Part 2

$$\delta[m,n] = \begin{cases} 1 & m,n=0 \\ 0 & \text{otherwise} \end{cases} \qquad x[m,n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k,l]\,\delta[m-k,n-l]$$

$x[m,n]$ is superposition of shifted 2D impulse signals. We can assume $x[k,l]$ as the coefficients of $\delta[m-k,n-l]$

we can write $\delta[n-k] = h[n-k]$ because of the property of space invariance of the system.

By Linearity property, we have

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

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

$$= x[m,n] ** h[m,n]$$

Part 4

As it can be seen by figures, the most clear one is when $B=0.2$. when B values increases quality of the photo decreases and when B decreases the quality of the photo increases

Part 5

For y_1, vertical lines between the objects and colors are emphasized

For y_2, horizontal lines between the objects and colors are emphasized,

For y_3, it uses a combination of vertical and horizontal filters therefore edges are more apparent and it is much more similar to original image.

Part 6

$|y[m,n]|$

There are a lot of bright points in the image but the most brightest ones are near to the center of football players' faces, or let say their noses,

For $|y[m,n]|^4$, bright points are much more visible and understable. Also it is easier to differ from other points than last one.

For $|y[m,n]|^6$, most of the bright points are disappeared

The most successful one is clearly $|y[m,n]|^4$ because there are less bright points than $|y[m,n]|$ and more accurate and gives true values than $|y[m,n]|^6$.

Atahan İyiekici
21801985
EEE321-001
Lab Report 4

EEE321 Lab Report 4

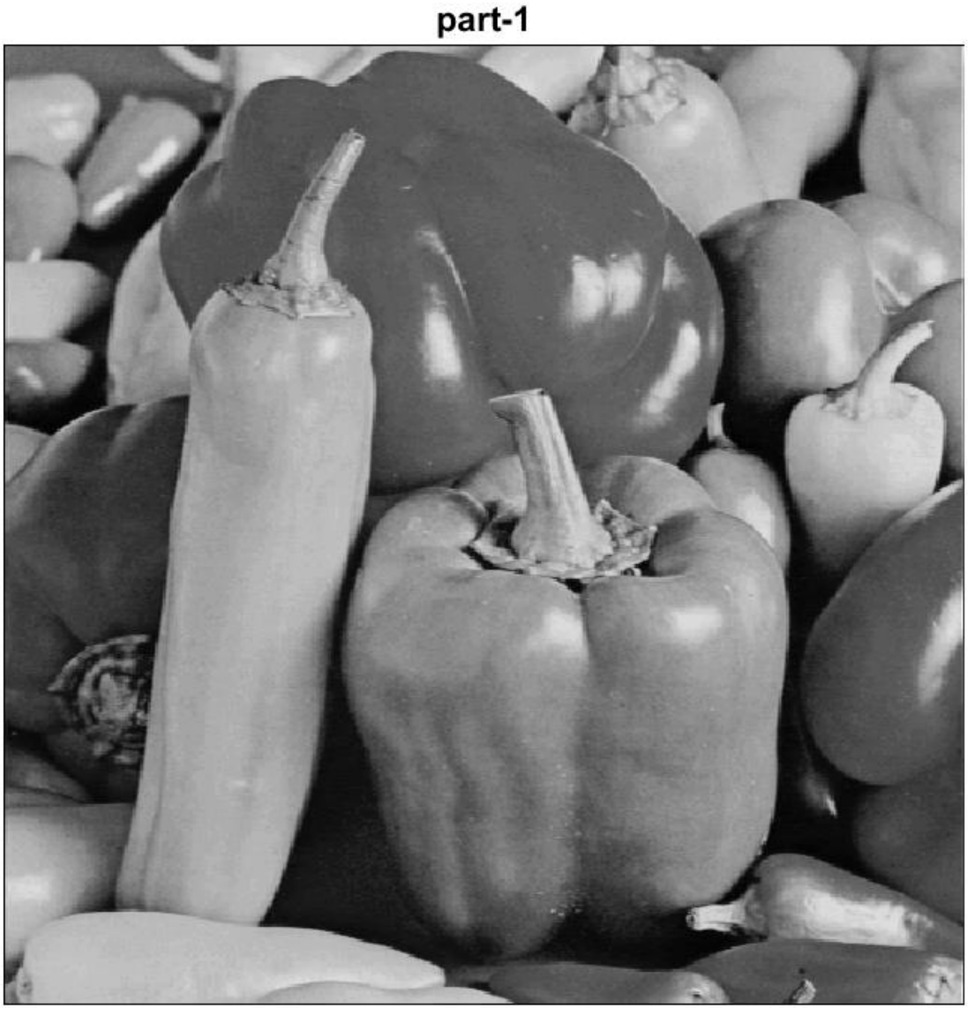

Figure 1: Output image of Part 1

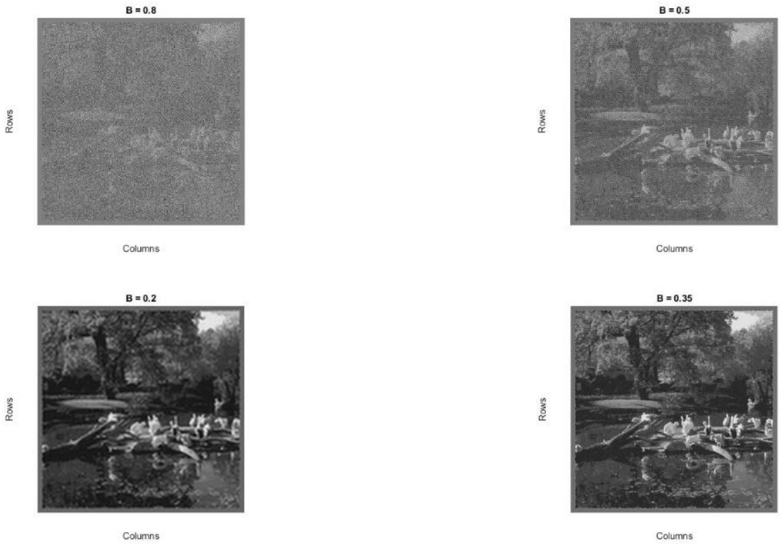

Figure 2: Output images of Part 4 with original image

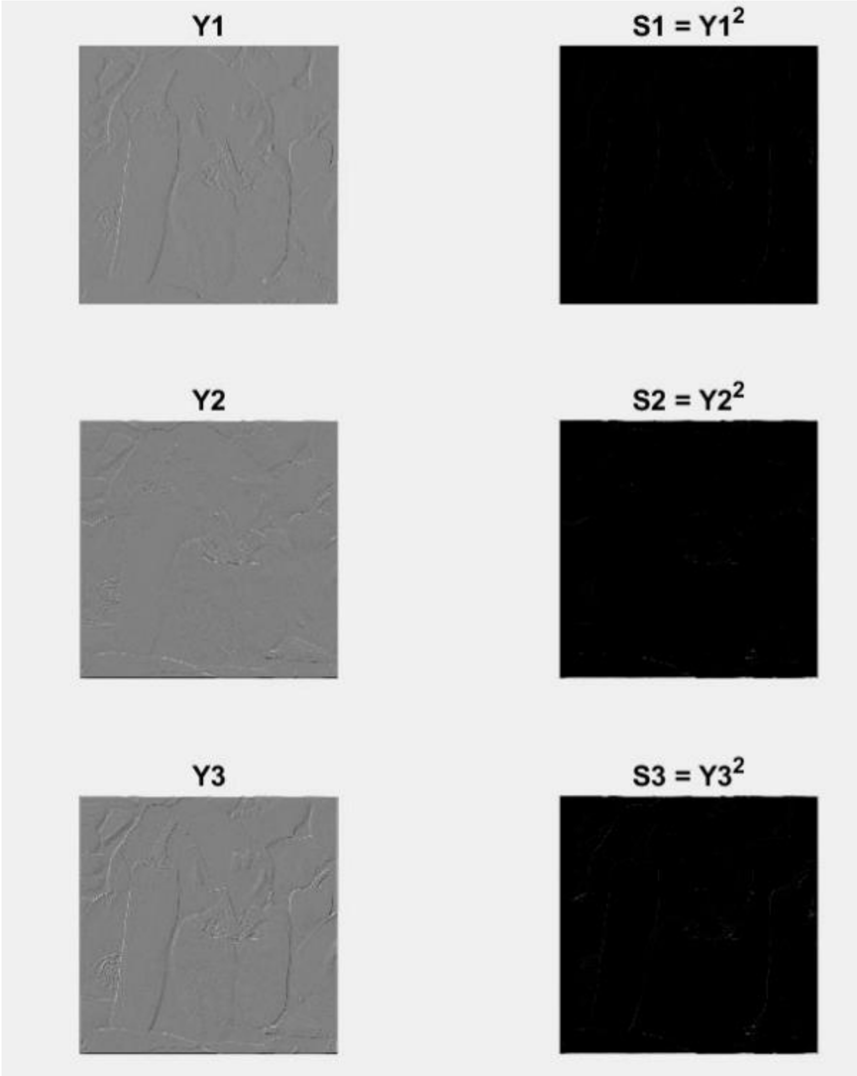

Figure 3: Filtered images from Part 5

Figure 4: Input/Given image for Part 6

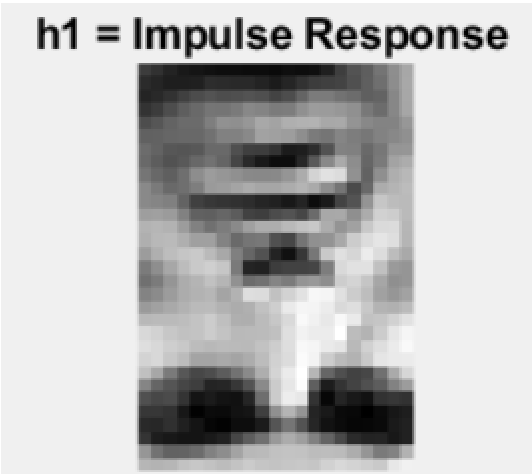

Figure 5: Impulse response

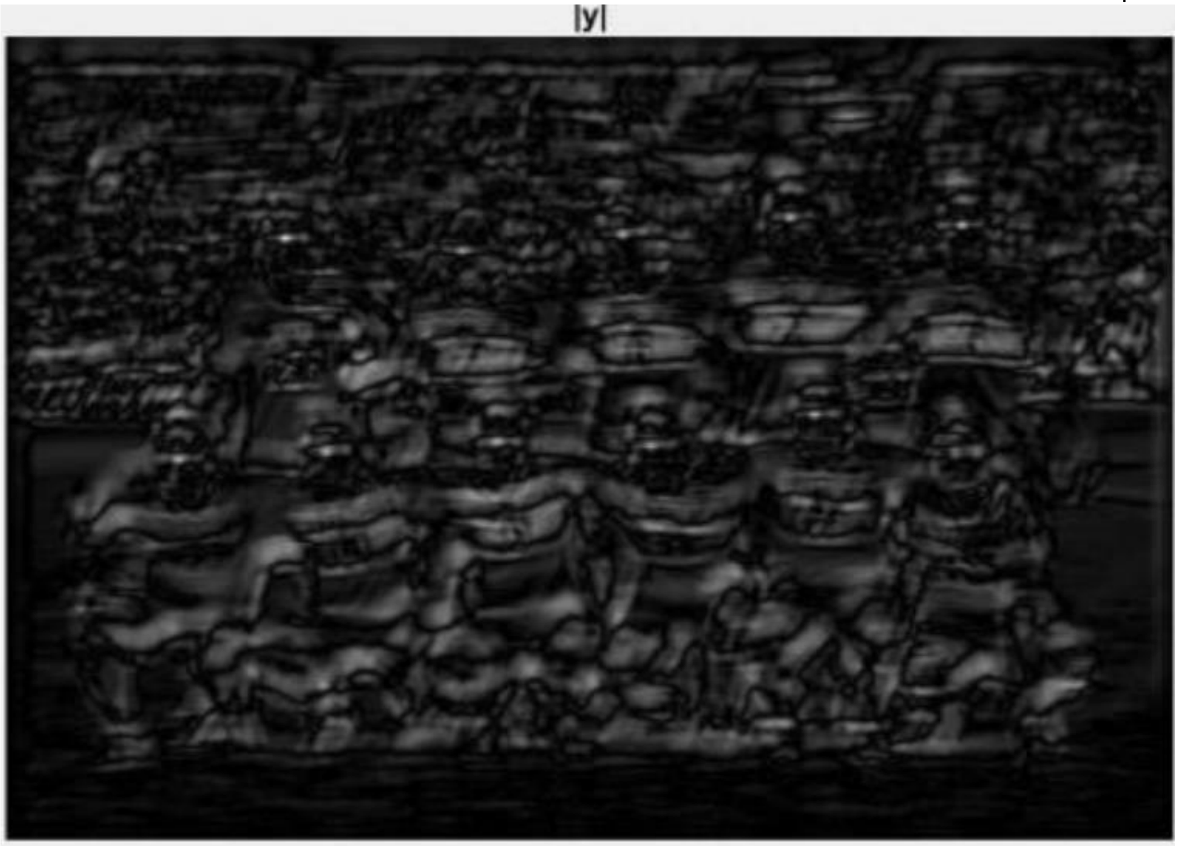

Figure 6: The image when |y[m,n]|

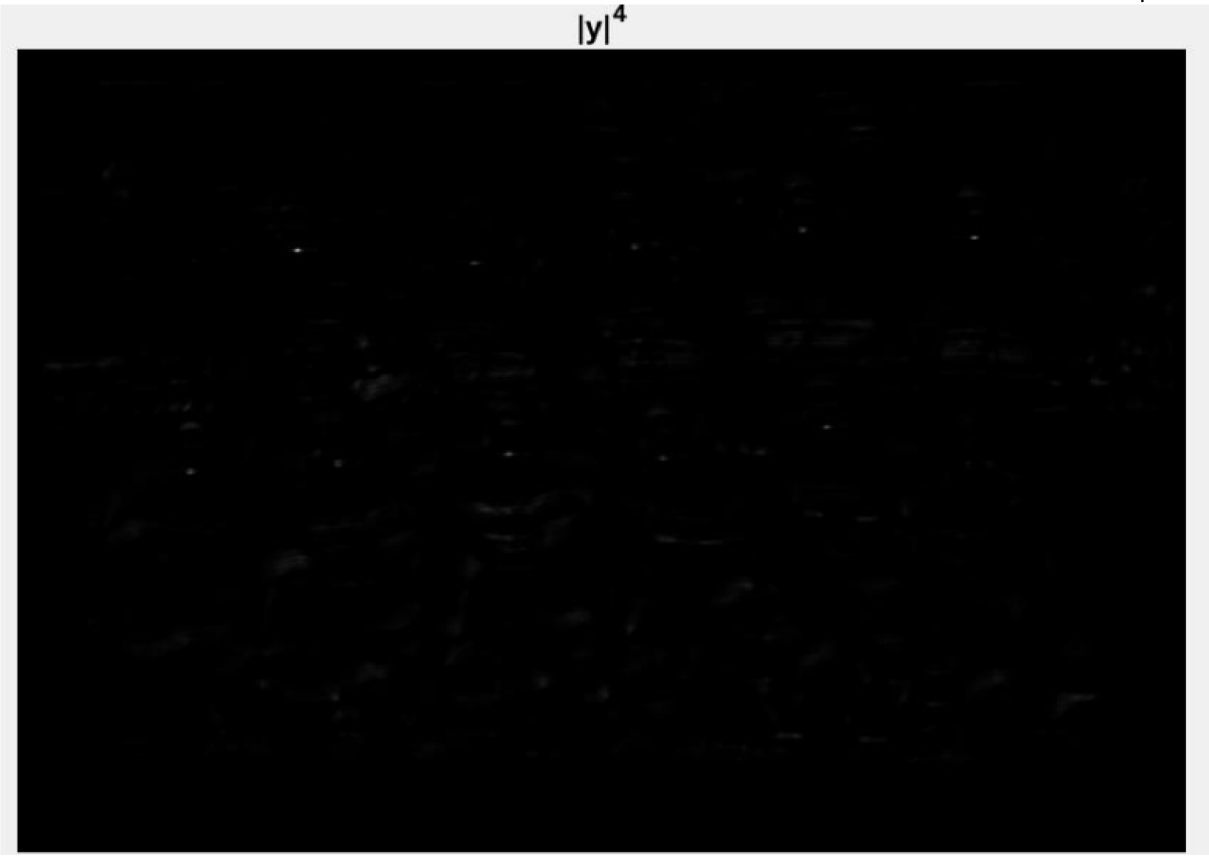

Figure 7: The image when |y[m,n]|^4

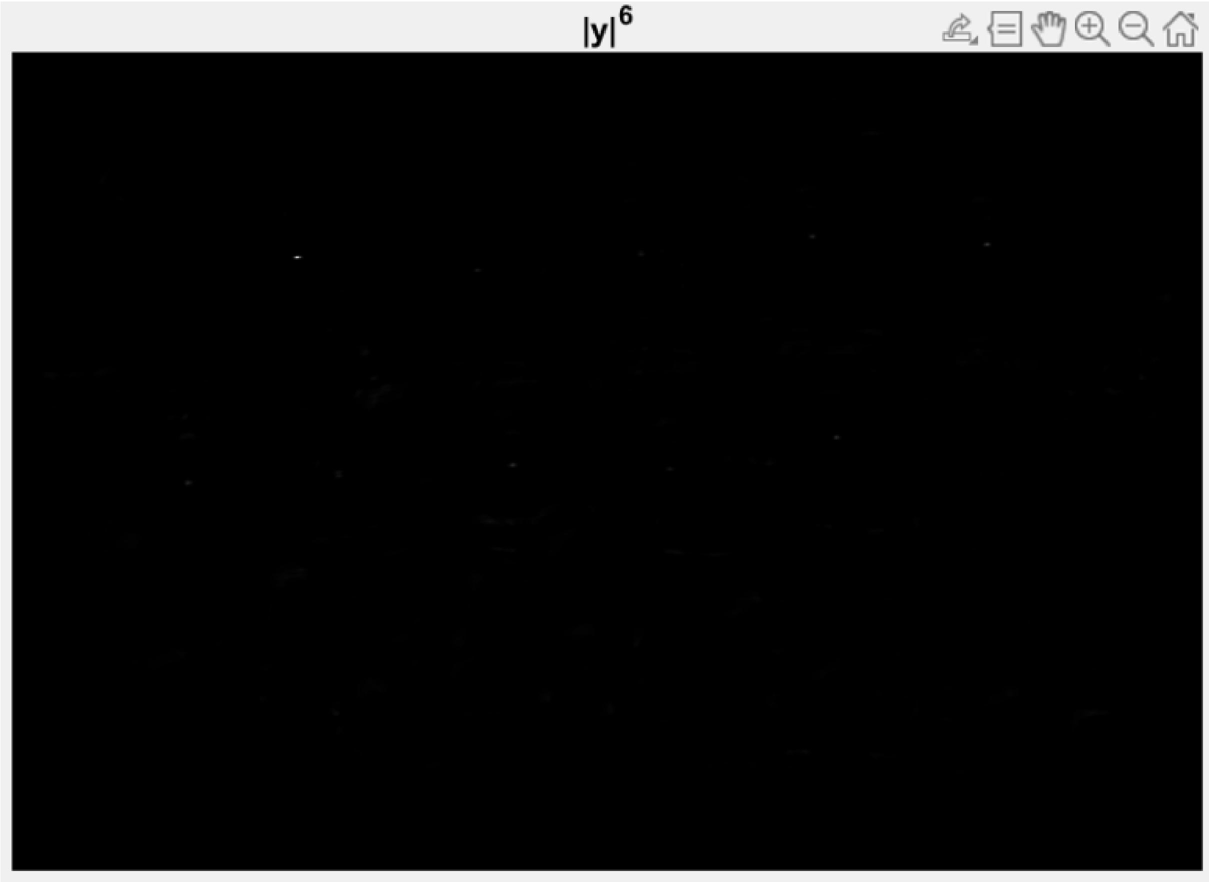

Figure 8: The image when |y[m,n]|^6

Atahan İyiekici
21801985
EEE321-001
Lab Report 4

Matlab Codes

```
%% PART 1 %%
image = ReadMyImage("Part5.bmp");
figure; DisplayMyImage(image); title("part-1");
%% PART 3 %%
x = [-1 2 6; 0 2 4; 5 6 3];
h = [3 5; -1 4];
y = DSLSI2D(h,x);
%% PART 4 %%
x = ReadMyImage("Part4.bmp"); figure;
subplot(2,2,1);
DisplayMyImage(x); title("Given Image");
D8 = rem(21801985,8);
Mh = 25 + D8;
Nh = 25 + D8;
B =[0.8 0.5 0.2];
for i = 1:3
 B_ind= B(i);
 for k = 1: Mh
 for l = 1:Nh
 h(k,l) = sinc((B_ind) * (k - 1 - (Mh -1) / 2)) * sinc((B_ind) * (l - 1 - (Nh -1)
/ 2));
 end
 end
 y = DSLSI2D(h,x);
 subplot(2,2,i+1);
 DisplayMyImage(y);
 title(strcat("Resulting Image when B = ",num2str(B_ind)));
end

%% PART 5 %%
x = ReadMyImage('Part5.bmp');
figure; DisplayMyImage(x); title('Original IMage');
h1 = [1/2, -1/2];
y_1 = DSLSI2D(h1,x);
figure; subplot(3, 2, 1); DisplayMyImage(y_1); title('Y1');
s1 = y_1.^2;
subplot(3, 2, 2); DisplayMyImage(s1); title('S1 = Y1^2');
h2 = [1/2; -1/2];
y_2 = DSLSI2D(h2,x);
subplot(3, 2, 3);DisplayMyImage(y_2); title('Y2');
s2 = y_2.^2;
subplot(3, 2, 4); DisplayMyImage(s2); title('S2 = Y2^2');
h3 = 0.25*h1 + 0.25*h2;
y_3 = DSLSI2D(h3,x);
subplot(3, 2, 5); DisplayMyImage(y_3); title('Y3');
s3 = y_3.^2;
subplot(3, 2, 6); DisplayMyImage(s3); title('S3 = Y3^2');
```

Atahan İyiekici
21801985
EEE321-001
Lab Report 4

```
%% PART 6 %%
x = ReadMyImage('Part6x.bmp');
figure; DisplayMyImage(x); title('Given IMage');
h_1 = ReadMyImage('Part6h.bmp');
figure; DisplayMyImage(h_1); title('h1 = Impulse Response ');
y = DSLSI2D(h_1,x);
figure; DisplayMyImage(abs(y)); title('|y|');
y1 = (abs(y)).^4;
figure; DisplayMyImage(y1); title('|y|^4');
y2 = (abs(y)).^6;
figure; DisplayMyImage(y2); title('|y|^6');
%% FUNCTIONS %%
function [y] = DSLSI2D(h, x)
Mh = size(h, 1); Nh = size(h, 2);
Mx = size(x, 1); Nx = size(x, 2);
y = zeros(Mh + Mx - 1, Nh + Nx - 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
function [x] = ReadMyImage(string)
x=double((rgb2gray(imread(string))));
x=x-min(min(x));
x=x/max(max(x));
x=x-0.5;
end

function []=DisplayMyImage(Image)
Image=Image-min(min(Image));
figure;
imshow(uint8(255*Image/max(max(abs(Image)))));
end
```