subplot (1,3,1);
imshow (a); title ('Original');
subplot (1,3,2);
imshow (V); title ('vertical');
subplot (1,3,3);
imshow (H); title ('Horizontal');
```

(ii) without flip ();

```
a = imread ('D:\img.jpeg');
X = size (a,1);
Y = size (a,2);
V = a;
H = a;
for i = 1 : X
    for j = 1 : Y
        V (X-i+1, j, :) = a (i, j, :);
        H (i, Y-j+1, :) = a (i, j, :);
    end
end
end.
Suplot (1,3,1);
```

Practical 7:

Roll no - 40

Objective: Extract the ROI by using threshold technique.

Code:-

```
im = imread ('cameraman.tif');
im = imresize (im, [256 256]);
imshow (im);
[cal row] = ginput (4);
c = cal;
r = row;
Binary mask = roipoly (im, c, r);
figure;
imshow (Binary mask); title ('selected ROI');
NONROI = zeros (256, 256);
ROI = zeros (256, 256);
for i=1 : 256
if (Binary mask (i, j) == 1)
ROI (i, j) = im (i, j);
else NONROI (i, j) = im (i, j);
end
end
end
figure,
subplot (1, 2, 1);
imshow (ROI, []); title ('ROI');
subplot (1, 2, 2);
imshow (NONROI, []);
title ('NONROI');
```

Practical - 8

Roll no - 40

Objective - 8 : perform the bitplane slicing and plot the planes of the image.

Code:-

```
i1 = imread('cameraman.tif');
P1 = bitget(i1,1);
P2 = bitget(i1,2);
P3 = bitget(i1,3);
P4 = bitget(i1,4);
P5 = bitget(i1,5);
P6 = bitget(i1,6);
P7 = bitget(i1,7);
P8 = bitget(i1,8);
subplot(3,3,1);
imshow(i1);
subplot(3,3,2);
imshow(logical(P1));
subplot(3,3,3);
imshow(logical(P2));
subplot(3,3,4);
imshow(logical(P3));
subplot(3,3,5);
imshow(logical(P4));
subplot(3,3,6);
imshow(logical(P5));
subplot(3,3,7);
```

Aim: To perform transform on Image

Implementation

Negative transformation:

```
im = imread ('cameraman.tif');
it = im;
y = zeros (255,255);
for i=1:255
    for j=1:255
        im (i,j) = 255 - im (i,j);
    end
end
y = im;
subplot (2,2,1);
imshow (i,1); title ('Before');
subplot (2,2,2);
imshow (y), title ('after Neg');
```

Log transformation:-

```
i1 = imread ('cameraman.tif');
im = i1;
c = 12.5;
for i=1:255
    for j=1:255
        y = cast (i1 (i,j), 'Double');
        i2 (i,j) = c * log (1+y);
    end
end
subplot (1,2,1);
imshow (im); subplot (1,2,2); imshow (i2);
```

Power transformation:-

```
i1 = imread ('cameraman.tif');
im = i1; i2 = i1; i3 = i1; c = 12.5, a = 1;
for i=1:255
    for j=1:255
```

Experiment - 10

Roll No - 40

Object:- To perform and implement Histogram

Implementation:-

```
im = imread ('Cameraman.tif');
it = zeros (256,2);
for i=1 : 256
    it (i,1) = i-1;
end
for i= 1: 256
    for j=1 : 256
        it (im (i,j)+1,2) = it (im (i,j)+1,2)+1;
end;
end;
subplot (1,2,1); bar (it) (:,1), it (:,2));
subplot (1,2,2) ; imhist (im);
```

Experiment -11 Roll no - 40

Objective:- To perform histogram Equalization.

implementation:

```
a = imread ('cameraman.tif');
b = zeros (256,7);
c = zeros (256,7);
for i=1:256
    b(i,1) = i-1;
end
  for j=1:256
    for i=1:256
      b(a(j,i)+1,2) = b(a(j,i)+1,2)+1;
  end
end
 for i=1:256
    b(i,3)=(b(i,2)/(256*256));
end
b(1,4) = b(1,3);
 for i=2:256
    b(i,4) = (b(i,3)+b(i-1,4));
 end
  for i=1:256
    b(i,5) = (b(i,4)*255));
  end.
  for i=1:256:
      b(i,6) = round (b(i,5));
  end.
for i=1:256
  for j=1:256
```

```
if (b(i,1) == b(j,6))
    b(i,7) = b(i,7) + b(j,2);
end
end
end

for k=1:256
  for i=1:256
    for j=1:256
      if (a(i,j) == (k-1))
        c(i,j) = b(k-1,6);
      end
    end
  end
end
c = cast(c,'uint8');
subplot(1,2,1);
bar(b(:,1), b(:,7));
subplot(1,2,2);
imhist(c);
```

Experiment - 12 Roll No - 40

Objective: To apply average, min, max, median, filters on an image.

Implementation 1) Average filter mask:

```
a = imread ('cameraman.tif');
a = double (a);
b = a;
[m,n] = size(a)
s = input ('Enter the size of mask = ');
f = ones (s);
c = (s+1)/2;
for i = c : m-c+1
    for j = c : n-c+1
        sum = 0;
        for k = 1:s
            for l = 1:s
                sum = sum + a (i-c+k, j-c+l) * f(k,l);
    end
end
b(i,j) = sum / (s*s);
    end
end
a = cast (a, 'uint8');
b = cast (b, 'uint8');
subplot (1,2,1);
imshow (a);
subplot (1,2,2);
imshow (b);
```

min, max, meadian filter

```
a = imread('cameraman.tif');
i1 = a; i2 = a; i3 = a;
[m,n] = size(a);
s = input('Enter mask = ');
f = zeros(s);
c = (s+1)/2;
for i=c: m-c+1
    for j= c: n-c+1
        for k=1:s
            for l= 1:s
                f(k,l)= a(i-c+k, j-c+l);
            end
        end
    i1(i,j) = meadian(f(:));
    i2(i,j) = min(f(:));
    i3(i,j) = max(f(:));
  end
end
subplot(1,3,1);
imshow(i1); title('meadian');
subplot(1,3,2);
imshow(i2); title('minimum');
subplot(1,3,3);
imshow(i3); title('maximum');
```

Objective: To perform Dilation on an image.

Implementation (without function).

```
a = imread ('cameraman.tif');
z = im2bw (a);
dif = zeros (256);
d = zeros (256);
[m,n] = size (dif);
mask = ones (3);
dif (2:257, 2:257) = z;
c = 2;
for i = c: m-c+1:
    for j = c: n-c+1
        for k = 1:3
            for l = 1:3
                if mask (k,l) == dif (i-c+k, j-c+l);
                    d (i-1, j-1) = 1;
                end
            end
        end
    end
end
subplot (1,2,1);
imshow (z); title ('orginal');
subplot (1,2,2);
imshow (d); title ('dilation');
```

With function;

```
a = imread ('cameraman.tif');
i = imshow (a);
mask = ones (3);
dil = imdilate (i, mask);
subplot (1,2,1);
```

Experiment - 15 Roll No - 40

Objective: To perform opening an closing on an image.

Implementation (opening):

```
a = imread ('cameramen.tif');
I = imshow (a)
ma = ones (3);
er = imdilate (er, ma);
op = imopen (I, ma);
Subplot (1,3,1);
imshow (i); tittle ('orginal');
Subplot (1,3,2);
imshow (dt); tittle ('opening without function');
Subplot (1,3,3);
imshow (op); tittle ('with function opening');
```

closing implementation:

```
a = imread ('cameramen.tif');
i = imshow (a);
ma = ones (3);
dt = imdilate (I, ma);
er = imerode (dt, ma);
closing = imclose (i, ma);
Subplot (1,3,1);
imshow (i); tittle ('orginal');
Subplot (1,3,2);
imshow (er); tittle ('without function closing');
Subplot (1,3,3);
imshow (closing); tittle ('with function closing');
```

Experiment 14 Roll No - 40

Objective:- To perform Erosion on an image

Implementation: (additional function):

```
a = imread ('Cameraman.tif');
z = im2bw (a);
dt = zeros (256);
d = ones (256);
[m,n] = size (dt);
ma = ones (3);
dt = (2:257, 2:257) = z;
c = 2;
for i=c : m-c+1
    for j=c:n-c+1
        for K=1:3
            for l=1:3
            if ma (K,l) ==1
            if in = dt (i-c+K, j-c+l)
                d (i-1, j-1) = 0;
            end
        end
        if 0 == dt (i-c+K, j-c+l)
        end d (i-1, j-1) = d (i-1, j-1);
    end end end end
subplot (1,2,1);
imshow (z); title ('orginal');
subplot (1,2,2);
imshow (d); title ('Erosion');
```