

# Experiment - 1

Objective :- Introduction with MATLAB Commands.

- clc → clears command window
- clear all → clear the work space
- date → displays current date
- linspace → creates regularly spaced vector
- max → returns largest element
- min → returns smallest element
- reshape → changes size
- size → computes array size
- eye → creates an identity matrix
- ones → creates an array of ones
- zeros → creates an array of zeroes
- plot → generates my plot
- subplot → creates plots in subwindows
- hold → freezes current plot

Ques-1 - write a program to represent in different ways

$$\rightarrow a = [1, 2, 3, 4, 5]$$

$$b = [1:5]$$

$$c = [1:2:5]$$

$$d = [1; 2; 3]$$

Ques-2 - write a program to represent array in 3 dimensions.

$$\rightarrow a = [1:3; 4:6; 7:9]$$

Ques-3 - write a program to add two matrices (dimension of matrix need to be same).

→  $a = [1:3; 4:6; 7:9]$   
     $b = [7:9; 4:6; 1:3]$   
     $c = a + b$

(Ques-4) Write a program to show multiplication of matrix?

→  $a = [1:3; 4:6; 7:9]$   
     $b = [1:3; 4:6; 7:9]$   
     $c = a * b$



## Experiment - 2

Objective: Use of linspace, plot, subplot, holdon & holdoff

Linspace, plot, holdon and holdoff :-

$$x = 0 : \pi/4 : 2 * \pi$$

$$y = \sin(x)$$

plot (x, y)

linspace (x, y)

linspace (0, 2 \* pi, 200)

plot (x, y)

x = linspace (0, 2 \* pi, 200)

y = sin(x)

plot (x, y);

z = cos(x);

plot (x, z);

hold on; plot (x, y);

plot (x, z);

hold off;

Subplotting :-

i1 = imread ("cameraman.tif")

i2 = imread ("pout.tif")

subplot (1, 2, 1);

imshow (i1);

subplot (1, 2, 2);

imshow (i2);

## Experiment - 3

Objective : Read the image and display it and study about attributes.

```
i1 = imread ("cameraman.tif");  
imshow(i1);
```

## Experiment - 4

Objective: Create a on-diagonal and off-Diagonal.

```
ii = imread ("cameraman.tif");
for i = 1:256
    ii(i,i) = 0;
end
for i = 1:256
    ii(i, 256-i+1) = 0;
end
imshow(ii);
```

# Experiment - 5

Objective: Create the horizontal and vertical strip

Horizontal strip:

```
ii = zeroes(200);
i = 21;
j = 0;
k = 1;
while i < 200
    while j < 10
        while k <= 200
            ii(i+j, k) = 255;
            k = k + 1
        end
        j = j + 1;
        k = 1;
    end
    i = i + 30;
    j = 0;
end
imshow(ii);
```

Vertical strip:

```
ii = zeroes(200);
i = 21;
j = 0;
k = 1;
while i < 200
```

```
while j < 10
    while K <= 200
        if (K, i+j) = 255;
            K = K + 1;
        end
        j = j + 1;
        K = 1;
    end
    i = i + 30;
    j = 0;
end
imshow(i1);
```

# Experiment - 6

Objective: Read the image and perform alternate pixel to zero.

```
i1 = imread ("cameraman.tif");
```

```
for r = 1:256
```

```
    if rem(r,2) == 0
```

```
        for c = 1:2:256
```

```
            i1(r,c) = 0;
```

```
        end
```

```
    else
```

```
        for c = 2:2:256
```

```
            i1(r,c) = 0;
```

```
        end
```

```
    end
```

```
end
```

```
imshow(i1);
```

# Experiment - 7

Objective: Read the image and perform alternate rows and columns to zero

$i_1 = \text{imread} ("cameraman.tif");$

for  $r = 1:256$

    for  $c = 2:2:256$

$i_1(r,c) = 0;$

    end

end

for  $r = 1:256$

    for  $c = 2:2:256$

$i_1(c,r) = 0;$

    end

end

$\text{imshow}(i_1);$

## Experiment - 8

Objective: Create a frame on the image.

$i_1 = \text{imread}("cameraman.tif");$

for  $c = 10 : 195$

    for  $j = 10 : 195$ .

        if ( $c >= 10 \& c <= 15 \& j >= 10 \& j <= 15$ )

$i_1(c, j) = 0;$

    end

        if ( $c >= 190 \& c <= 195 \& j >= 190 \& j <= 195$ )

$i_1(c, j) = 0;$

    end

end

imshow(i1).

# Experiment - 9

Objective: Read the coloured image and check attributes

```
i1 = imread("cameraman.tif");
```

```
r = i1(:,:,1);
```

```
g = i1(:,:,2);
```

```
b = i1(:,:,3);
```

```
imshow(i1);
```

# Experiment - 10

Objective: Perform TCC & FCC

i = imread ("images.jpg");

r = i(:,:,1);

g = i(:,:,2);

b = i(:,:,3);

RGB = cat (3,r,b,g);

GBR = cat (3,g,b,r);

GRB = cat (3,g,r,b);

BRG = cat (3,b,r,g);

BGR = cat (3,b,g,r);

subplot (2,3,1);

imshow(i);

subplot (2,3,2);

imshow(RGB);

subplot (2,3,3);

imshow(GBR);

subplot (2,3,4);

imshow(GRB);

subplot (2,3,5);

imshow(BRG);

subplot (2,3,6);

imshow(BGR);

# Experiment - 11

Objective: Generate the primary and secondary colors.

i<sub>1</sub> = cat(3, 0, 0, 0);

i<sub>2</sub> = cat(3, 0, 0, 1);

i<sub>3</sub> = cat(3, 0, 1, 0);

i<sub>4</sub> = cat(3, 0, 1, 1);

i<sub>5</sub> = cat(3, 1, 0, 0);

i<sub>6</sub> = cat(3, 1, 0, 1);

i<sub>7</sub> = cat(3, 1, 1, 0);

i<sub>8</sub> = cat(3, 1, 1, 1);

subplot(2, 4, 1);

imshow(i<sub>1</sub>);

Subplot(2, 4, 2);

imshow(i<sub>2</sub>);

subplot(2, 4, 3);

imshow(i<sub>3</sub>);

subplot(2, 4, 4);

imshow(i<sub>4</sub>);

Subplot(2, 4, 5);

imshow(i<sub>5</sub>);

subplot(2, 4, 6);

imshow(i<sub>6</sub>);

Subplot(2, 4, 7);

imshow(i<sub>7</sub>);

Subplot(2, 4, 8);

imshow(i<sub>8</sub>);

# Experiment -12

Objective: Implementation of basic intensity Transformation function (linear and non-linear).

```
i1 = imread('cameraman.tif');
subplot(1,2,1);
imshow(i1);
i2 = i1;
for r = 1:256
    for c = 1:256
        i2(r,c) = 255 - i1(r,c);
    end
end
subplot(1,2,2);
imshow(i2);
```

# Experiment - 13

Objective - Histogram of Image.

```
I = imread ("cameraman.tif");
```

```
I2 = zeroes (256, 2);
```

```
for i = 1:256
```

```
    I2(i,1) = i-1;
```

```
end
```

```
a = 1;
```

```
for i = 0:255
```

```
    C = 0;
```

```
    for j = 1:256
```

```
        for K = 1:256
```

```
            if (I(j,k) == i)
```

```
                C = C + 1;
```

```
            end
```

```
        end
```

```
    I2(a,2) = C;
```

```
    a = a + 1;
```

```
end
```

```
bar (I2(:,1), I2(:,2))
```

# Experiment - 14

Objective: Histogram Equalization

```
I = imread('cameraman.tif');
I2 = zeroes(256, 6);
for i = 1:256
    I2(i,1) = i-1;
end
a = 1;
for i = 0:255
    c = 0;
    for j = 1:256
        for k = 1:256
            if(I(j,k) == i)
                c = c + 1;
            end
        end
    end
    I2(a,2) = c;
    a = a + 1;
end

for i = 1:256
    I2(i,3) = I2(i,2)/65536;
end

I2(1,4) = I2(1,3);
for i = 2:256
    I2(i,4) = I2(i-1,4) + I2(i,3);
end
```

```

for i=1:256
    I2(i,5)=I2(i,4)*255;
end
for i=1:256
    I2(i,6)=rand(I2(i,5));
end

I3=zeros(256,2);
for i=1:256
    I3(i,1)=i-1;
end

for i=1:256
    main=0;
    for j=1:256
        if(I3(i,1)==I2(j,6))
            main=I2(j,2)+main;
        end
    end
    I3(i,2)=main;
end

subplot(2,1,1)
bar(I2(:,1),I2(:,2))
subplot(2,1,2)
bar(I3(:,1),I3(:,2))

j=histeq(I);
imshow(j);

for i=1:256
    for j=1:256
        I(i,j)=I2(i,j),6);
    end
end

imshow(I)

```

# Experiment - 15

Objective: Power Log Transformation

```
I = imread('cameraman.tif');  
I1 = cast(I,'double');  
c = 1;
```

```
subplot(2,3,1)
```

```
S = c*(I1)^0.6;  
S1 = cast(S,'uint8');  
imshow(S1);
```

```
subplot(2,3,2)
```

```
S = c*(I1)^0.2;  
S1 = cast(S,'uint8');  
imshow(S1)
```

```
subplot(2,3,3)
```

```
S = c*(I1)^1;  
S1 = cast(S,'uint8');  
imshow(S1);
```

```
subplot(2,3,4)
```

```
S = c*(I1)^1;  
S1 = cast(S,'uint8');  
imshow(S1)
```

# Experiment - 16

Objective: Logarithmic transformation of an Image.

```
t1 = imread('cameraman.tif');
```

```
t2 = cast(t1, 'double');
```

```
s1 = 1 * log(1 + t2);
```

```
s2 = 5 * log(1 + t2);
```

```
s3 = 10 * log(1 + t2);
```

```
s4 = 15 * log(1 + t2);
```

```
s5 = 20 * log(1 + t2);
```

```
s6 = 25 * log(1 + t2);
```

```
s7 = 30 * log(1 + t2);
```

```
s8 = 35 * log(1 + t2);
```

```
s9 = 40 * log(1 + t2);
```

```
subplot(3, 3, 1)
```

```
s1 = cast(s1, 'uint8')
```

```
imshow(s1)
```

```
subplot(3, 3, 2)
```

```
s2 = cast(s2, 'uint8')
```

```
imshow(s2)
```

```
subplot(3, 3, 3)
```

```
s3 = cast(s3, 'uint8')
```

```
imshow(s3)
```

```
subplot(3, 3, 4)
```

```
s4 = cast(s4, 'uint8')
```

```
imshow(s4)
```

```
subplot(3, 3, 5)
S5 = cast(S5, 'uint8')
imshow(S5)

subplot(3, 3, 6)
S6 = cast(S6, 'uint8')
imshow(S6)

subplot(3, 3, 7)
S7 = cast(S7, 'uint8')
imshow(S7)

subplot(3, 3, 8)
S8 = cast(S8, 'uint8')
imshow(S8)

subplot(3, 3, 9)
S9 = cast(S9, 'uint8')
imshow(S9)
```

# EXPERIMENT - 17

Objective: Bit plane Slicing of an Image

```
I = imread("cameraman.tif");
bit1 = bitget(i,1);
I1 = cast(bit1,'logical');
bit2 = bitget(i,2);
I2 = cast(bit2,'logical');
bit3 = bitget(i,3);
I3 = cast(bit3,'logical');
bit4 = bitget(i,4);
I4 = cast(bit4,'logical');
bit5 = bitget(i,5);
I5 = cast(bits,'logical');
bit6 = bitget(i,6);
I6 = cast(bit6,'logical');
bit7 = bitget(i,7);
I7 = cast(bit7,'logical');
bit8 = bitget(i,8);
I8 = cast(bit8,'logical');
subplot(2,4,1)
imshow(I1)
subplot(2,4,2) → title("bit1")
imshow(I2) → title("bit2")
subplot(2,2,3)
imshow(I3) → title("bit3")
subplot(2,4,4)
imshow(Z4)
title ("bit4")
```

subplot(2,4,5)

imshow(I5)

title("bit5")

subplot(2,4,6)

imshow(I6)

title("bit6")

subplot(2,4,7)

imshow(I7)

title("bit7")

subplot(2,4,8)

imshow(I8)

title("bit8")

# Experiment - 18

Objective: Resizing an Image.

```
I = imread ("https://tse3.mm.bing.net/th?id=OIP.iSu2RCCdm78X6XNDMJSqHaEo&pid=Api&p=0&h=180");  
I1 = imresize(I,[64,NaN]);  
I2 = imresize(I,[32,NaN]);  
I3 = imresize(I,[128,NaN]);
```

```
figure(1)  
imshow(I1)
```

```
figure(2)  
imshow(I2)
```

```
figure(3)  
imshow(I3)
```

# Experiment - 19

Objective - Applying smoothing filter on an Image.

I = imread ("cameraman.tif");

J2 = imnoise(I, 'gaussian');

J3 = zeroes(258, 258);

I3(2:257, 2:257) = 12;

J3 = imnoise(I, 'salt & pepper');

J3 = double(J3);

J4 = zeroes(256, 256);

M = [1 1 1; 1 1 1; 1 1 1];

w = 1/9 \* (M);

for i = 2:255

    for j = 2:255

        sum = 0;

        for k = 1:3

            for l = 1:3

                sum = sum + J3(i-2+k, j-2+l) \* M  
                end

            end

        end

        J4(i, j) = sum / 9;

    end

J4 = cast(J4, 'uint8');

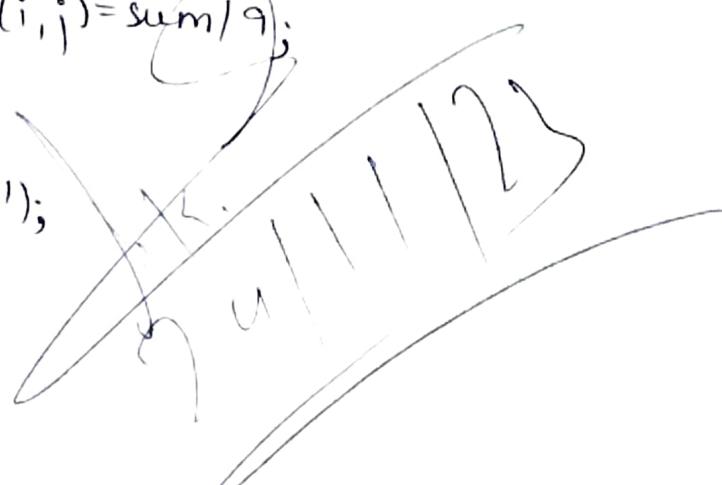
imshow(J4);

subplot(1, 2, 1);

imshow(J3);

subplot(1, 2, 2);

imshow(J4);



# Experiment - 19

Objective - Applying smoothing filter on an Image.

```
I = imread('cameraman.tif');
J2 = imnoise(I,'gaussian');
J3 = zeros(258,258);
I3(2:257,2:257)=12;
J3 = imnoise(I,'salt & pepper');
J3 = double(J3);
I4 = zeros(256,256);
M = [1 1 1; 1 1 1; 1 1 1];
w = 1/9*(M);
for i=2:255
    for j=2:255
        sum = 0;
        for k=1:3
            for l=1:3
                sum = sum + I3(i-2+k,j-2+l)*M(k,l);
            end
        end
        I4(i,j) = sum/9;
    end
end
I4 = cast(I4,'uint8');
imshow(I4);
subplot(1,2,1);
imshow(J3);
subplot(1,2,2);
imshow(I4);
```



# Experiment - 20

Objective: Dilation of an Image.

I1 = zeroes(10,10);

for i = 4:6

    for j = 4:7

        I1(i,j) = 1;

    end

end

se = ones(3,3);

I2 = zeroes(10,10);

for r = 2:size(I1,1)-1

    for c = 2:size(I1,2)-1

        if I1(r,c) == se(2,2)

            if isequal(I1(r-1:r+1,c-1:c+1),se)

                I2(r,c) = 1;

            end

        end

    end

imshow(I2);

# Experiment - 21

Objectiu: Erosion of an Image

```
I1 = zeroes(10,10);
for i=4:6
    for j=4:7
        I1(i,j) = 1;
    end
end
Se = ones(3,3);
I2 = zeroes(10,10);
for r=1:size(I1,1)
    for c=1:size(I1,2)
        if I1(r,c) == Se(2,2)
            for i=1:size(Se,1)
                for j=1:size(Se,2)
                    if (r+i-2 >= 1 & r+i-2 <= size(I2,1)) & &
                        (c+j-2 >= 1 & c+j-2 <= size(I2,2))
                        I2(r+i-2, c+j-2) = 1;
                end
            end
        end
    end
end
imshow(I2);
```

# Experiment - 22

Objective: Opening

$I_1 = \text{imread}("cameraman.tif");$

$I_1 = \text{im2bw}(I_1);$

$L = \text{length}(I_1);$

$Se = \text{ones}(3,3)$

$I_2 = \text{zeros}(1,1);$

for  $r = 1:L$

    for  $c = 1:L$

        if ( $r == 1 || c == 1 || r == L || c == L$ )

$I_2(r,c) = 0;$

        else

            if all( $(I_1(r-1:r+1), (c-1:c+1)) == Se$ )

$I_2(r,c) = 1;$

            end

        end

    end

end

$I_3 = \text{zeros}(1,L);$

for  $r = 1:L$

    for  $c = 1:L$

        if ( $I_2(r,c) == Se(2,2)$ )

            for  $i = 1:3$

                for  $j = 1:3$

$I_3(r+i-2, (c+j-2)) = 1;$

                end

            end

        end

    end

end

subplot(1,3,1);

imshow(I1);

subplot(1,3,2);

imshow(I2);

subplot(1,3,3);

imshow(I3);

} - Opening

imshow(22); } Fission

# Experiment - 33

Objective: Closing.

```
I1 = imread ("cameran.tif");
I1 = im2bw(I1);
L = length(I1);
se = ones(3,3);
I2 = zeros(L,L);

for r = 1:L
    for c = 1:L
        if (I1(r,c) == se(2,2))
            for i = 1:3
                for j = 1:3
                    I2(r+i-1, c+j-1) = 1;
            end
        end
    end
end

I3 = zeroes(L,L);
for r = 2:L-1
    for c = 2:L-1
        if all(I2(r-1:r+1,c-1:c+1) == se)
            I3(r,c) = 1;
        end
    end
end

subplot(1,3,1);
imshow(I1);
title("Original")
subplot(1,3,2)
imshow(I2)
title("Dilation")
subplot(1,3,3)
imshow(I3)
title("closing")
```

# Experiment - 24

Objective: External Boundary Extraction.

```

I1 = imread("cameraman.tif");
I1 = imbinarize(I1);
L = length(I1);
S1 = ones(3,3);
I2 = zeros(1,1);
for r=1:L
    for c=1:L
        if (I1(r,c) == S1(2,2))
            for i=1:3
                for j=1:3
                    I2(r+i-1,c+j-1)=1;
            end
        end
    end
end
I3 = zeroses(1,1);
for r=1:L
    for c=1:L
        if (I2(r,c)==0 & I1(r,c) == 1)
            I3(r,c)=0;
        else
            I3(r,c)=I2(r,c)-1*I1(r,c);
        end
    end
end
subplot(1,3,1) } - Original
imshow(I1);
subplot(1,3,2) } - Dilation
imshow(I2);
subplot(1,3,3) } - External Boundary.
imshow(I3);

```

# EXPERIMENT - 25

Objective: Internal Boundary Extraction

I1 = imread("cameraman.tif");

I1 = imbinarize(I1);

L = length(I1);

se = ones(3,3);

I2 = zeroes(L,L);

for r = 1:L

    for c = 1:L

        if (r == 1 || c == 1 || r == L || c == L)

            I2(r,c) = 0;

        else

            if all(I1(r-1:r+1, c-1:c+1) == se)

                I2(r,c) = 1;

            end

        end

    end

end

I3 = zeroes(1,L);

for r = 1:L

    for c = 1:L

        if (I1(r,c) == 0 & I2(r,c) == 1)

            I3(r,c) = 0;

        else

            I3(r,c) = I1(r,c) - I2(r,c);

        end

    end

end

subplot(1,3,1);

imshow(I1);

title("original"); } - Erosion

subplot(1,3,2)

imshow(I2)

subplot(1,3,3)

imshow(I3); } - Internal  
boundary.

# Experiment - 26

Objective: Line Detection of an Image

$I_2 = \text{imread('cameraman.tif')}$ ;  
 $L = \text{Length}(I_2)$ ;

$M1 = [-1 -1 -1; 2 2 2; -1 -1 -1]$ ;

$M2 = [-1 -1 2; -1 2 -1; 2 -1 -1]$ ;

$M3 = [-1 2 -1; -1 2 -1; -1 2 -1]$ ;

$M4 = [2 -1 -1; -1 2 -1; -1 -2 -1]$ ;

$I2\_p = \text{zeros}(L+2, L+2)$ ;

for  $r = 1:L$

for  $c = 1:L$

$I2\_p(r+1, c+1) = I2(r, c)$ ;

end end

$I2\_pd = \text{cast}(I2\_p, 'double')$ ;

$I2\_m1 = \text{zeros}(L, L)$ ;

$I2\_m2 = \text{zeros}(L, L)$ ;

$I2\_m3 = \text{zeros}(L, L)$ ;

$I2\_m4 = \text{zeros}(L, L)$ ;

for  $r = 1:L$

for  $c = 1:L$

$\text{sum1} = 0$ ;

$\text{sum2} = 0$ ;

$\text{sum3} = 0$ ;

$\text{sum4} = 0$ ;

for  $i = 1:3$

for  $j = 1:3$

$\text{sum1} = \text{sum1} + I2\_pd(r+i-1, c+j-1) * M1(i, j)$ ;

$\text{sum2} = \text{sum2} + I2\_pd(r+i-1, c+j-1) * (M2(i, j))$ ;

$\text{sum3} = \text{sum3} + I2\_pd(r+i-1, c+j-1) * M3(i, j)$ ;

$\text{sum4} = \text{sum4} + I2\_pd(r+i-1, c+j-1) * M4(i, j)$ ;

end end

$I2\_m1(r, c) = \text{sum1}$ ;

$I2\_m2(r, c) = \text{sum2}$ ;

$I2\_m3(r, c) = \text{sum3}$ ;

I2\_m4(r, c) = sum^4;

end

end

I2\_r1 = cast(I2\_m1, 'uint8');

I2\_r2 = cast(I2\_m2, 'uint8');

I2\_r3 = cast(I2\_m3, 'uint8');

I2\_r4 = cast(I2\_m4, 'uint8');

max\_filtered = max(cat(3, I2\_r1, I2\_r2, I2\_r3, I2\_r4), [ ], 3);

figure(7)

subplot(2, 3, 1);

imshow(I2);

subplot(2, 3, 2);

imshow(I2\_r1);

subplot(2, 3, 3);

imshow(I2\_r2);

subplot(2, 3, 4);

imshow(I2\_r3);

subplot(2, 3, 5);

imshow(I2\_r4);

subplot(2, 3, 6);

imshow(max\_filtered);