SCILAB PRACTICALS

Digital Image Processing

Prerna (20201420)

Exam Roll no. - 20020570025

1. Write program to read and display digital image using MATLAB or SCILAB

a. Become familiar with SCILAB/MATLAB Basic commands
b. Read and display image in SCILAB/MATLAB
c. Resize given image
d. Convert given color image into gray-scale image
e. Convert given color/gray-scale image into black & white image
f. Draw image profile
g. Separate color image in three R G & B planes

#SOLUTIONS

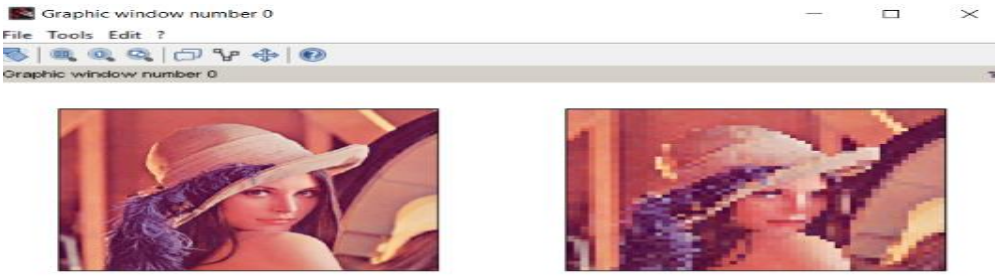
1(a)

```
imshow('2 * 3 - 4 + 8 / 3 \ 9 =', 2*3-4+8/3\9);

x = linspace(0,8,100);

plot(sin(x),'o-');plot(cos(x),'r.-');

xtitle('sin & cos waves');

legend('sin(x)','cos(x)', 3);

xgrid(0,1,7);
```

1(b) - read and show image

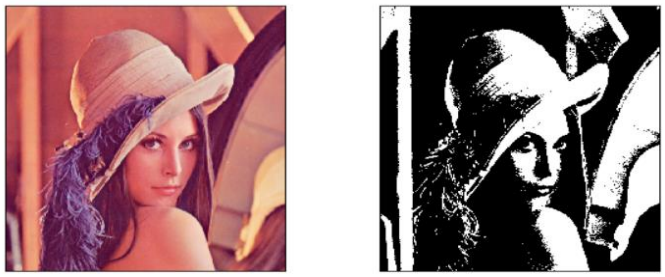
```
I = imread("lenna.jpg")
```

1(c) - resize the image
x = imresize(I,0.1)

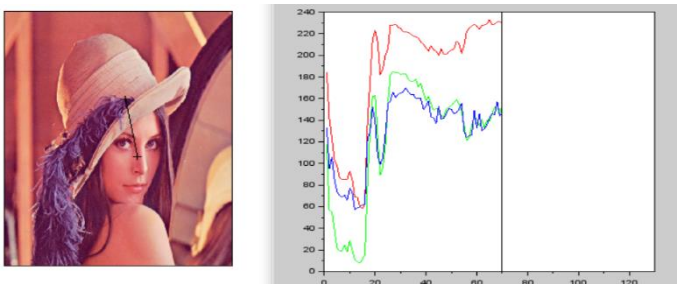


1(d)
a = rgb2gray(I)

1(e)
b = im2bw(I)



1(f)
i = fitsread(I)
improfile(I)



1(g)

```
I = imread('lenna.jpg');
r = size(I, 1);
c = size(I, 2);
R = zeros(r, c, 3);
G = zeros(r, c, 3);
B = zeros(r, c, 3);
R(:, :, 1) = I(:, :, 1);
G(:, :, 2) = I(:, :, 2);
B(:, :, 3) = I(:, :, 3);
figure, imshow(uint8(R));
figure, imshow(uint8(G));
figure, imshow(uint8(B));
```

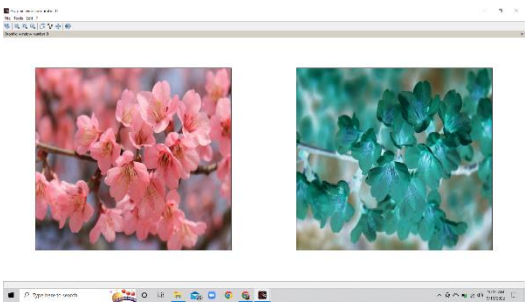
2. To write and execute image processing programs using point processing method

a. Obtain Negative image
b. Obtain Flip image
c. Thresholding
d. Contrast stretching

#SOLUTIONS

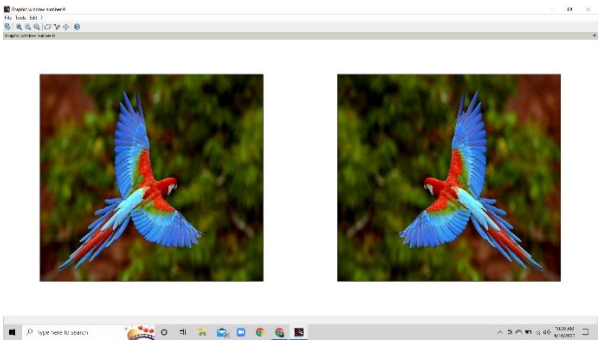
2(a)

```
i = imread('sakura.png')
imshow(i)
L = 2 ^ 16;
neg = (L - 1) - i;
subplot(1, 2,2)
imshow(neg);
```



2(b)

```
i = imread("sakura.png")
img = flipdim(i,2)
imshow(img)
```



3. To write and execute programs for image arithmetic operations

a. Addition of two images
b. Subtract one image from other image
c. Calculate mean value of image

3(a)

```
I = imread('camera.png')
J = imread('rice.jpg')

im1 = imresize(I,[256 256])
im2 = imresize(J,[256 256])
K = imadd(im1,im2)
subplot(1,3,1)
imshow(im1)
subplot(1,3,2)
imshow(im2)
subplot(1,3,3)
imshow(K)
```

3(b)

I = imread('circle.jpg')J = imread('square.jpg')

```
im1 = imresize(I,[256 256])
im2 = imresize(J,[256 256])
K = imadd(im1,im2)
subplot(1,2,1)
imshow(K)
x = imsubtract(K,im1)
subplot(1,2,2)
imshow(x)
```

3(c)

```
a = mean(I(:));
imshow(a)
```

>> output – 8.7

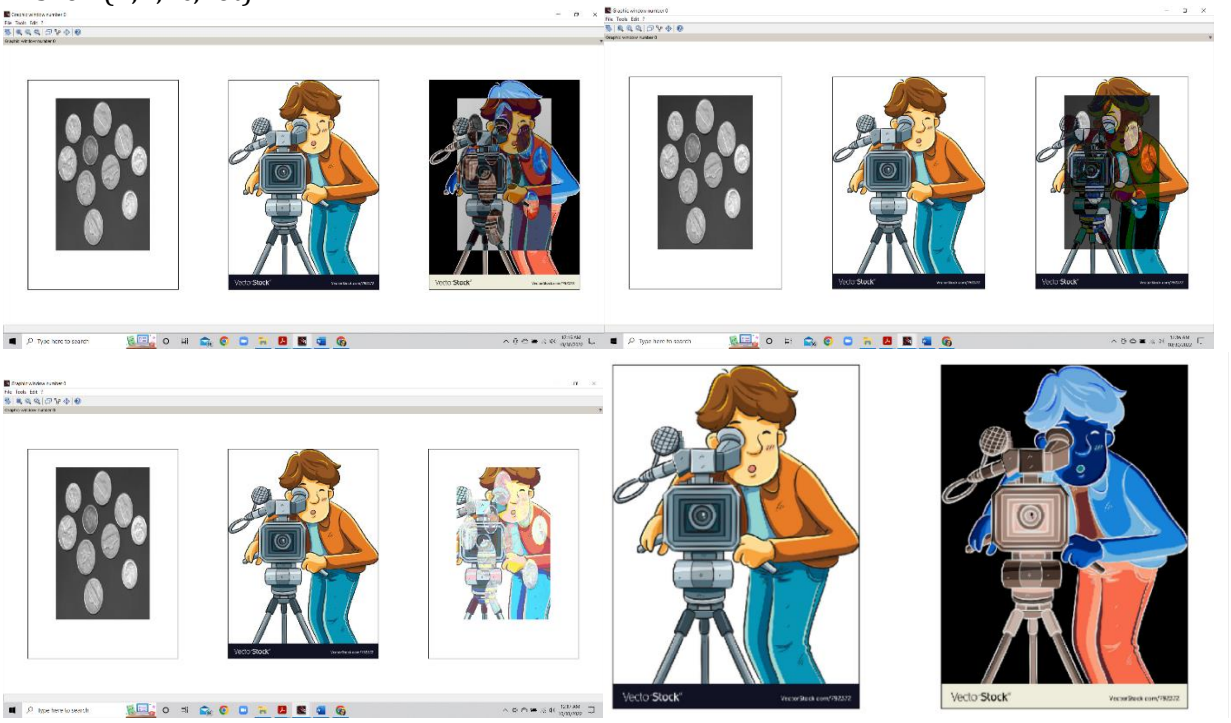
4. To write and execute programs for image logical operations

a. AND operation between two images
b. OR operation between two images
c. Calculate intersection of two images
d. NOT operation (Negative image)

#SOLUTION

```
I1 = imread("coins.png")
I2 = imread("cameraman.jpg")
im1 = imresize(I1,[576 576])
im2 = imresize(I2,[576 576])
subplot(1,3,1)
imshow(im1)
subplot(1,3,2)
imshow(im2)
K = bitand(im1,im2);//AND   --→(a)
Ac = bitor(im1,im2);//or   --→(b)
B = imabsdiff(im1,im2)//intersection   ---------→(c)
```

not = bitcmp(im2);*//not --------→(d)*
subplot(1,3,3)
imshow(B,K,Ac,not)



5. To write a program for histogram calculation and equalization using

a. Standard MATLAB function

```
I = rgb2gray(imread('parrot.jpeg'));
Iequal = imhistequal(I);
[qtd, level] = imhist(I);
[qtde, levele] = imhist(Iequal);
subplot(221);
imshow(I);
subplot(222);
plot2d3(level, qtd);
subplot(223);
imshow(Iequal);
subplot(224);
plot2d3(levele, qtde);
```

Q. 6) To write and execute program for geometric transformation of image :

a. Translation

b. Scaling

c. Rotation

d. Shrinking

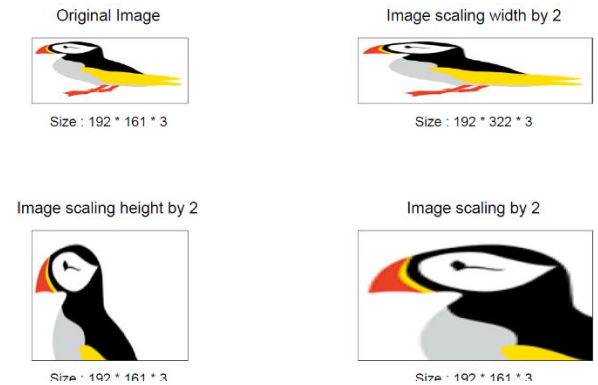
e. Zooming

6(a)

```
S1 = imread('Test_images/lena.jpeg');

mat = [  1 0 0;...

       0 1 0;...

      20 0 1];

S2 = imtransform(S1,mat,'affine');

mat(3, 1:2) = [0 -20];

S3 = imtransform(S1,mat,'affine');

mat(3, 1:2) = [-20 30];

S4 = imtransform(S1,mat,'affine');

subplot(2,2,1), title('Original Image'), imshow(S1);

subplot(2,2,2), title('Translation for x = 20'), imshow(S2);

subplot(2,2,3), title('Translation for y = -20'), imshow(S3);

subplot(2,2,4), title('Translation for (-20,30)'), imshow(S4);
```

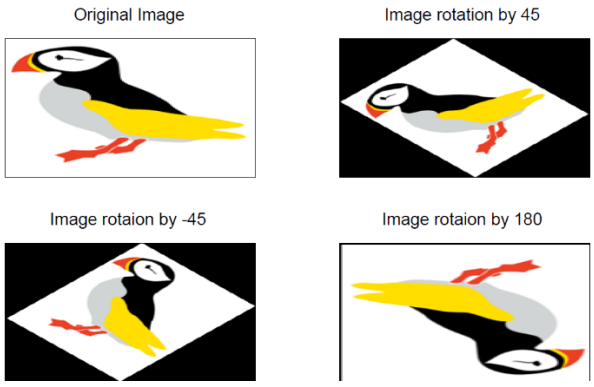
6(b)

```
s_img = imread(fullpath(getIPCVpath() + "/images/puffin.png"));
width = size(s_img, 'c');     // column pixels = width
height = size(s_img, 'r');   // row pixels = height
mat = [ 2 0;
      0 1;
      0 0];
mat([1,5]) = [1 2];
sc2 = imtransform(s_img, mat, 'affine', width*mat(1), height*mat(2));
mat([1,5]) = [2 2];
sc3 = imtransform(s_img, mat, 'affine', width*mat(1), height*mat(2));
function s = str(img)
   s = 'Size : ' + strcat(string(size(img)), ' * ');
endfunction;
subplot(3,3,1), title('Original Image'), xlabel(str(s_img)), imshow(s_img);
subplot(3,2,2), title('Image scaling width by 2'),xlabel(str(sc1)), imshow(sc1);
subplot(2,3,4), title('Image scaling height by 2'),xlabel(str(sc2)), imshow(sc2);
subplot(2,2,4), title('Image scaling by 2'),xlabel(str(sc3)), imshow(sc3);
```



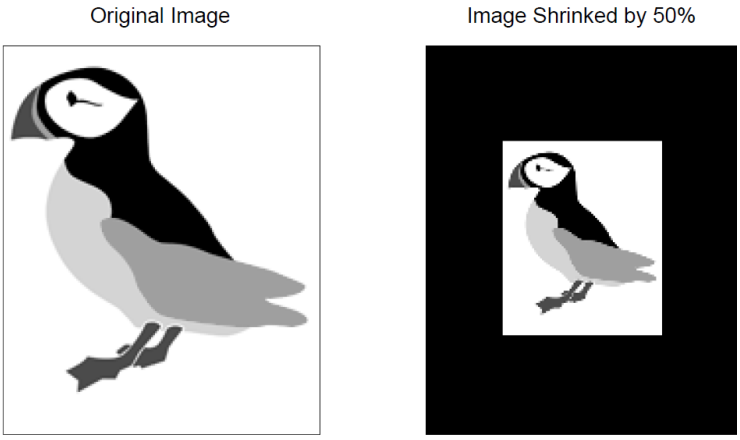
6(c)

```
subplot(2,2,1), title('Original Image'), imshow(s_img);
subplot(2,2,2), title('Image rotation by 45'), imshow(imrotate(s_img, 45));
subplot(2,2,3), title('Image rotaion by -45'), imshow(imrotate(s_img, -45));
subplot(2,2,4), title('Image rotaion by 180'), imshow(imrotate(s_img, 180));
```



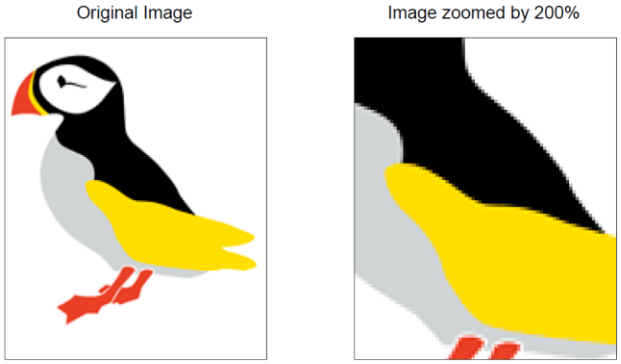
6(d)

```
[r c] = size(s_img);
f = 0.5;
im_50 = zeros(r, c, 'uint8');
shrinked = rgb2gray(imresize(s_img, f));
im_50(48:143, 40:120) = shrinked;


subplot(121), title('Original Image'), imshow(rgb2gray(s_img));
subplot(122), title('Image Shrinked by 50%'), imshow(im_50);
```



6(e)

```
f = 2;
im2 = imresize(s_img, f);
subplot(121), title('Original Image'), imshow(s_img);
subplot(122), title('Image zoomed by 200%'), imshow(im2(96:287, 81:241, :));
```



7.To understand various image noise models and to write programs for :
a. image restoration
b. Remove Salt and Pepper Noise
c. Minimize Gaussian noise
d. Median filter

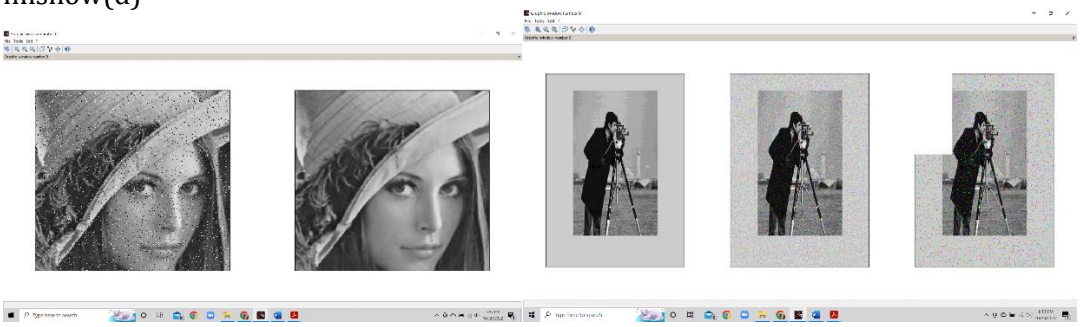
7(a)

```
a = imread('camera.png');
figure(1)
imshow(a)
```

```
b = imnoise(a,'gaussian')
figure(2)
imshow(b)
c = imnoise(a,'salt & pepper')
figure(3)
imshow(c)
d = imnoise(d,'speckle')
figure(4)
imshow(d)
imwrite(b,"salt_Lenna.jpg")
c = imnoise(a,'speckle')
imshow(c)
imwrite(c,"speckle.jpg")
```

7(b) – 7(d)

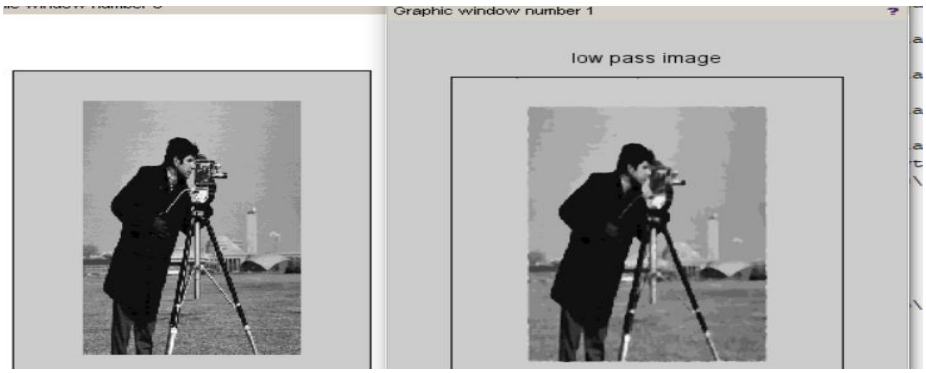
```
//7(b) Removal of salt & pepper noise and median filter
a = imread('noisy.png')
subplot(1,2,1),
imshow(I);
[m,n] = size(I)
for i = 2:m-1
   for j=2:n-1
   d(i,j) = median([a(i-1,j+1),a(i,j+1),a(i+1,j+1); a(i-1,j),a(i,j),a(i+1,j); a(i-1,j-1),a(i,j-1),a(i+1,j-1)]);
   end
end
subplot(1,2,2),
imshow(d)
```



8. Write and execute programs to use spatial low pass and high pass filters

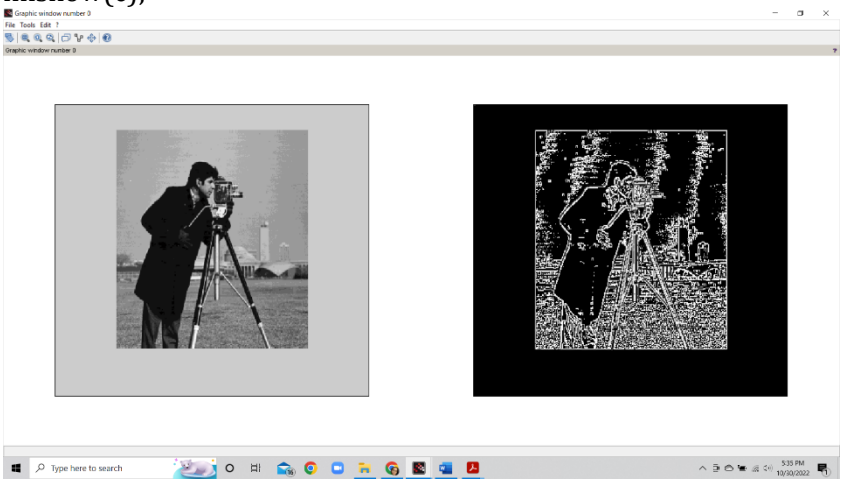
#Low pass

```
a=imread('camera.png');
b=size(a);
subplot(121)
imshow(a)
if size(b,2)==3
   a = rgb2gray(a);
end
a = imnoise(a,'salt & pepper');

n=input("Enter the size of mask");
n1=ceil(n/2);
a=double(a);
lpf=(1/n^2)*ones(n);
hpf=-lpf;
hpf(n1,n1)=(n^2-1)/n^2;

c=0;
h=0;
for i=n1:b(1)-n1
   for j=n1:b(2)-n1
      p=1;
      for k=1:n
         for l=1:n
         c(p)=a(i-n1+k,j-n1+l);
         p=p+1;
         end
      end

      d(i,j)=median(c);
      c=0;
   end
end
e=uint8(d);
subplot(122)
figure;imshow(e);title('low pass image');
```



#HIGH PASS

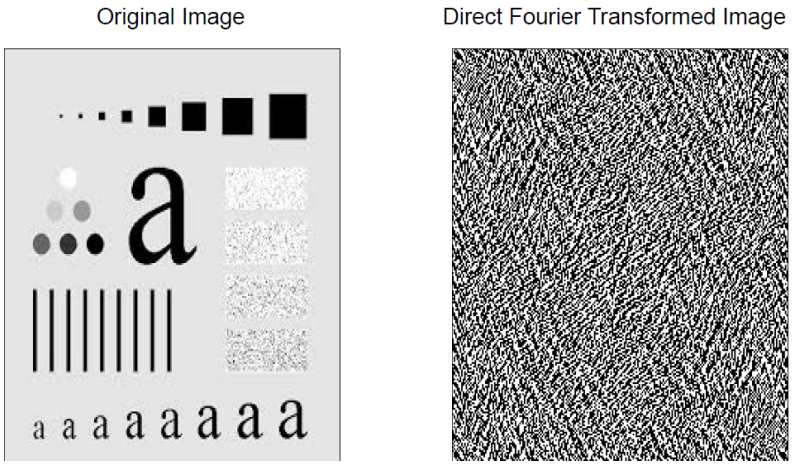
```
a1 = imread('camera.png');
subplot(121)
imshow(a1)
a = double(a1);
[m,n] = size(a);
w = [-1 -1 -1;-1 8 -1;-1 -1 -1];
for i = 2:m-1
   for j = 2:n-1
      b(i,j) = [w(1)*a(i-
1,j+1)+w(2)*a(i,j+1)+w(3)*a(i,j+1)+w(4)*a(i+1,j+1)+w(5)*a(i,j)+w(6)*a(i+1,j)+w(7)*a(i-1,j-
1)+w(8)*a(i,j-1)+w(9)*a(i+1,j-1)]/9
end
end
c = uint8(b);
subplot(122);
imshow(c);
```



9. Write and execute programs for image frequency domain filtering
a. Apply FFT on given image
b. Perform low pass and high pass filtering in frequency domain
c. Apply IFFT to reconstruct image

9(a)

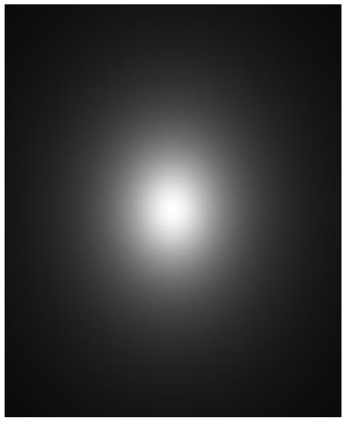
```
img = rgb2gray(imread('Test_images/sample.jpeg'));
ft_img = fft(double(img));
subplot(1,2,1), title('Original Image'), imshow(img);
subplot(1,2,2), title('Direct Fourier Transformed Image'),imshow(ft_img);
```



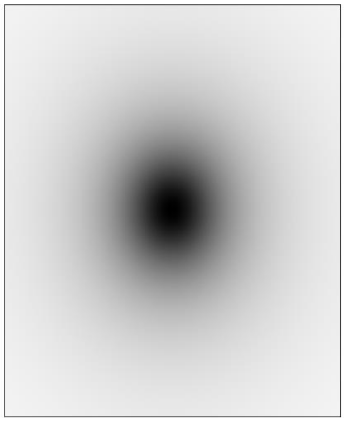
9(b)

```
G11 = mkfftfilter(img, 'butterworth1', 0.3);
H11 = 1 - G11;
subplot(121), title('DFT Butterworth Low Pass Image'), imshow(G11);
subplot(122), title('DFT Butterworth High Pass Image'),imshow(H11);
```

DFT Butterworth Low Pass Image



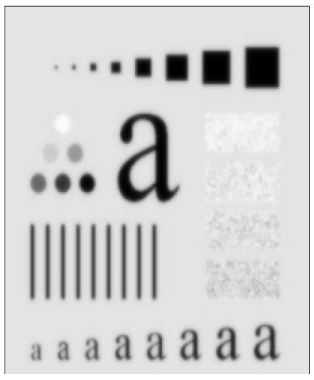
DFT Butterworth High Pass Image



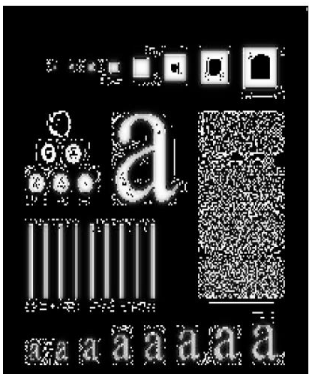
9(c)

```
S2 = ft_img .* fftshift(G11);
bwh_l = uint8(ifft(S2));
S2 = ft_img .* fftshift(H11);
bwh_h = uint8(ifft(S2));
subplot(121), title('DFT Butterworth Low Pass Image'), imshow(bwh_l);
subplot(122), title('DFT Butterworth High Pass Image'),imshow(bwh_h);
```

DFT Butterworth Low Pass Image

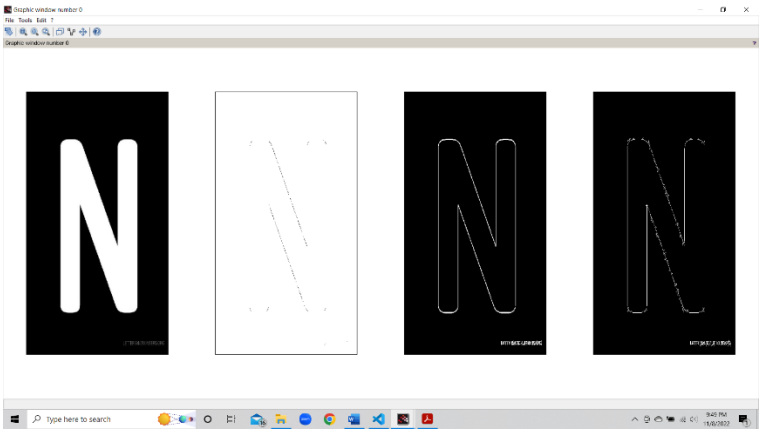


DFT Butterworth High Pass Image



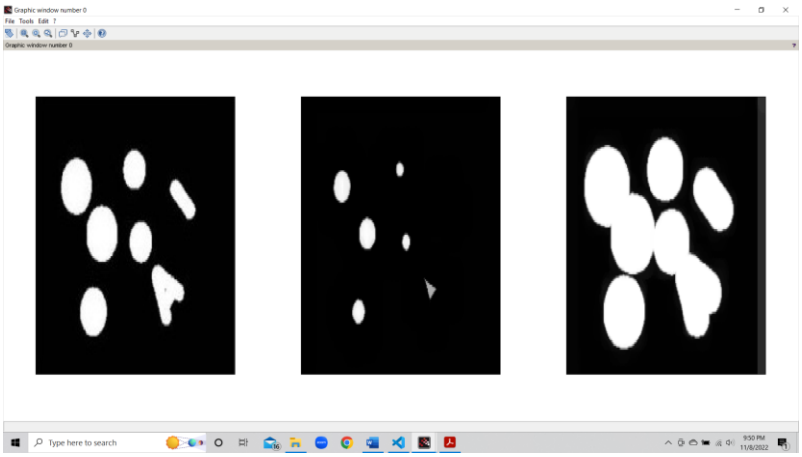
10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask

```
I = imread('edge.jpg');
subplot(1,4,1),
imshow(I)
b = edge(I,'sobel');
c = edge(I,'prewitt');
d = edge(I,'log');
e = edge(I,'canny');
subplot(1,4,2),
imshow(b)
subplot(1,4,3),
imshow(c)
subplot(1,4,4),
imshow(e);
```



11. Write and execute program for image morphological operations erosion and dilation.

```
I = imread('dilate.jpg')
//StructureElement = CreateStructureElement ('square',3);
//Im1 = ErodeImage (I, StructureElement );
SE = imcreatese('ellipse',15,15);
d = imdilate(I,SE);
e = imerode(I,SE);
subplot(131);
imshow(I)
subplot(132);
imshow(e)
subplot(133);
imshow(d)
```


Graphic window number 0
File Tools Edit ?
Graphic window number 0
Type here to search
9:30 PM
11/8/2022