EEE321 - Signals & Systems

Lab Assignment IV

Mohammed Abed (21903608)

Part 1:

-The image was read and displayed successfully.

Part 2:

-The Derivation of the 2D Convolution is uploaded in a separate pdf file.

Part 3:

-The answer of the analytical part is the separate pdf file.

Code:

```
function [y]=DSLSI2D(h,x)

Mx = height(x);

Nx = width(x);

Mh = height(h);

Nh = width(h);

N_y = Nx + Nh;

M_y = Mx + Mh;

y = zeros(M_y -1,N_y -1);
```

```
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:

```
D = 21903608;
D7 = rem(D, 7);
Mh = 30 + D7;
Nh = 30 + D7;
b = 0.7;
b2 = 0.4;
b3 = 0.1;
x = ReadMyImage('Part4.bmp');
for m=1:Mh-1
for n=1:Nh-1
        h1 (m, n)= sinc(b*(m-((Mh-1)/2))).*sinc(b*(n-((Nh-1)/2)));
        h2 (m, n)= sinc(b2*(m-((Mh-1)/2))).*sinc(b2*(n-((Nh-1)/2)));
```

```
        h3 (m, n)= sinc(b3*(m-((Mh-1)/2))).*sinc(b3*(n-((Nh-1)/2)));
 end
end
denoised_x =  DSLSI2D(h1,x);
denoised_x2 =  DSLSI2D(h2,x);
denoised_x3 =  DSLSI2D(h3,x);
figure
subplot(2,2,1)
imshow(x);
subplot(2,2,2)
imshow(denoised_x);
subplot(2,2,3);
imshow(denoised_x2);
subplot(2,2,4);
imshow(denoised_x3)
```

Figure 1: Denoised Images with Different Values of the Parameter B Shown on the Same Plot

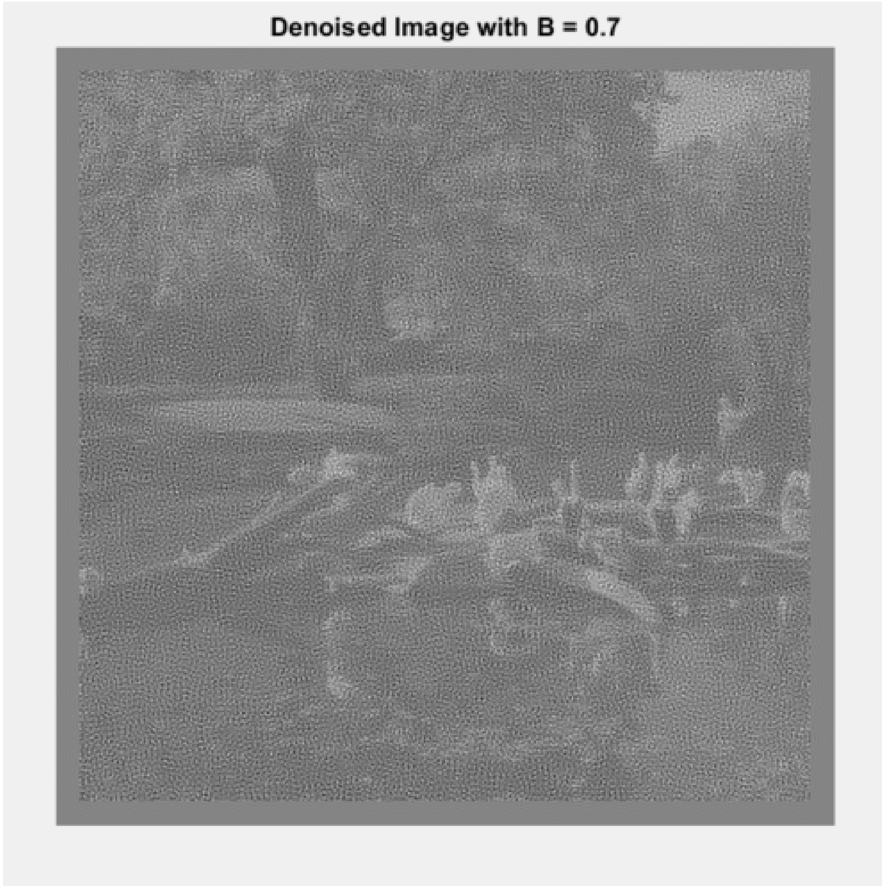

Figure 2: Denoised Image with B = 0.7

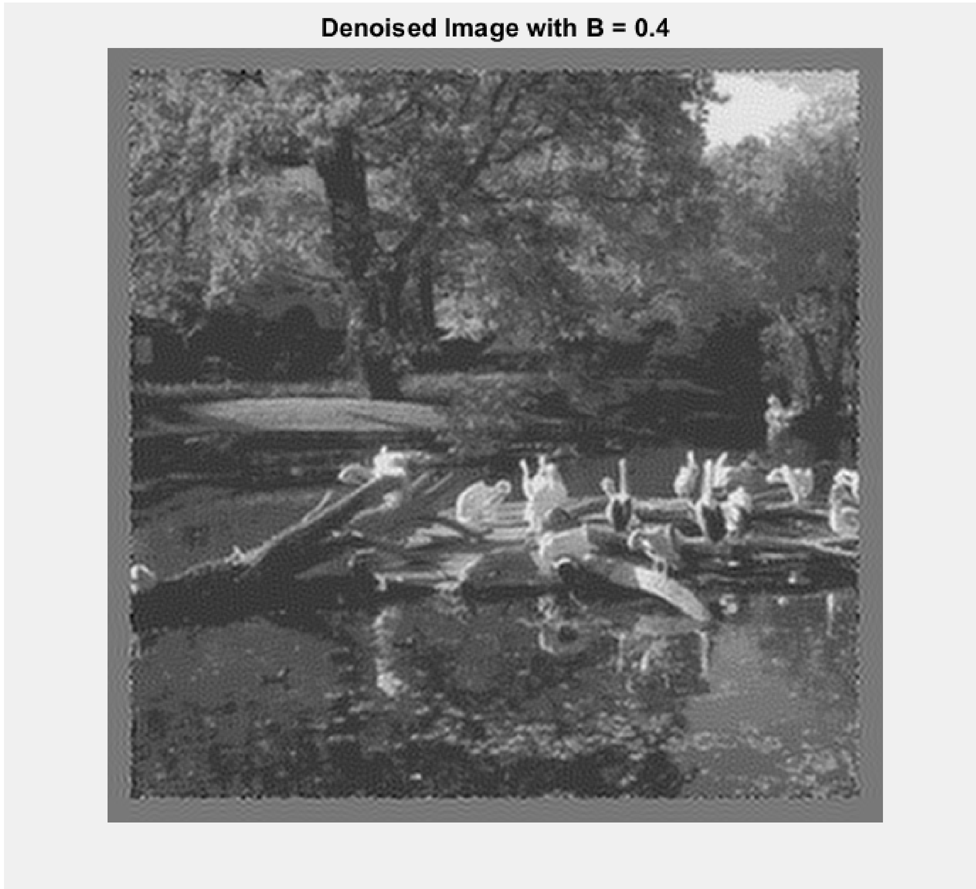

Figure 3: Denoised Image with B = 0.4

Figure 4: Denoised Image with B = 0.1

Comments: the free parameter B = 0.4 is the most appropriate for denoising the image. For B = 0.7 the noise was minimally reduced, and for B = 0.1 the noise was significantly reduced, however it damaged (distorted) the image.

Part 5:

Code:

```
%% Edge Detection

x = ReadMyImage('Part5.bmp');

% DisplayMyImage(x);

h1 = [0.5 -0.5];

h2 = [0.5 ; -0.5];

h3 = 0.5*h1 + 0.5*h2;

y1 = DSLSI2D(h1,x);

s1 = y1.*y1;

figure

subplot(1,2,1)

imshow(y1,[]);

subplot(1,2,2);

imshow(s1,[]);

y2 = DSLSI2D(h2,x);

s2 = y2.*y2;

figure

subplot(1,2,1);

imshow(y2,[]);
```

```
subplot(1,2,2);

imshow(s2,[]);

y3 = DSLSI2D(h1,x);

s3 = y3.*y3;

figure

subplot(1,2,1)

imshow(y3,[]);

subplot(1,2,2)

imshow(s3,[])
```

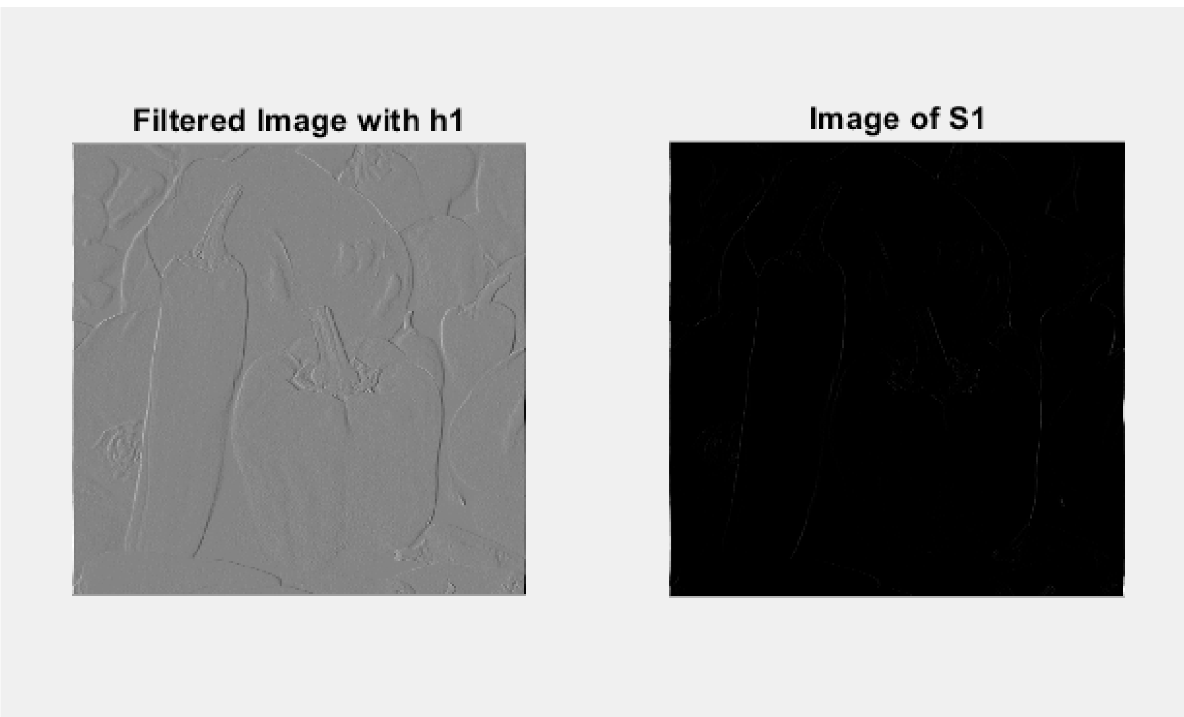

Figure 5: Image Filtered with $h_1[m,n]$ (Vertical Edges Defined)

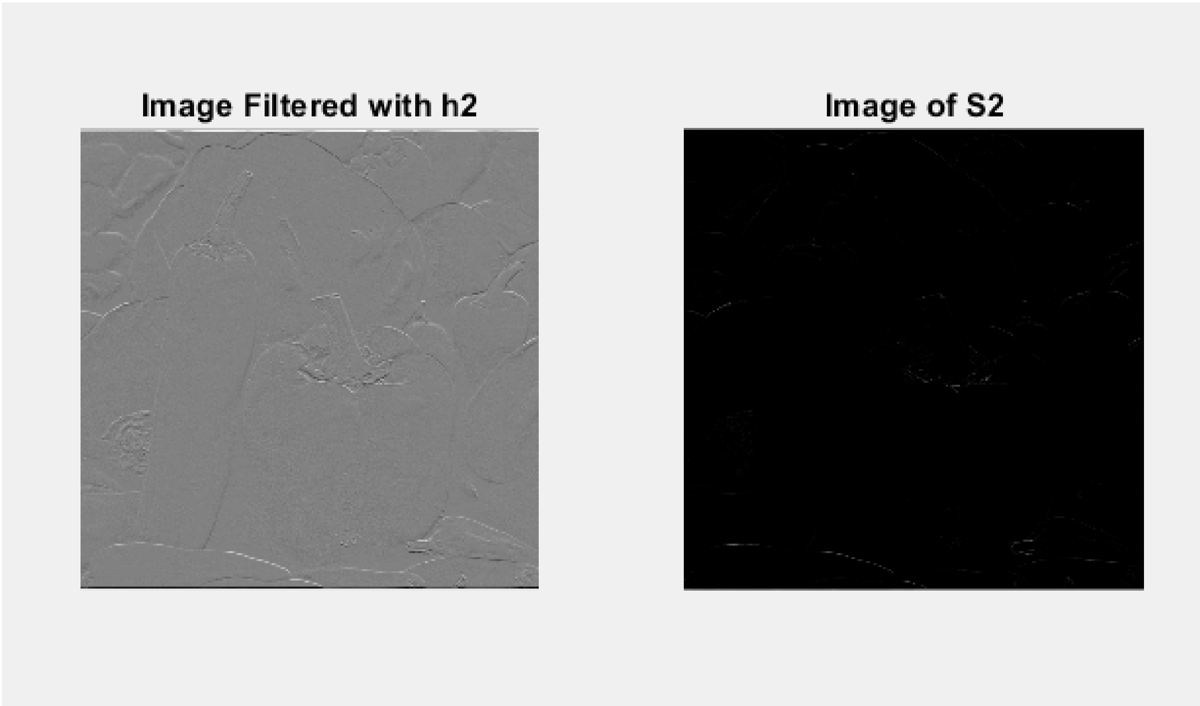

Figure 6: Image Filtered with $h_2[m,n]$ (Horizontal Edges Defined)

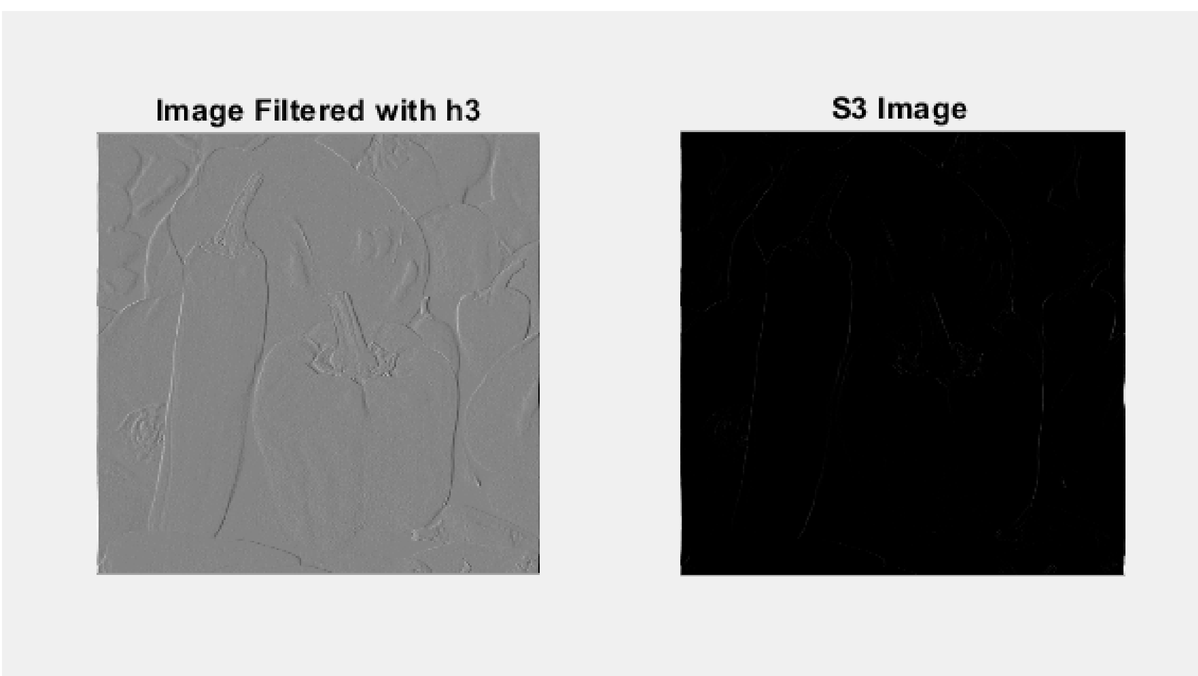

Figure 7: Image Filtered with $h_3[m,n]$ (Vertical and Horizontal Edges Defined)

Comments: for the first HPF($h_1[m,n]$), the vertical edges of the image were emphasized. For the second HPF ($h_2[m,n]$), the horizontal edges were emphasized. Finally, for the linear combination of the high pass filters ($h_3[m,n]$), which is another HPF, the effect of both filters were seen (vertically and horizontally), and that is because the image was convolved with both filters due to the linearity of the operation.

Part 6:

Code:

```
x6 = ReadMyImage('Part6x.bmp');

h6 = ReadMyImage('Part6h.bmp');

y6 = DSLSI2D(h6,x6);

DisplayMyImage(abs(y6));

DisplayMyImage((abs(y6)).^3);

DisplayMyImage((abs(y6)).^5);
```

Figure 8: The Image of abs(y[m,n])

Figure 9: The Image of abs(y[m,n])^3

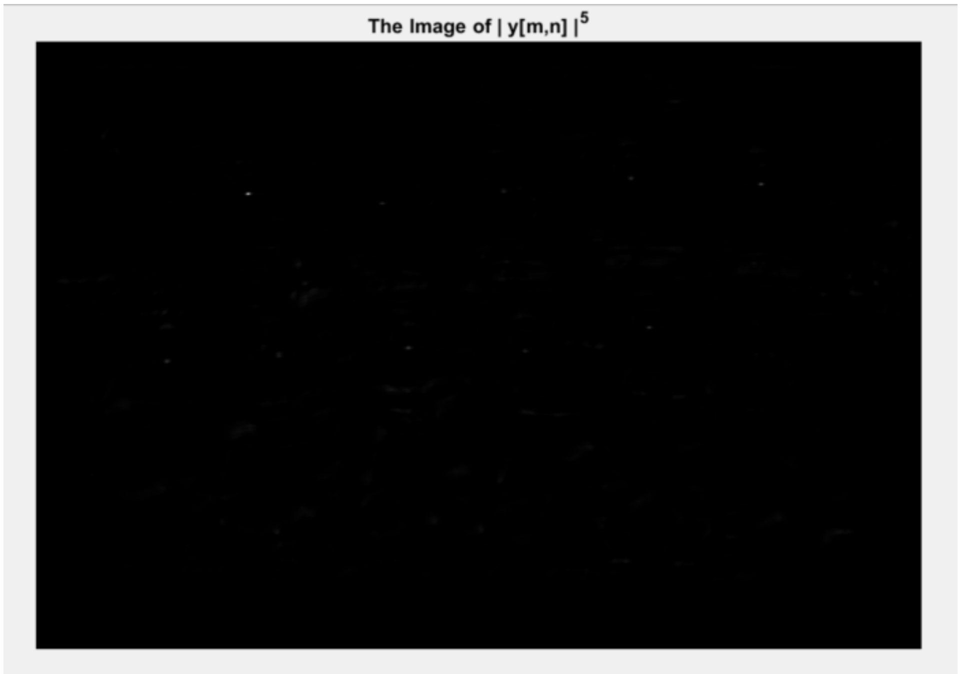

Figure 10: The Image of abs(y[m,n])^5

Comments: For the image produced by the signal abs(y[m,n]), all faces had bright dots, and it was hard to distinguish the detected face. Similarly, for the image produced by the signal abs(y[m,n])^3, it was hard to navigate the desired face, almost all faces had bright spots on them. However, for the image produced from the signal abs(y[m,n])^3, there was only one bright spot, which makes the third approach the best for pattern recognition.