Experiment - 1

Name: Lalit Saini
Roll No. : 34

Object: To perform basic command in MATLAB.

Theory: MATLAB is an abbreviation for matric Laboratory, while other programming 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 command window.
Clear all - delete variable from workspace.
Quit - Stop MATLAB
who - List current variables.
function - purpose
char - convert to char array

Basic operations:

Input A=50 , B= 2

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

1-15 Attended.
20/05/20

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; 2 3 4; 1 2 5]
b = det(a)
```

output

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

b = -2
```

Experiment: 2

Object: To perform plot(), subplot(), linspace() functions.

Theory:

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

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

Linspace: It is similar to the colon operator, but gives 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, y1, x, y2)
```

Subplot →

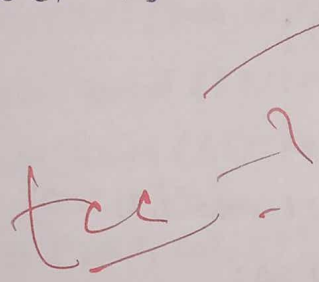
```
subplot(2,1,1);
x = linspace(0, 1.0);
y1 = sin(x);
plot(x, y1);
subplot(2,1,2);
y2 = sin(5*x);
plot(x, y2);
```

Roll No: 34

Experiment: 3

Object: To read an image and extract the bands and perform the TCC and FCC on it.

```
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);
```



Experiment : 4

Object: To make alternative pixel to 0, alternative column and rows to 0 in an image.

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(a);
```

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);
```

ROLL NO: 34

Experiment: 5

Object: Implement the checkerboard effect on an image.

a. 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);
```

b. Without using imresize() function

```
a= imread('D:\img.jpeg');
Scale = [10,10];
oldS = size(a);
newS = max[floor(Scale.*oldS(1:2)),1);
Col = min(round(((1:newS(2))-0.5)./Scale(2)+0.5),oldS(2));
Row = min(round(((1:newS(1))-0.5)./Scale(1)+0.5),oldS(1));
z = a(row, col, :);
Subplot(1,2,1);
imshow(a);
Subplot(1,2,2);
imshow(z);
```

Experiment - 9

Object: Flip the image in horizontal and vertical dirⁿ.

using flip():

```
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');
```

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
subplot(1,3,1);
imshow(a); title('original');
subplot(1,3,2);
imshow(V); title('vertical flip');
subplo(1,3,3);
imshow(H); title('Horizontal flip');
```

Experiment: 7

Roll No: 34

Object: To perform Region of Interest (ROI) in an image.

Implementation:

```
im= imread('cameramen.tif');
im = imresize(im, [256 256]);
imshow(im);
[col row] = ginput(4);
   c = col;
   r = row;
Binary mask = roipoly(im, c, r);
 figure,
imshow(Binarymask); title('selected ROI');
NONROI = Zeros(256, 256);
    ROI = Zeros(256, 256);
for i=1:256
   for j=1:256
      if(Binarymask(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('NON ROI');
```

Experiment-8

Object: To implement Bit plane Slicing.

Implementation:

```
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);
  imshow( logical (p6));
  Subplot (3,3,8);
   imshow (logical(p7));
  Subplot (3,3,9);
  imshow (logical (p8));
```

Experiment: 9

Roll No: 34

Object: To perform transformation on Image.

Implementation:

Negative transformation:

```
im = imread('cameramen.tif');
i1 = 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(i1); title('Before');
subplot(2,2,2);
imshow(y); title('after neg');
```

Log transformation:

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

power transformation:

```
i1 = imread('cameramen.tif');
im = i1; i2 = i1; i3 = i1; c = 12.5; u = 1;
for i=1:255
    for j=1:255
        y = cast(i1(i,j), 'Double');
        i1(i,j) = 12.5 * y.^(1.1);
        i2(i,j) = 14.6 * y.^(0.9);
        i3(i,j) = 13.4 * y.^(1.2);
    end
end
subplot(2,2,1); imshow(im); subplot(2,2,2); imshow(i1);
subplot(2,2,3); imshow(i2); subplot(2,2,4); imshow(i3);
```

Experiment: 10

Object: To perform and implement Histogram.

Implementation:

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

Experiment : 11

Object: 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
```

Experiment: 12

Objective: To apply average, min, max, median filter 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, 'unit8');
b = cast (b, 'unit8');
subplot (1,2,1) ;
imshow (a);
subplot (1,2,2);
imshow(b);
```

Roll No.:- 54

Experiment: 13

Objective: To perform Dilation on an image.

implementation: (without function)

```
a = imread ('cameraman.tif');
z = im2bw (a);
dil = zeros (258);
d = zeros (256);
[m n] = size (dil);
mask = ones (3);
dil (2:257, 2:257) = z;
c = z;
for i = c:m-c+1
    for j = c:n-c+1
        for k = 1:3
            for l = 1:3
                if mask (k,l) == dil (i-c+k, j-c+l);
                    d (i-1, j-1) = 1;
                end
            end
        end
    end
end
subplot (1,2,1);
imshow (z); title ('original');
subplot (1,2,2);
imshow (d); title ('dilation');
```

(with fuction):

```
a = imread ('cameraman.tif');
i = im2bw (a);
mask = ones (3);
dil = imdilate (i, mask);
subplot (1,2,1);
imshow (i); title ('original');
subplot (1,2,2);
imshow (dil); title ('dilation');
```

Experiment: 14

Objective: To perform Erosion on an image.

Implementation: (without function):

```
a = imread('cameraman.tif');
z = im2bw(a);
dt = zeros(258);
d = ones(258);
[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 1 == dt(i-c+k, j-c+l)
                        d(i-1, j-1) = 0;
                    end
                else
                    if 0 == dt(i-c+k, j-c+l)
                        d(i-1, j-1) = d(i-1, j-1);
                    end
                end
            end
        end
    end
end
subplot(1,2,1);
imshow(z); title('original');
subplot(1,2,2);
imshow(d); title('Erosion');
```

Experiment : 15

Objective: To perform opening & closing on an image.

implementation: (opening)

```
a = imread ('cameraman.tif');
i = im2bw(a);
ma = ones(3);
er = imerode(i, ma);
dt = imdilate(er, ma);
op = imopen(i, ma);
subplot(1,3,1);
imshow(i); title('original');
subplot(1,3,2);
imshow(dt); title('Opening without function');
subplot(1,3,3);
imshow(op); title(' with function opening');
```

closing implementation:

```
a = imread ('cameraman.tif');
i = im2bw(a);
ma = ones(3);
dt = imdilate(i, ma);
er = imerode(dt, ma);
closing = imclose(i, ma);
subplot (1,3,1);
imshow(i); title('original');
subplot(1,3,2);
imshow(er); title(' without function closing');
subplot(1,3,3);
imshow(closing); title('with function closing');
```