SAQIB UR REHMAN
BCSF20A510

ASSIGNMENT NO. 2
COMPUTER VISION

Question No 1:

(a)

```
function f = Gaussian_Generator(w,sigma)
    s = 2 *sigma*sigma;% 2 sigma square
    GKernel = zeros(w,w);
    low = floor(w/2);

    for i = -low:low
        for j = -low : low
            r = sqrt(i*i+j*j);
            GKernel(i+low+1,j+low+1) = ((exp(-(r*r)/s))/(3.14 * s));
        end
    end
    f = (GKernel);
End
```

Input:

```
a = imread('im_q1.tif');
b =   Gaussian_Generator(3,1);
c  =  conv2(a,b);
imshow(uint8(a));
```

(b) 9x9

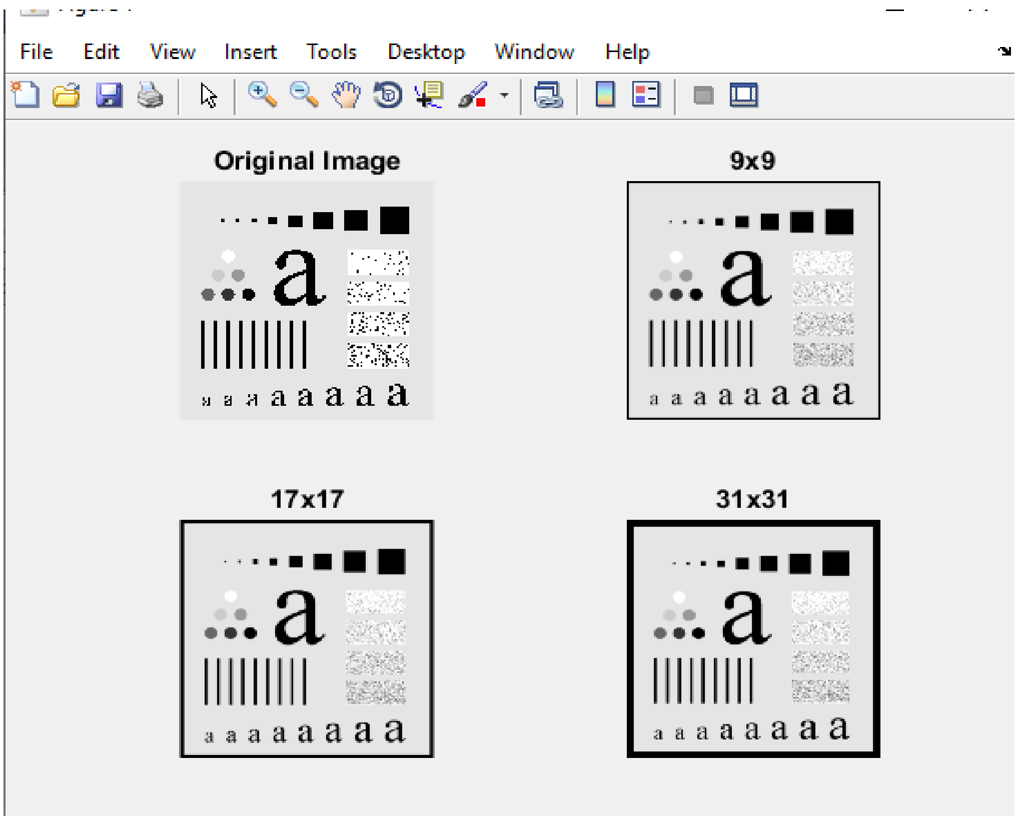
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	3	5	3	1	0	0
0	0	3	15	25	15	3	0	0
0	0	5	25	41	25	5	0	0
0	0	3	15	25	15	3	0	0
0	0	1	3	5	3	1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

17 x 17

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	3	5	3	1	0	0	0	0	0	0
0	0	0	0	0	0	3	15	25	15	3	0	0	0	0	0	0
0	0	0	0	0	0	5	25	41	25	5	0	0	0	0	0	0
0	0	0	0	0	0	3	15	25	15	3	0	0	0	0	0	0
0	0	0	0	0	0	1	3	5	3	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31 x 31

Very large matrix to show here.

C - 

Question No 2:

(a)

Both the above filters are linear because we know that any filter that performs sliding sum of products with the image pixels and they involve derivatives are linear.

(b)

The gradient of an image **measures how it is changing**. It provides two pieces of information. The magnitude of the gradient tells us how quickly the image is changing, while the direction of the gradient tells us the direction in which the image is changing most rapidly.We will observe no change in the output if gradient filter is applied first And then averaging filters.

(c) _

The S filter can be obtained by multiplying the B and G and then adding G to the result . Then, the resultant matrix will be a sobel filter which finds derivatives in x direction.

Question No 3:

(A) (B) and (C) part

```
img  = imread('im_q3.tif');
img = double(img);
lap_filter_90  = [0 1 0;1 -4 1;0 1 0];
lap_filter_45 = [1 1 1;1 -8 1;1 1 1];
unscaled_result1 = conv2(img,lap_filter_90,'same');
unscaled_result2 = conv2(img,lap_filter_45,'same');

% perform scaling of result 1
min_value = min(result1(:));
result1 = result1 - min_value;
% now normalize
max_value = max(result1(:));
result1 = result1./max_value;
result1 = result1 * 255;

% perform scaling of result 2
min_value1 = min(result2(:));
result2 = result2 - min_value1;
% now normalize
max_value1 = max(result2(:));
result2 = result2./max_value1;
result2 = result2 * 255;

%Now sharpen result 1 & result2
sharp_image1 = img - unscaled_result1;
sharp_image2 = img - unscaled_result2;



subplot(2,2,1);imshow(uint8(result1));title('90 degree isotropic with scaling');
subplot(2,2,2);imshow(uint8(result2));title('45 degree isotropic with  scaling');
subplot(2,2,3);imshow(uint8(sharp_image1));title('90 degree isotropic sharp');
subplot(2,2,4);imshow(uint8(sharp_image2));title('45 degree isotropic sharp');
```

Output:

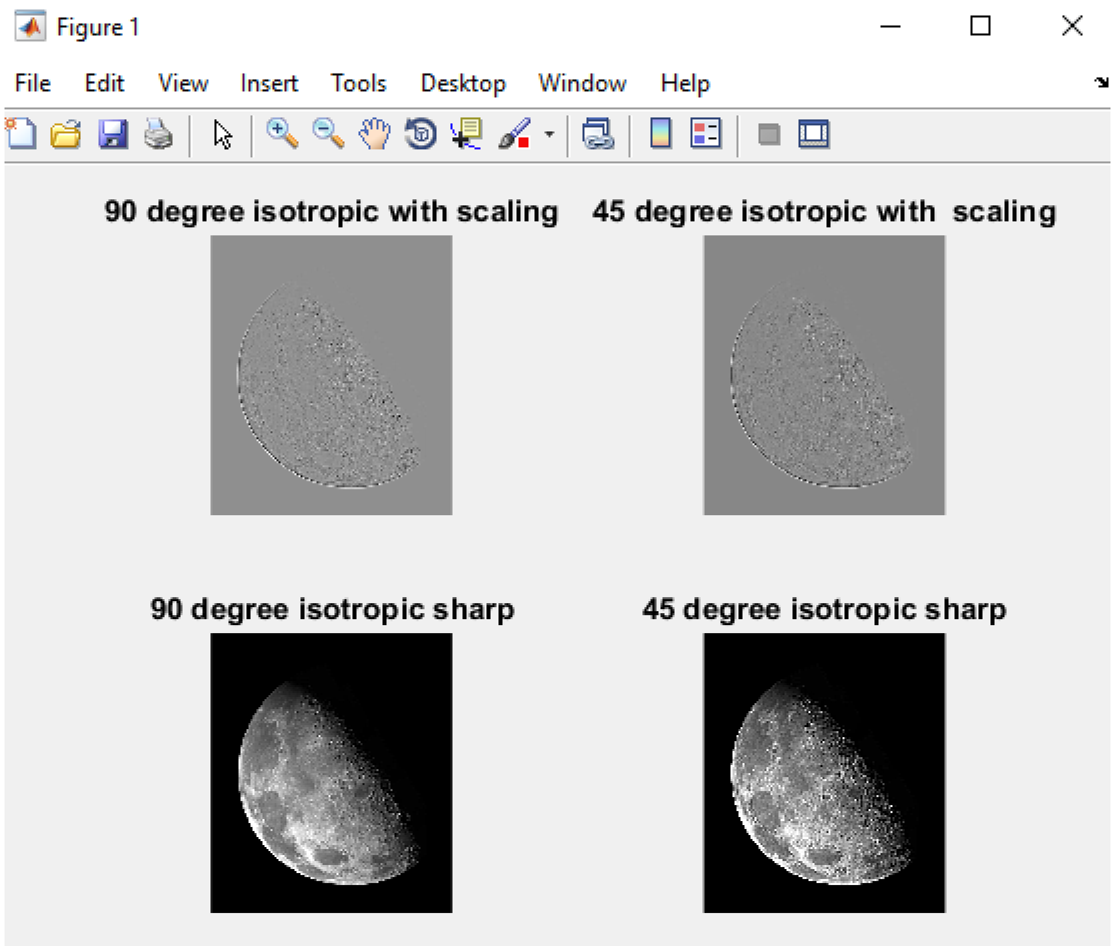


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
90 degree isotropic with scaling
45 degree isotropic with scaling
90 degree isotropic sharp
45 degree isotropic sharp

Original Image:

END OF ASSIGNMENT