Practical - 1

Introduction to Matlab:

- Making a Matrix

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

- for finding min -> min(u);
- for finding max -> max(u);
- for finding determinant -> det(u);
- for finding inverse -> inv(u);
- making all zeros matrix

```
z=zeros(3);
```

- making all ones matrix

```
o=ones(3);
```

u * u -> scalar multiplication

u .* u -> value multiplication

Practical - 2

Plot the sin and cosine curve.

```
x=linspace(0, 2*pi, 50);
y=sin(x);
z=cos(x);
hold on
plot(x,y,'r');
plot(x,z,'g');
legend('sin');
legend('cos');
```

Experiment -3

Plotting the various curves (sine, cosine, square, exponential)

Code -

```
x = linspace(0, 2*pi, 500);
y = sin(x);
z = cos(x);
w = x.*x2
t = exp(x);
subplot(2,2,1);
plot(x,y);
legend('sin curve');
title('sin curve');
subplot(2,2,2);
plot(x,z)
legend('cos curve')
title('cos curve')
subplot(2,2,3);
plot(x,w);
legend('square curve');
title('square curve');
subplot(2,2,4);
plot(x,t);
legend('exp');
title('exponential curve')
```

Experiment - 4

To flip the image vertical and Horizontal.

```
img = imread('cameraman.tif');
[m,n] = size(img);
for i = 1:m
    for j = 1:n
        img1(i,j) = img(i,n-j+1);
    end
end
for i = 1:m
    for j = 1:n
        img2(i,j) = img(m-i+1,j);
    end
end

subplot(1,3,1);
imshow(img);
title('original image');
subplot(1,3,2);
imshow(img1);
title('Horizontal image');
subplot(1,3,3);
imshow(img2);
title('vertical flip image');
```

Practical - 5

Implement Gray level slicing with and without Background.

```
img=imread('bout.tif');
[m,n]=size(img);
for i=1:m
    for j=1:n
        if (img(i,j)>=100 && img(i,j)<=300)
            img1(i,j)=255;
            img2(i,j)=255;
        else
            img1(i,j)=0;
            img2(i,j)=img(i,j);
        end
    end
end
subplot(1,3,1);
imshow(img);
subplot(1,3,2);
imshow(img2);
title('without Background');
subplot(1,3,3);
imshow(img2);
title('with Background');
```

Practical - 6

Various Intensity Transformation (Negative, log and gamma)

```
img=imread('cameraman.tif');
[m,n]=size(img);
for i=1:m
    for j=1:n
        img1(i,j)=255-img(i,j);
    end
end
for
c=input('enter c:')
for i=1:m
    for j=1:n
        img2(i,j)=c*log(double(1+img(i,j)));
    end
end
for gamma=input('enter gamma:')
for i=1:m
    for j=1:n
        img3(i,j)=c*double(img(i,j)).^gamma;
    end
end
```

```
subplot(3,1,1);
imshow(img1);
title('Negative Transformation');
subplot(3,1,2);
imshow(img2);
title('log Transformation');
subplot(3,1,3);
imshow(img3);
title('Gamma Transformation');
```

Practical - 7

To implement bitplane slicing.

```
img=imread('cameraman.tif');
b1=mod(img,2);
b2=mod(bitsrl(img,1),2);
b3=mod(bitsrl(img,2),2);
b4=mod(bitsrl(img,3),2);
b5=mod(bitsrl(img,4),2);
b6=mod(bitsrl(img,5),2);
b7=mod(bitsrl(img,6),2);
b8=mod(bitsrl(img,7),2);

subplot(3,3,1);
imshow(img);
title('original');
subplot(3,3,2);
imshow(logical(b1));
title('plane 1');
subplot(3,3,3);
imshow(logical(b2));
title('plane 2');
subplot(3,3,4);
imshow(logical(b3));
title('plane 3');
```

```
subplot(3,3,5);
imshow(logical(b5));
title('plane 5');
subplot(3,3,7);
imshow(logical(b6));
title('plane 7');
subplot(3,3,9);
imshow(logical(b8));
title('plane 8');
```

Practical - 8

To subplot image histogram and equalise image histogram and also comput contrast stretching.

```
img=imread('boot.tif');
subplot(4,2,1);
imshow(img);
subplot(4,2,2);
imhist(img);
img_2=histeq(img);
subplot(4,2,3);
imshow(img_2);
subplot(4,2,4);
imhist(img_2);
img_3=imadjust(img);
subplot(4,2,5);
imshow(img_3);
subplot(4,2,6);
imhist(img_3);
[n,m]=size(img);
img_4=img;
for i=1:n
    for j=1:m
        img_4(i,j)=((img(i,j)-74)*250)/150;
    end
end
```

```
img_4 = uint8(img_4);
subplot(4,2,7);
imshow(img_4);
subplot(4,2,8);
imhist(4,2)
imhist(img_4);
```

Practical - 9

To implement Histogram Equalization.

```
img=imread('cameraman.tif');
[n,m]=size(img);
for i=1:256
    t(i,1)=i-1;
    t(i,2)=0;
end
for i=1:n
    for j=1:m
        hiu=img(i,j)+1;
        t(hiu,2)=t(hiu,2)+1;
    end
end
for i=1:256
    t(i,3)=t(i,2)/(256*256);
end
t(1,4)=t(1,3)
for i=2:256
    t(i,4)=t(i-1,4)+t(i,3);
end
for i=256
    t(i,5)=t(i,4)*255;
end
```

```
for i:256
    t(i,6)=round(t(i,5));
end
for i=1:256
    s(i,1)=i-1;
    s(i,2)=0;
end
for i=1:256
    bin=t(i,6)+1;
    s(bin,2)=s(bin,2)+t(i,2);
end
img
img_2=img
for i=1:n
    for j=1:m
        img2(i,j)=t(img(i,j)+1,6);
    end
end
img_3=histeq(img);

for i=1:n
    for j=1:m
        bin=img_3(i,j)+1;
        p(bin,2)=p(bin,2)+1;
    end
end
```

```
subplot(3,2,1);
imshow(img);
title('original');
subplot(3,2,2);
bar(t(:,1),t(:,2));
title('input hist');
subplot(3,2,3);
imshow(img_2);
title('output hist');
subplot(3,2,4);
bar(s(:,1),s(:,2));
title('output hist');
subplot(3,2,5);
imshow(img_3);
subplot(3,2,6);
bar(p(:,1),p(:,2));
```

Practical-10

To implement Average filter.

```
img=imread('bout.tif');
[n,m]=size(img);
img2=img;
img3=img;
s=3;
f=ones(s,s); f1=[1,2,1;2,4,2;1,2,1];
c=(s+1)/2;
for i=c:n-c+1
    for j=c:m-c+1
        sum=0;
        sum1=0;
        for k=1:s
            for l=1:s
                sum=sum+double(img(i-c+k,j-c+l)*f(k,l));
                sum1=sum1+double(img(i-c+k,j-c+l)*f1(k,l));
            end
        end
        img2(i,j)=sum/9;
    end
end
```

```
subplot(1,3,1);
imshow(img);
title('original image');
subplot(1,3,2);
imshow(img_2);
title('after applying average filter');
subplot(1,3,3);
imshow(img_3);
title('after apply weighted filter');
```

Experiment - 11

To Implement Min, Max and Median filters

```
img = imread('cameraman.tif');
[n, m] = size(img);
img_2 = img;
img_3 = img;
img_4 = img;
c = 2;
for i = c : n-c+1
    for j = c : m-c+1
        img_2(i,j) = min(min(img(i-1:i+1, j-1:j+1)));
        img_3(i,j) = max(max(img(i-1:i+1, j-1:j+1)));
        img_4(i,j) = median(median(img(i-1:i+1, j-1:j+1)));
    end
end
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(img_2);
title('Output min filtering');
```

```
subplot(2,2,3);
imshow(img_3);
title('After mean filtering');
subplot(2,2,4);
imshow(img_4);
title('After median filtering');
```

Experiment - 12

To implement Laplacian filters (both L-4 & L-8);

```
img = imread('cameraman.tif')
[n,m] = size(img);
img = double(img);
s=3;
f=[0,1,0;1,-4,1;0,1,0];
f1=[1,1,1;1,-8,1;1,1,1];
img_2 = img;
img_3 = img;
for i=2:n-(s-1)
  for j=2:m-(s-1)
    sum=0;
    sum1=0;
    for k=1:s
      for l=1:s
        sum=sum+img(i-(s-k),j-(s-l))*f(k,l);
        sum1=sum1+img(i-(s-k),j-(s-l))*f1(k,l);
      end
    end
  img_2(i,j)=sum;
  img_3(i,j)=sum1;
end
end
```

```
subplot(1,3,1);
imshow(img);
title('Original');
subplot(1,3,2);
imshow(img_2);
title('After L-4 filtering')
subplot(1,3,3);
imshow(img_3);
title('After L-8 filtering');
```

Experiment - 13

Implementing Morphological Operations

```
// with Downloaded Image
img = imread('C:\Users\Downloads\Image.jpg');
bin_img = im2bw(img);
se = strel('disk',5);
img_2 = imdilate(bin_img, se);
img_3 = imerode(bin_img, se);
img_4 = imdilate(img_3, se);
img_5 = imerode(img_2, se);
img_6 = imopen(bin_img, se);
img_7 = imclose(bin_img, se);
img_8 = img-bin_img - img3;
img_9 = img2 - bin_img;

subplot(3,3,1);
imshow(img);
title('Original');
subplot(3,3,2);
imshow(img_2);
title('dilated');
```

```
Sk
subplot(3,3,3);
imshow(img_3);
title('eroded');
subplot(3,3,4);
imshow(img_4);
title('opening');
subplot(3,3,5);
imshow(img_5);
title('closing');
subplot(3,3,6);
imshow(img_6);
subplot(3,3,7)
imshow(img_7);
title('in-built closing');
subplot(3,3,8);
imshow(img_8);
title('internal boundary')
subplot(3,3,9);
imshow(img_9);
title('external boundary');
```

// with created binary image

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

mat_str = [0,1,0; 1,1,1; 0,1,0];

dilate = imdilate(mat, mat_str);
erode = imerode(mat, mat_str);
in_bndry = mat - erode;
ex_bndry = dilate - mat;
close = imerode(dilate, mat_str);
open = imdilate(erose, mat_str);

subplot(2,4,1);
imshow(&mat);
title('Original')
subplot(2,4,2);
imshow(dilate);
title('After dilation');
subplot(2,4,3);
imshow(erode);
title('After erosion');
```

```
subplot(2,4,4);
imshow(in_bndry);
title('Internal Boundary');
subplot(2,4,5);
imshow(en_bndry);
title('External Boundary');
subplot(2,4,6);
imshow(close);
title('closing');
subplot(2,4,7);
imshow('Open');
title('Opening');
subplot(2,4,8);
imshow(
title(
```