INTRODUCTION TO MATLAB

MATLAB is a software package

Commands to create 1-D Array :-

a = [1, 2, 3, 4, 5]

b = [1 2 3 4 5]

c = [1 : 5]

output of a, b, c matrix :-

a = 1 2 3 4 5

b = 1 2 3 4 5

c = 1 2 3 4 5

Command to create 1-D Array using step-size :-

d = start : step : end

Example :- • d = 1 : 02 : 10

output :- d = 1.0000 3.0000 5.0000 7.0000 9.0000

• e = 1 : 0.5 : 5

output :- e = 1.0000 1.5000 2.0000 2.5000 3.0000 3.5000 4.0000 4.5000 5.0000

• f = 5 : -1 : 1

output :- 5.0000 4.0000 — — — 1.0000

Command to create 2-D Array :-

1. a = [1, 2, 3 ; 4, 5, 6]

output :- a = 1 2 3
4 5 6

2- Magic Command — Magic (a) command is used to create a matrix of size axa with random numbers.

Example - magic (2)

output :- $\begin{bmatrix} 1 & 5 \\ 7 & 10 \end{bmatrix}$

Command to clear screen — clc

Command to clear workspace — clear all

Command for type Casting — i2 = cast (i2 : 'datatype');

Zeros command: zeros command is used to create a matrix with 0's

Example :- • zeros(2)

Output :- $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

• zeros(2,3)

output :- $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Ones command :- Ones command is used to create a matrix with is

Example :- • ones(2)

output :- $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

• ones(2,3)

output :- $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Command to multiply two matrices :-

$C = a * b$ where a & b are two matrices of size $m \times n$ & $n \times p$ respectively. then, we get a matrix of size $m \times p$

Example :- a = ones(2)
b = ones(2)
c = a * b

output :- $\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$

Scalar Operator :- It is used to multiply indexwise.

Example :- a = ones(2)
b = ones(2)
c = a .* b

output :- $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

Command to square a matrix :-

Let a be matrix of size $m \times n$

(1) a.*a or a.^2 → used to square indexwise elements

example :- let a = [1, 2; 3, 4]
a.*a

output :- $\begin{bmatrix} 1 & 4 \\ 9 & 16 \end{bmatrix}$

command to read an image :- i1 = imread('path of image . format');

Command to show an image :- imshow(i1);

ASSIGNMENT – 2

Intro with hold on, hold off and subplot.

hold → Retain current plot when adding new plots.

hold on – hold on retains plots in the current axes so that new plots added to the axes do not delete existing plots.

hold off – hold off sets the hold state to off so that new plots added to the axes clear existing plots and reset all axes properties.

```
X = linespace(-pi, pi);
Y1 = sin(X);
plot (X, Y1)
hold on
Y2 = cos(X);
plot = (X, Y2)
hold off
```

Subplot function :-

Subplot (m, n, p) divided the current figure into an m by n grid and creates axes in the position specified by p.

```
i1 = imread ('cameraman.tif');
Subplot (2, 2, 1)
imshow (i1);
Subplot (2, 2, 2)
imshow (i1);
```

ASSIGNMENT -3

Read the image and perform the TCC and FCC.

TCC → True Colour Composite

Redband – Red, Greenband – Green, Blueband – blue

FCC → False colour Composite

Any other combination of colours

Cat command – It is used to concatenate two or more commands.

```
i1 = imread ('C:\users\C1235\Pictures\Saved

b1 = i1(:, :, 1);
b2 = i1(:, :, 2);
b3 = i1(:, :, 3);
Subplot (2,2,1);
imshow (i1);
Subplot (2,2,2);
imshow (b1);
Subplot (2,2,3);
imshow (b2);
Subplot (2,2,4);
imshow (b3);
fcc = cat (3, b1, b2, b3);
imshow (fcc);
fcc = cat (3, b3, b1, b2);
imshow (fcc);
```

ASSIGNMENT -04

Implement the checkerboard effect

```
i1 = zeros(256, 256);
for i = 1:256
    for j = 1:256
        if (i==j)
            i1(i,j) = 0;
        else if (mod(j,2)==0) && (mod(i,2)==0)
            i1(i,j) = 0;
        else if (mod(j,2)== 0 || (mod(i,2)==0)
            i1(i,j) = 1;
        end
    end end
end
i1 = cast(i1, "uint8");
imshow(i1);
```

ASSIGNMENT -05

Perform alternate pixel 0 and alternate row and column 0.

```
i1 = imread ('cameraman.tif');
for i = 1: 256
    for j = 1: 256
        if mod (i, 2) == 0
            i1 (i, j) = 0;
        end
        if mod (j, 2) == 0
            i1 (i, j) = 0;
        end
end  end
imshow (i1);
```

ASSIGNMENT -06

Flip the image with function and without function.

(1) With function -

```
i1 = imread ('cameraman.tif');
i2 = flip (i1, 1);      → # vertically
i3 = flip (i1, 2);      → # horizontally
Subplot (2,2,1); imshow (i1);
Subplot (2,2,2); imshow (i2);
Subplot (2,2,3); imshow (i3);
```

(2) Without function :-

vertically :-

```
i1 = imread ('cameraman.tif');
i2 = zeros (256, 256);
for i = 1:256
 i2 = (256 - i+1 : ) = i1 (i,:);
end
 i2 = cast (i2, 'uint8');
 Subplot (1, 2, 1);
 imshow (i1);
 Subplot (1,2,2);
 imshow (i2);
```

ASSIGNMENT – 07

Perform transformation functions :–

(1) Negative transformation

```
i1 = imread ('cameraman.tif');
[m,n] = size (i1);
i2 = zeros (m,n);
for i = 1:m
    for j = 1:n
        i2(i,j) = 255 - i1(i,j);
    end
end
i2 = cast (i2, 'uint8');
subplot (2,2,1);
imshow (i1);
subplot (2,2,2);
imshow (i2);
```

(2) Logarithmic transformation :–

```
i1 = imread ('cameraman.tif');
[m,n] = size (i1);
i2 = zeros (m,n);
i1 = cast (i1, 'double');
c = input ('enter the value of c');
for i = 1:m
    for j = 1:n
        i2(i,j) = c * log (1 + i1(i,j));
    end
end
i1 = cast (i1, 'uint8');
i2 = cast (i1, 'uint8');
subplot (1,2,1); imshow (i1);
subplot (1,2,2); imshow (i2);
```

~~(3) Perform ROI~~

ASSIGNMENT-08

Perform ROI / gray level thresholding

```
i1 = imread ('cameraman.tif');
i2 = i1;
for i = 1:256
    for j = 1:256
        if i2(i,j) < 10 || i2 (i,j) > 20
            i2 (i,j) = 255;
        end
    end
end
subplot (1,2,1);
imshow (i1);
subplot (1,2,2);
imshow (i2);
```

ASSIGNMENT – 09

Perform Bitplane slicing

```
i1 = imread ('cameraman.tif');
subplot (1, 9, 1);
imshow (i1);
i1 = bitget (i1, 1);
i2 = bitget (i1, 2);
i3 = bitget (i1, 3);
i4 = bitget (i1, 4);
i5 = bitget (i1, 5);
i6 = bitget (i1, 6);
i7 = bitget (i1, 7);
i8 = bitget (i1, 8);
subplot (1, 9, 2);
p1 = cast (p1, 'logical');
imshow (p1);
subplot (1, 9, 3);
p2 = cast (p2, 'logical');
imshow (p2);
subplot (1, 9, 4);
p3 = cast (p3, 'logical');
imshow (p3);
subplot (1, 9, 5);
p4 = cast (p4, 'logical');
imshow (p4);
subplot (1, 9, 6);
p5 = cast (p5, 'logical');
imshow (p5);
subplot (1, 9, 7);
p6 = cast (p6, 'logical');
imshow (p6);
subplot (1, 9, 8);
p7 = cast (p7, 'logical');
imshow (p7);
subplot (1, 9, 9);
p8 = cast (p8, 'logical');
imshow (p8);
```

Assignment - 10

Perform Linear filter

(i) Average filter:-

```
i1 = imread ('cameraman.tif');
i1 = double(i1);
img = i1;
S= input ('Enter the size of mask');
C= (S+1)/2;
f= ones (5);
[m,n] = size(i1);
for i= C: m-C+1
    for j= C:n-C+1
        sum = 0;
        for k= 1:5
            for l= 1:5
                sum = sum + i1 (i-C+k, j-C+l) * f(k,l);
            end
        end
        img(i,j) = sum/(S*S);
    end
end
img = cast (img, 'uint8');
i1= cast (i1, 'uint8');
subplot (1,2,1);
imshow (i1);
subplot (1,2,2);
imshow (img);
```

(ii) Weighted Average Filter :-

```
i1 = imread ('cameraman.tif');
i2 = double (i1);
i3 = i2;
f1 = ones (3);
[m,n] = size (i1);
[x,y] = size (f1);

for i=2 : m-1
    for j= 2:n-1
        for k=1:3
            t= i2(i-1,j-1)*f1(1,1) + i2(i-1,j)*f1(1,2) + i2(i-1,j+1)*f1(1,3)+
               i2(i,j-1)*f1(2,1)+ i2(i,j)*f1(2,2) +i2(i,j+1) *f1(1,3)
               + i2(i+1,j-1)*f1(3,1)+i2(i+1,j)*f1(3,2) +
                    i2(i+1,j+1)*f1(3,3);

        t=t/9;
        i3(i,j) =t;
    end
end end

i3 = cast (i3, 'uint8');
Subplot (1,2,1);
imshow (i1);
Subplot (1,2,2);
imshow (i3);
```

Assignment - 11

Non - linear filters:

Order static filter:

(i) max Filter:

```
is = imread ('cameraman.tif');
img = is;
s = input ('Enter size of mask');
c = (s+1)/2
[m,n] = size (is);
    for i = c: m-c+1
        for j = c: n-c+1
            img(i,j) = max (max (is (i-c+1: i-c+s, j-c+1: j-c+s)));
        end
    end
subplot (1, 2, 1);
imshow (is);
subplot (1,2, 2);
imshow (img);
```

(ii) min Filters:

```
is = imread ('cameraman.tif');
img = is;
s = input ('Enter size of mask');
c = (s+1)/2;
[m,n] = size (is);
for i = c: m-c+1
    for j = c: m-c+1
        img (i,j) = min (min (is (i+c+1; i-c+s, j-c+1: j-c+s)));
    end
end
subplot (1,2,1);
imshow (is);
subplot (1,2,2);
imshow (img);
```

Assignment - 12

Perform histogram and equalization.

```
is = imread('cameraman.tif');
for i=1:256
    his(i,1) = i-1;
end
// for second column
for K=1:256
    count=0;
    for i=1:256
        for j=1:256
            if is(i,j)==K-1
                count= count+1;
            end
        end
    his(K,2)= count;
    end
end

// for 3rd column
for i=1:256
    his(i,3) = his(i,2)/ans;
end

// for 4th column
his(1,4) = his(1,3)
for i=2:256
    his(i,4)= his(i,3)+ his(i-1,4);
end

// for "(L-1)*sigma" 5th column
for i=1:256
    his(i,5) = his(i,4)*255;
end

// for Roundoff 6th column
for i=1:256
    his(i,6) = round(his(i,5));
end
```

```
for i= 1:256
    hiseq(i,1)= i-1;
end
for i= 1:256
    sum = 0;
    for j= 1:256
        if his(j,6)== i-1;
        sum = sum + his(j,2);
        end
    end
    hiseq(i,2) = sum;
end
subplot(2,2,1);
bar(his(:,1), his(:,2));
subplot(2,2,2);
bar(hiseq(:,1), hiseq(:,2));
t1 = i1;
for K= 1:256
    for i= 1:256
        for j= 1:256
            if(t1(i,j)== K-1;
                t1(i,j)= his(K,6);
            end
        end
    end
end
subplot(2,2,3);
imshow(i1);
subplot(2,2,4);
imshow(t1);
```

Assignment – 13

Perform derivative filter

→ Laplacian filter :-

```
i1 = imread('cameraman.tif');
i2 = double(i1);
i3 = i2;
i4 = i2;
i5 = i2;
f1 = ones(3);
f2 = [1,2,1; 2,4,2; 1,2,1];
f3 = [0,-1,0; -1,4,-1; 0,-1,0];
[m,n] = size(i1);
[x,y] = size(f1);
[p,q] = size(f3);
t1 = 0;
t2 = 0;
for i = 2:m-1
    for j = 2:n-1
        t = (i2(i-1,j-1)*f1(1,1) + i2(i-1,j)*f1(1,2) + i2(i-1,j+1)*f1(1,3) +
             i2(i,j-1)*f1(2,1) + i2(i,j)*f1(2,2) + i2(i,j+1)*f1(2,3) +
             i2(i+1,j-1)*f1(3,1) + i2(i+1,j)*f1(3,2) + i2(i+1,j+1)*f1(3,3));
        t1 = (i2(i-1,j-1)*f2(1,1) + i2(i-1,j)*f2(1,2) + i2(i-1,j+1)*f2(1,3) +
              i2(i,j-1)*f2(2,1) + i2(i,j)*f2(2,2) + i2(i,j+1)*f2(2,3) +
              i2(i+1,j-1)*f2(3,1) + i2(i+1,j)*f2(3,2) + i2(i+1,j+1)
              *f2(3,3));
        t2 = (i2(i-1,j-1)*f3(1,1) + i2(i-1,j)*f3(1,2) + i2(i-1,j+1)*f3(1,3) +
              i2(i,j-1)*f3(2,1) + i2(i,j)*f3(2,2) + i2(i,j+1)*f3(2,3) +
              i2(i+1,j-1)*f3(3,1) + i2(i+1,j)*f3(3,2) + i2(i+1,j+1)*f3(3,3));
```

```
        t = t(9);
        i3(i,j) = t;
        t1 = t1/16;
        i4(i,j) = t1;
        i5(i,j) = t2;
    end
end
i3 = cast(i3, 'uint8');
i4 = cast(i4, 'uint8');
i5 = cast(i5, 'uint8');
subplot(1,4,1);
imshow(i3);
subplot(1,4,2);
imshow(i4);
subplot(1,4,3);
imshow(i5);
```

Assignment - 14

Perform morphological operation using function on an image.

```
i1 = imread ('cameraman.tif');
i2 = im2bw (i1);
i3 = strel ('diamond', 2);
i4 = imerode (i2, i3);
i5 = imdilate (i2, i3);
i6 = imopen (i2, i3);
i7 = imclose (i2, i3);
Subplot (2,3,1);
imshow (i2);
Subplot (2,3,2);
imshow (i4);
Subplot (2,3,3);
imshow (i5);
Subplot (2,3,4);
imshow (i6);
Subplot (2,3,5);
imshow (i7);
Subplot (2,3,6);
imshow (i1);
```

Assignment - 15

Perform morphological operations using function on a matrix.

```
i1 = [0,0,0,0,0,0,0,0;
      0,0,0,0,0,0,0,0;
      0,1,0,0,0,0,1,0;
      0,1,1,1,1,1,1,0;
      0,1,1,1,1,1,1,0;
      0,1,0,0,0,0,1,0;
      0,0,0,0,0,0,0,0;
      0,0,0,0,0,0,0,0];

se = [1,1,1; 1,1,1; 1,1,1];

se1 = [0,1,1; 1,0,1; 1,1,0];

i2 = imdilate (i1, se);
i3 = imerode (i1, se);
i4 = imdilate (i3, se);
i5 = imerode (i2, se);
i6 = bwhitmiss (i1, se, se1);
i7 = i1 - i3;

subplot (3,3,1);
imshow (i1);
subplot (3,3,2);
imshow (se);
subplot (3,3,3);
imshow (se1);
subplot (3,3,4);
imshow (i2);
subplot (3,3,5);
imshow (i3);
subplot (3,3,6);
imshow (i4);
subplot (3,3,7);
imshow (i5);
subplot (3,3,8);
imshow (i6);
subplot (3,3,9);
imshow (i7);
```

Assignment - 15

Perform line segmentation.

```
im = imread ('cameraman.tif');
f1 = [-1,-1,-1; 2,2,2; -1,-1,-1];
f2 = [-1,-1,2; -1,2,-1; 2,-1,-1];
f3 = [-1,2,-1; -1,2,-1; -1,2,-1];
f4 = [2,-1,-1; -1,2,-1; -1,-1,2];
i1 = im2bw(im);
for i=2:255
    for j=2:255
        sum=0; sum1=0; sum2=0; sum3=0;
        for k=1:3
            for l=1:3
                sum = sum + i1(i-2+k, j-2+l)*f1(k,l);
                sum1 = sum1 + i1(i-2+k, j-2+l)*f2(k,l);
                sum2 = sum2 + i1(i-2+k, j-2+l)*f3(k,l);
                sum3 = sum3 + i1(i-2+k, j-2+l)*f4(k,l);
            end
        end
        sum = abs(sum);
        sum1 = abs(sum1);
        sum2 = abs(sum2);
        sum3 = abs(sum3);
        res = max([sum, sum1, sum2, sum3]);
        i2(i,j) = res;
    end
end
subplot(1,2,1);
imshow(im);
subplot(1,2,2);
imshow(i2);
```

Assignment - 17

Perform the edge detection.

```
i1 = imread ('cameraman.tif');
f1 = [-1,-2,-1; 0,0,0; 1, 2,1];
f2 = [-1, 0,-1; -2, 0, 2; -1, 0, 1];
a = input ('enter the thresshold');
for i = 2:255
   for j = 2:255
       sum1 = 0;
       sum2 = 0;
       for k = 1:3
        for l = 1:3
           sum1 = sum1 + i1(i-2+k, j-2+l) * f1(k,l);
           sum2 = sum2 + i1(i-2+k, j-2+l) * f2(k,l);
        end
      end
      sum1 = abs(sum1);
      sum2= abs (sum2);
      i2(i,j) = sum1 + sum2;
  end
end
 for i = 1:255
    for j = 1:255
      if (i2(i,j) < a)
         i2(i,j) = 0;
      else
         i2(i,j) = 255;
    end
  end
end
  subplot (1,2,1);
  imshow (i1);
  subplot (1,2,2);
  imshow (i2);
```