LAB 04 REPORT

Part 2

If $x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$, then $x[m,n] = \sum_{k=-\infty}^{\infty}\sum_{\ell=-\infty}^{\infty} x[k,\ell]\delta[m-k,$

Signal $x[m,n]$ can also be obtained by superposition of shifted impulses.

Since the system is Time-Invariant , $\delta[m-k,n-\ell] \xrightarrow{S} h[m-k,n-\ell]$ (impulse

Then by using the linearity property , following input-output relation of the 2D

can be acquired :

$$y[m,n] = \sum_{k=-\infty}^{\infty}\sum_{\ell=-\infty}^{\infty} x[k,\ell]\,h[m-k,n-\ell] = x[m,n] ** h[m,n]$$

let $u = m-k \Rightarrow k = m-u$

$v = n-\ell \Rightarrow \ell = n-v$

$$x[m,n] ** h[m,n] = \sum_{v=-\infty}^{\infty}\sum_{u=-\infty}^{\infty} h[v,u]\,x[m-v,n-u] = h[m,n] ** x[m,n]$$ ✓

Part 3

$$y[m,n] = \sum_{k=0}^{M_h-1}\sum_{\ell=0}^{N_h-1} h[k,\ell]\,x[m-k,n-\ell]$$

Nonzero Conditions :

① $h[m,n]$ can be non zero within $0 \le k \le M_h-1$, $0 \le \ell \le N_h-1$

② $x[m-k,n-\ell]$ can be nonzero within $0 \le m-k \le M_x-1$, $0 \le n-\ell \le N_x-1$

$y[m,n]$ can be nonzero within $0 \le m \le M_y-1$, $0 \le n \le N_y-1$

①,② $\Rightarrow$ $0 \le m \le M_h+M_x-2$, $0 \le n \le N_h+N_x-2$

$$\Rightarrow M_y-1 = M_h+M_x-2 \Rightarrow \boxed{M_y = M_h+M_x-1}$$

$$N_y-1 = N_h+N_x-2 \Rightarrow \boxed{N_y = N_h+N_x-1}$$

Code is Given Below

⇓

Part 3 Code:

```
x=[8 1 6; 3 5 7; 4 9 2];
h=[1 3; 4 2];
y=DSLSI2D(h,x)


function [y] = DSLSI2D(h,x)
     [Mh,Nh]=size(h);
     [Mx,Nx]=size(x);

    My = Mh+Mx-1;
    Ny = Nh+Nx-1;

    y = zeros(My,Ny);

    for k=0:Mh-1
        for l=0:Nh-1
            y(k+1:k+Mx,l+1:l+Nx)=…
y(k+1:k+Mx,l+1:l+Nx)+h(k+1,l+1)*x;
        end
    end
end
```

Part 4:

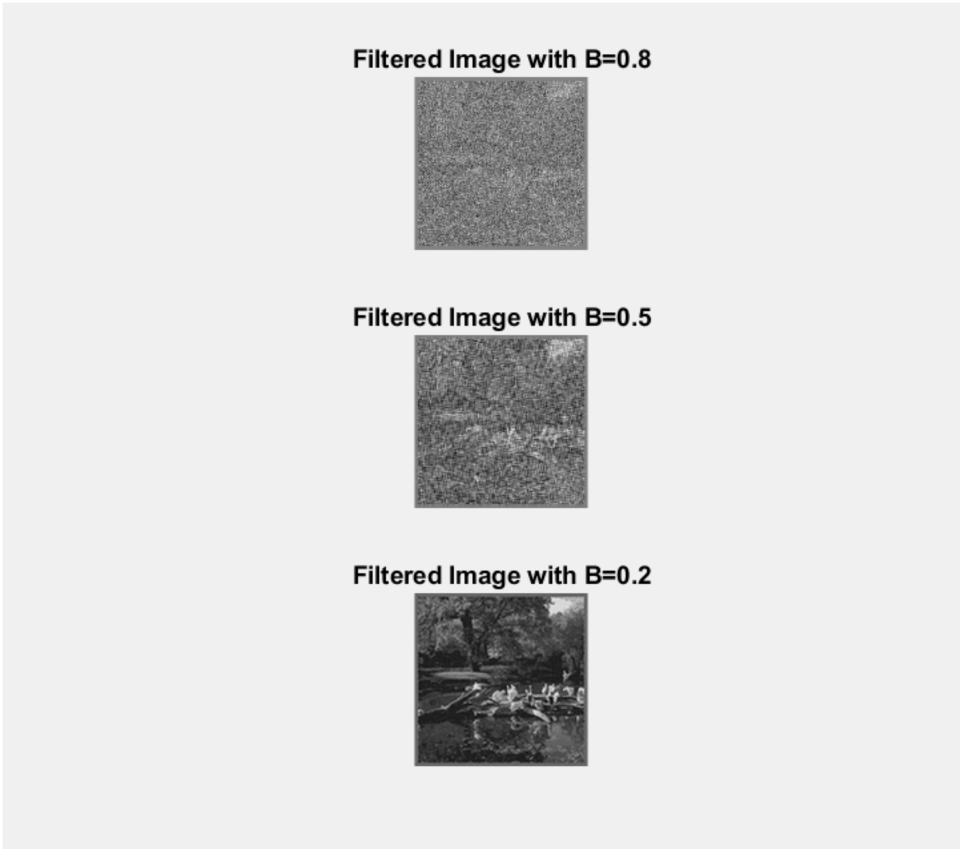
Code for preparing h:

```
function [h]= FINDH(B,Mh,Nh)
    h=zeros(Mh,Nh);
    for m=0:Mh-1
        for n=0:Nh-1
            x1=sym(B*(m+1/2-Mh/2));
            s1=sinc(x1);
            x2=sym(B*(n+1/2-Nh/2));
            s2=sinc(x2);
            h(m+1,n+1)=s1*s2;
        end
    end
end
```

Part 4 Code:

```
x=ReadMyImage('Part4.bmp');
imshow(x,[]);
title("Original Image");
figure;

D17=rem(21703190,17);
Mh=20+D17;
Nh=Mh;

h1=FINDH(0.8,Mh,Nh);
filtered_im1 = DSLSI2D(h1,x);
h2=FINDH(0.5,Mh,Nh);
filtered_im2 = DSLSI2D(h2,x);
h3=FINDH(0.2,Mh,Nh);
filtered_im3 = DSLSI2D(h3,x);
subplot(3,1,1),imshow(filtered_im1,[]);
title("Filtered Image with B=0.8");
subplot(3,1,2),imshow(filtered_im2,[]);
title("Filtered Image with B=0.5");
subplot(3,1,3),imshow(filtered_im3,[]);
title("Filtered Image with B=0.2");
```

Images:


Filtered Image with B=0.8
Filtered Image with B=0.5
Filtered Image with B=0.2

Part 5:

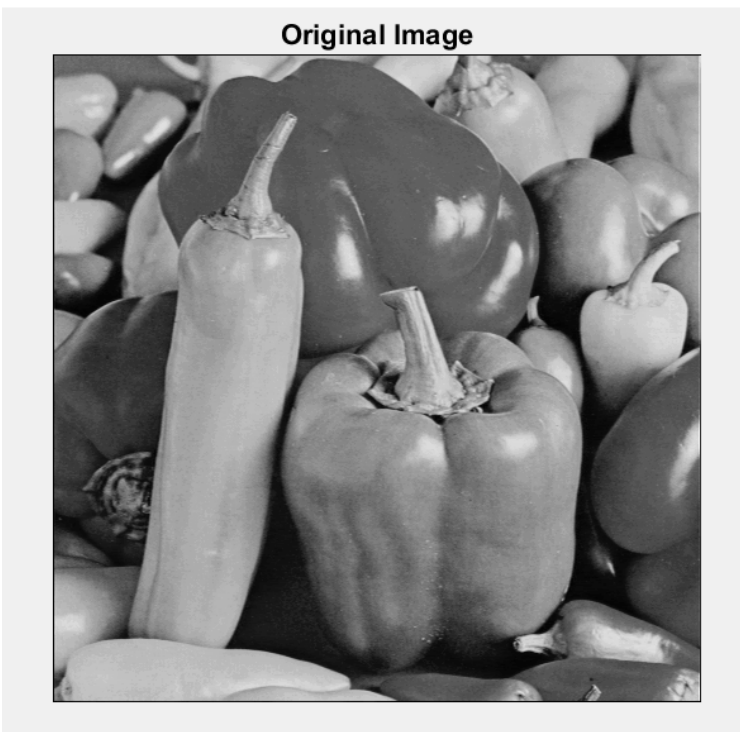
① In $s_1[m,n]$, vertical edges of the image are emphasized.

② In $s_2[m,n]$, horizontal edges of the image are emphasized. Whe
both images, I can see that the edges in the image 1 seem to be dra
and in the image 2, they seem to be drawn with horizontal lines. T
between the images of h_1 and h_2 filters is h_1 filters catch the high freque
and h_2 filter catch the high frequency in horizontal direction. They both

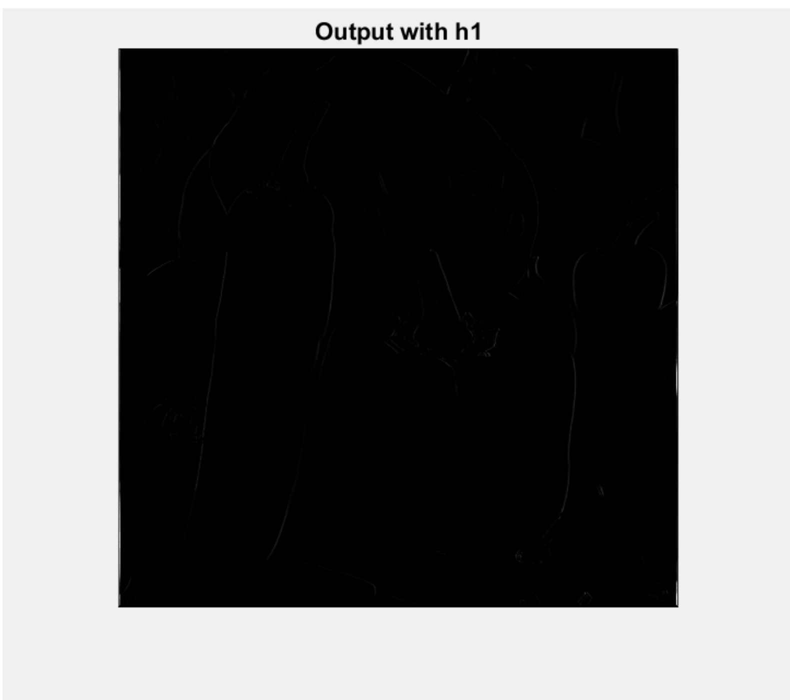
③ Both vertical edges and horizontal edges are visible in the third image. In
because h_3 contains both h_1 and h_2. Thus, new filter, h_3, finds both
$h_3 = \alpha h_1 + \beta h_2$ so by adjusting the coefficients, α and β, the propor
can be decided. For example, if $\alpha > \beta$, more vertical edges will be fo

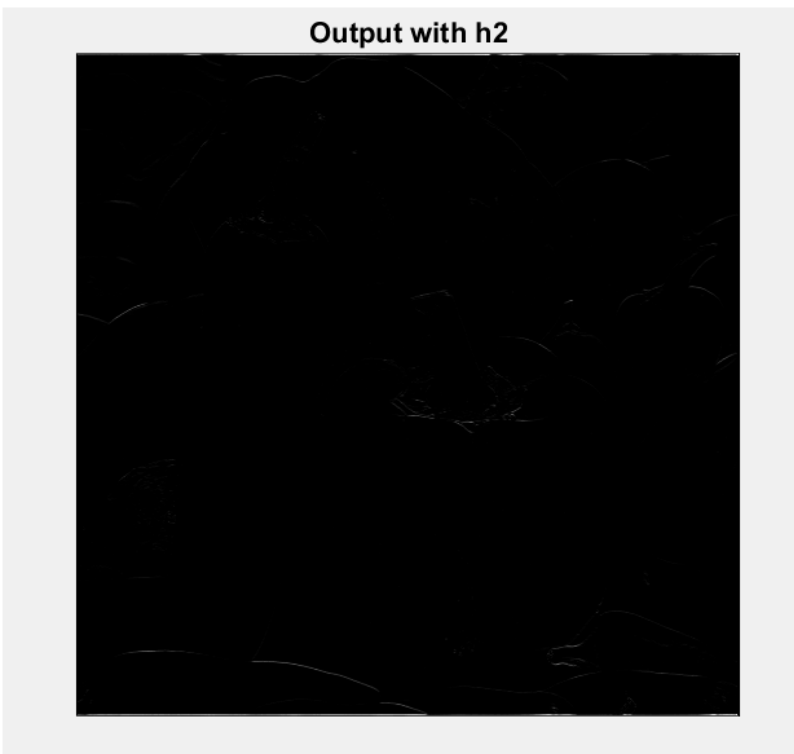
Part 4

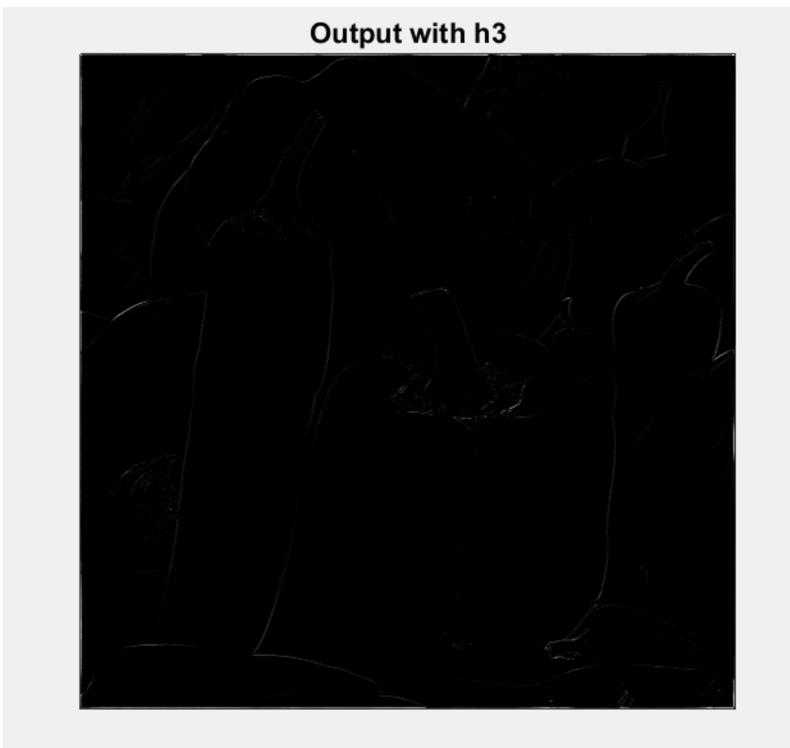
$\beta = 0.2$ is the most appropriate one but this does not mean that as clearer and all noises are removed. I tried $\beta = 0.01$ and I got blurred image were indistinguishable so original image was damaged.



Images:


Output with h1


Output with h2


Output with h3

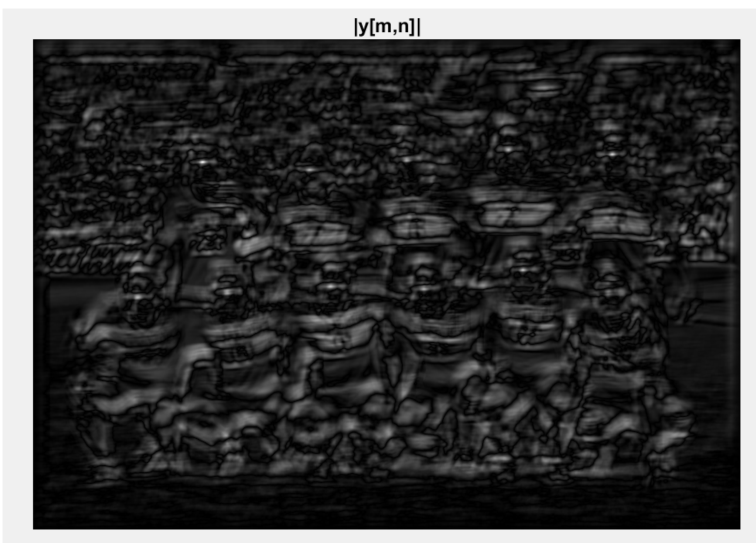
Part 5 Code:

```
x=ReadMyImage('Part5.bmp');
imshow(x,[])
title("Original Image");
figure;
h1=[1 -1;0 0];
y1=DSLSI2D(h1,x);
s1=y1.*y1;
imshow(s1,[]);
title("Output with h1");
figure;
%DisplayMyImage(s1)
h2=[1 0;-1 0];
y2=DSLSI2D(h2,x);
s2=y2.*y2;
imshow(s2,[]);
title("Output with h2");
figure;
h3 = 0.5*h1+0.5*h2;
y3=DSLSI2D(h3,x);
s3 = y3.*y3;
imshow(s3,[]);
title("Output with h3");
```

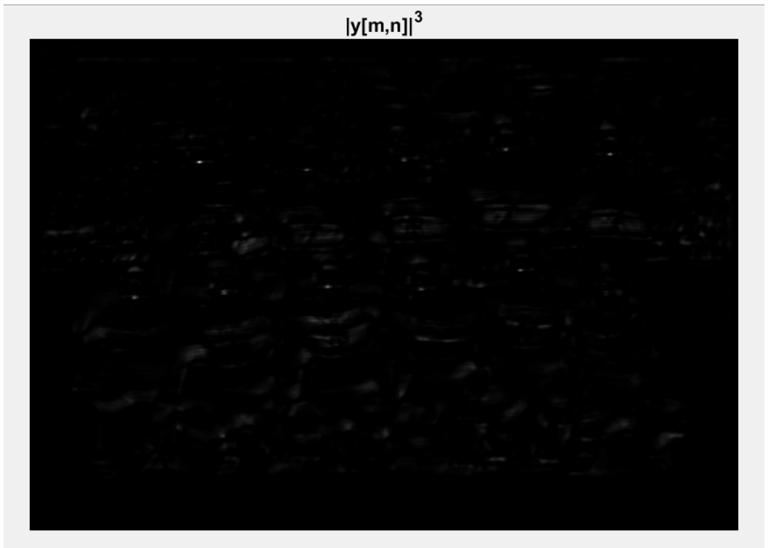
Part 6 Code:

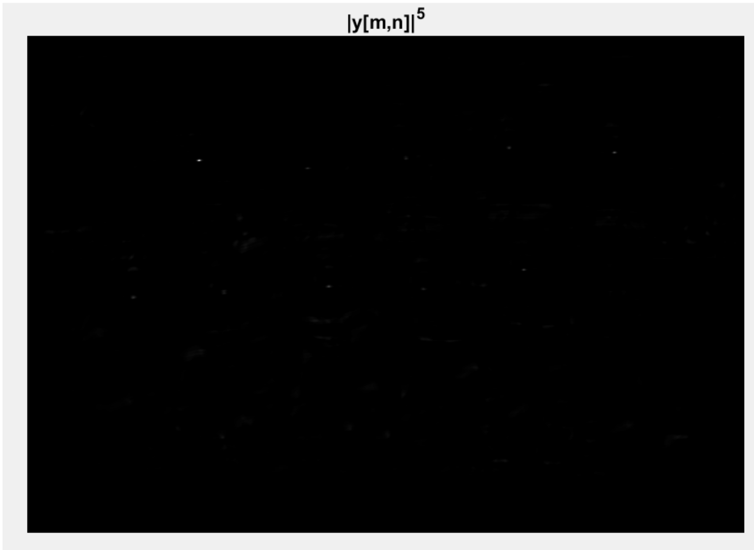
```
x=ReadMyImage('Part6x.bmp');
DisplayMyImage(x);
h=ReadMyImage("Part6h.bmp");
DisplayMyImage(h);

y=DSLSI2D(h,x);
```

```
imshow(abs(y),[]);
title("|y[m,n]|");
figure;
imshow(abs(y).^3,[]);
title("|y[m,n]|^3");
figure;
imshow(abs(y).^5,[]);
title("|y[m,n]|^5");
```



Part 6:


|y[m,n]|^3


|y[m,n]|^5

Part 6

① The bright points occur at the middle of the eyes of national so
I always see a face at the bright point but one face doesn't have a b

② $|y[m,n]|^3$ is sufficient for me to distinguish bright points easily. When I
some bright points are hardly seen due to the black background. When I did |y
see false bright points due to the crowded background. I think method is prett
10 faces from 11 faces.