DIP LAB FILE

Name Mradul Varshney

Section B

University Roll No 2015 00 414

Year. III rd Year (2022 - 2023)

DIP Lab File

SUBMITTED TO:

MR. ANKUSH AGRAWAL.

Signature

INDEX

04/11/22

(2015004141)

Experiment 1 → Introduction to Matlab

Matlab is a software package.

Commands to create 1-D Array

```
a = [1,2,3,4,5]
b = [1 : 5]
c = [1 : 2 : 10]
```

Output

```
a =  1  2  3  4  5
b =  1  2  3  4  5
c =  1  3  5  7  9
```

Commands to create 1-D Array Stepwise.

d = start : step : end.

Example d = 1 : 0.5 : 5

Output ⇒ 1 1.5 2 2.5 3 3.5 4 4.5 5

x = 5 : -1 : 1

Output ⇒ 5 4 3 2 1

Command to create 2D Array

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

Output

```
a =  1  2  3
     4  5  6
```

2) **Magic Command** – Magic command is used to create a matrix of axa with random numbers.

Ex – Magic(3)

$$\begin{bmatrix} 1 & 8 & 3 \\ 4 & 6 & 9 \\ 2 & 5 & 7 \end{bmatrix}$$

Zeros command

(2015004141)

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

Example zeros(2)

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

zeros(2,3)

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ones command

ones command is used to create a matrix with 1's

Example ones(2)

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

ones(3,2)

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Command to multiply two matrices

c = a * b where a and b are two matrices of size mxn and n xp respectively

example a = ones(2) b = ones(2)

c = a * b

Output

$$\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

Scalar operation

It is used to multiply indexwise

Example a = ones(2) b = ones(2)

c = a .* b

Output =

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

(2015004141)

Command to square a matrix

① a*a or a^2 → used to square indexwise element

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 file.format')

Command to show an image

imshow(i1);

clc → command to clear screen

Clear space all → clear the workspace

Clear space c → clear the variable c from workspace.

Experiment -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 = linspace (-pi, pi);
y1 = sin (x)
plot (x, y1);
hold on
y2 = cos(x);
plot (x, y2)
hold off
```

Subplot function

Subplot (m, n, p) divides the current figure into an m by n grid and create axes in the position specified by p

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

Experiment - 3

Read the image and perform the TCC and FCC

TCC True color composite

Redband - Red, Greenband - Green, Blueband - Blue.

FCC False colour composite

Any other combinations of color.

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

```
i1 = imread('C:\Users\CL235\Pictures\glau.jpg');
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);
tcc = cat(3, b1, b2, b3);
imshow(tcc);
fcc = cat(3, b3, b1, b2);
imshow(fcc);
```

Experiment – 4

(2015004114)

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

i1 = cast(i1,'uint8');
imshow(i1);
```

Image size = ?

Experiment - 5 (20500414)

Alternate row and coloum 0 & 1

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

Experiment – 6 (20150041 4)

Flip the image with function and without function.

With function

```
i1 = imread('cameraman.tif');
i2 = flip(i1,1);
i3 = flip(i1,2);
subplot(1,3,1);
imshow(i1);
subplot(1,3,2);
imshow(i2);
subplot(1,3,3);
imshow(i3);
```

Without function

```
i1 = imread('cameraman.tif');
i2 = zeros(256);
for i=1:256
    i2(257-i,:) = i1(i,:);
end
i2 = cast(i2,'uint8');
subplot(1,2,1);
imshow(i1);
subplot(1,2,2);
imshow(i2);
```

Experiment-7 (201500414)

Perform ROI / gray level thresholding

```
i1 = imread ('cameran.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);
```

Experiment - 8 (20150041414)

Perform bit plane slicing

```
i1 = imread('cameraman.tif');
subplot(1,9,1);
imshow(i1);
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(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);
```

(20150041 4)

```
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);
```

Experiment - 9 (2015004114)

Perform transformation function

```
i1 = imread('cameraman.tif');
i2 = i1;
subplot(1,4,1);
imshow(i1);
for i = 1:256
    for j=1:256
        i2(i,j) = 255 - i2(i,j);
    end
end
subplot(1,4,2);
imshow(i2);
i1 = cast(i1,'double');
i2 = i1;
for i=1:256
    for j=1:256
        i2(i,j) = 100 * log(1+i2(i+j));
    end
end
i2 = cast(i2,'uint8');
subplot(1,4,3);
imshow(i2);
i2 = i1;
for i=1:256
    for j=1:256
        i2(i,j) = 20* i2(i,j)^0.8;
    end
end
i2 = cast(i2,'uint8');
subplot(1,4,4);
imshow(i2);
```

Experiment - 10

Histogram and equalization.

```
i1 = imread ('cameraman.tif');
for i = 1 : 256
    for
    hist(i,1) = i-1;
end

for K = 1 : 256
    c = 0;
    for i = 1 : 256
        for j = 1 : 256
            if i1(i,j) == K-1
                c = c + 1;
            end
        end
    end
    hist (K,2) = c;
end

for i = 1 : 256
    hist (i,3) = hist (i,2) / 65536;
end
hist (1,4) = hist (1,3);
for i = 2 : 256
    hist (i,4) = hist (i,3) + hist (i-1,4);
end
for i = 1 : 256
    hist (i,5) = hist (i,4) * 255;
end
```

```
for i = 1 : 256
    hist (i,6) = round (hist (i,5))
end
imshow (hist);
```

Experiment-11

Filters linear, Non-linear & order statistical

```
i1 = imread('cameraman.tif');
subplot(1,5,1);
imshow(i1);
[m,n] = size(i1);
% Average filter
f = ones(3,3);
i2 = cast(i1,'double');
i3 = i2;
for i = 2:m-1
    for j = 2:n-1
        t = (i2(i-1,j-1) * f(1,1)) + (i2(i-1,j) * f(1,2)) +
        (i2(i-1,j+1) * f(1,3)) + (i2(i,j-1) * f(2,1)) +
        (i2(i,j) * f(2,2)) + (i2(i,j+1) * f(2,3)) +
        (i2(i+1,j-1) * f(3,1)) + (i2(i+1,j) * f(3,2)) +
        (i2(i+1,j+1) * f(3,3));
        t = t/9;
        i3(i,j) = t;
    end
end
i3 = cast(i3,'uint8');
subplot(1,5,2);
imshow(i3);
```

```
% Weighted Average
f = [1,2,1; 2,4,2; 1,2,1];
i2 = cast(i1, 'double');
i3 = i2;
for i = 2 : m-1
  for j = 2 : n-1
    t = (i2(i-1, j-1) * f(1,1)) + (i2(i-1), j) * f(1,2)) +
    (i2(i-1, j+1) * f(1,3)) + (i2(i, j-1) * f(2,1)) +
    (i2(i, j) * f(2,2)) + (i2(i, j+1) * f(2,3)) +
    (i2(i+1, j-1) * f(3,1)) + i2(i+1, j) * f(3,2) +
    (i2(i+1, j+1) * f(3,3));
    t = t/16;
    i3(i,j) = t;
  end
end

i3 = cast(i3, 'uint8');
subplot(1,5,3);
imshow(i3);

% Laplacian filter
f = [0,-1,0; -1,4,-1; 0,-1,0];
i2 = cast(i1, 'double');
i3 = i2;
for i = 2 : m-1
  for j = 2 : n-1
    t = (i2(i-1,j-1) * f(1,1)) + (i2(i-1,j) * f(1,2)) +
    (i2(i-1,j+1) * f(1,3)) + (i2(i,j-1) * f(2,1)) +
```

```
(i2(i,j+1) * f(2,2)) + (i2(i,j+1) * f(2,3)) + (i2(i+1,j-1)*
  f(3,1)) + (i2(i+1,j) * f(3,2)) + (i2(i+1,j+1)*f(3,3));
  i3(i,j) = t;
 end
end
i3 = cast(i3,'uint8');
subplot(1,5,4);
imshow(i3);
```

Order Statistics

```
% MIN filter
  i1 = imread('cameraman.tif');
  img = i1
  s = input('Enter size of mask');
  c = (s+1)/2
  [m,n] = size(i1);
  for i = c : m-c+1
      for j = c : m-c+1
          img(i,j) = min[min(i1(i+c+1;i-c+s, j-c+1:j-c+s)]);
      end
  end
  subplot(1,2,1);
  imshow(i1);
  subplot(1,2,2);
  imshow(img);
```

max filter

```
i1 = imread ('cameraman.tif');
img = i1;
s = input ('Enter size of mask');
c = (s+1)/2;
[m,n] = size (i1);
for i = c : m-c+1
    for j = c : n-c+1
        img (i,j) = max(max(i1( i-c+1 : i-c+s , j-c+1 : j-c+s )));
    end
end
subplot (1,2,1);
imshow (i1);
subplot (1,2,2);
imshow (img);
```

Experiment – 12

Perform morphological ~~filter~~ operations.

```
i1= imread ('cameraman.tif');
i1 = imbinarize (i1);
op = i1;
cl = i1;
x = i1;
y = i1;
w = ones (3);
for i = 2:255
    for j = 2:255
        % Dilation
        x(i,j)= max([i1(i-1,j-1)* w(1), i1(i-1,j)* w(2),
        i1(i-1,j+1)* w(3), i1(i,j-1)* w(4), i1(i,j)* w(5),
        i1(i,j+1)* w(6), i1(i+1,j-1) * w(7), i1(i+1,j)*
        w(8), i1(i+1,j+1) * w(9)]);
    % Erosion
    y(i,j) = min([i1(i-1,j-1)* w(1), i1(i-1,j)*w(2),
    i1(i-1,j+1)* w(3), i1(i,j-1)*w(4), i1(i,j)*w(5),
    i1(i,j+1)* w(6), i1(i+1,j-1) * w(7), i1(i+1,j)* w(8),
    i1(i+1,j+1) * w(9)]);
% opening output
op(i,j) = max([y(i-1,j-1)* w(1), y(i-1,j)*w(2),
y(i-1,j+1)* w(3), y(i,j-1) * w(4), y(i,j) * w(5),
y(i,j+1)* w(6), y(i+1,j-1)* w(7), y(i+1,j) * w(8),
```

```
        y(i+1,j+1) * w(9);
    % closing output
    cl(i,j) = min([x(i-1,j-1) * w(1), x(i-1,j) * w(2),
    x(i-1,j+1) * w(3), x(i,j-1) * w(4), x(i,j) * w(5),
    x(i+1,j+1) * (6), x(i+1,j) * w(7), x(i+1,j) * w(8),
    x(i+1,j+1) * w(9)]);
  end
end
a = i1 - y;        % internal boundary
b = x - i1;        % external boundary
subplot(1,7,1);
imshow(i1);        % actual image
subplot(1,7,2);
imshow(x);         % Dilation
subplot(1,7,3);
imshow(y);         % Erosion.
subplot(1,7,4);
imshow(a);         % internal boundary
subplot(1,7,5);
imshow(b);         % external boundary
subplot(1,7,6);
imshow(op);        % opening output.
subplot(1,7,7)
imshow(cl);        % closing output.
```

Experiment 13

Perform Segmentation

```
img1 = imread ('cameraman.tif');
[m,n] = size (img1);
img 3 = zeros(m,n);
for i = 1 : m
    for j = 1:n
        if ( img1(i,j)>=150)
            img3 (i,j) = 1;
        else
            img3(i,j)= 0;

        end
    end
end

subplot (1,2,1);
imshow (img1);
subplot (1,2,2);
imshow (img 3);
```

Experiment - 14

Perform line segmentation

```
img= 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(img);
 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);
          rg = max([sum, sum1, sum2, sum3]);
      end
 end
 subplot(1,2,1);
 imshow(img);
 subplot(1,2,2);
 imshow(i2);
```

Experiment - 15

Perform 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 threshold');
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);
```