Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

EEE 321 Lab 4

Part 1

```
A=ReadMyImage("Part5.bmp");
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000
2	-0.5000	-0.0397	-0.0476	-0.0675	-0.0556	-0.0556	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794
3	-0.5000	-0.0357	-0.0476	-0.0675	-0.0675	-0.0556	-0.0556	-0.0833	-0.0794	-0.0556	-0.0794	-0.0794	-0.0794	-0.0794	-0.1151
4	-0.5000	-0.0635	-0.0397	-0.0675	-0.0556	-0.0794	-0.0833	-0.0833	-0.1032	-0.0833	-0.0833	-0.0794	-0.0476	-0.0794	-0.1151
5	-0.5000	-0.0357	-0.0119	-0.0317	-0.0595	-0.0556	-0.0833	-0.0833	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.1151
6	-0.5000	-0.0397	-0.0397	-0.0873	-0.0873	-0.0794	-0.0794	-0.0794	-0.0794	-0.0833	-0.1151	-0.0913	-0.0556	-0.0556	-0.0794
7	-0.5000	-0.0754	-0.0754	-0.0913	-0.0833	-0.0833	-0.0794	-0.0833	-0.0833	-0.0913	-0.0794	-0.0556	-0.0794	-0.0794	-0.0794
8	-0.5000	-0.0635	-0.0437	-0.0913	-0.0833	-0.0556	-0.0794	-0.0833	-0.0794	-0.0833	-0.0913	-0.0556	-0.0556	-0.0794	-0.0794
9	-0.5000	-0.1032	-0.0794	-0.0556	-0.0833	-0.0794	-0.0833	-0.0833	-0.1032	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794	-0.0794
10	-0.5000	-0.0635	-0.0437	-0.0913	-0.0833	-0.0595	-0.0476	-0.0833	-0.0913	-0.0833	-0.0833	-0.0833	-0.0833	-0.0556	-0.0556
11	-0.5000	-0.0754	-0.0754	-0.0675	-0.0913	-0.0833	-0.0833	-0.1032	-0.1151	-0.1032	-0.0833	-0.1032	-0.1032	-0.0833	-0.0833
12	-0.5000	-0.1111	-0.0516	-0.0675	-0.0913	-0.0556	-0.0833	-0.0833	-0.0833	-0.0913	-0.0833	-0.0794	-0.0913	-0.0794	-0.1151
13	-0.5000	-0.1111	-0.0754	-0.0873	-0.0913	-0.1151	-0.1190	-0.1190	-0.1151	-0.0794	-0.0794	-0.0913	-0.0913	-0.0913	-0.0833
14	-0.5000	-0.1111	-0.0754	-0.0873	-0.1270	-0.0913	-0.0913	-0.1151	-0.1270	-0.1151	-0.0913	-0.0913	-0.1151	-0.0913	-0.0913
15	-0.5000	-0.0754	-0.0754	-0.0873	-0.0913	-0.1190	-0.1190	-0.1190	-0.1190	-0.0833	-0.1151	-0.1151	-0.0913	-0.0913	-0.1151
16	-0.5000	-0.1071	-0.0754	-0.1032	-0.1270	-0.0913	-0.0833	-0.0952	-0.0913	-0.0595	-0.0913	-0.0913	-0.0714	-0.0714	-0.0595
17	-0.5000	-0.1111	-0.0833	-0.1032	-0.0913	-0.0714	-0.0595	-0.0714	-0.0595	-0.0595	-0.0397	-0.0397	-0.0397	-0.0397	-0.0397
18	-0.5000	-0.1111	-0.0635	-0.0556	-0.0476	-0.0159	-0.0159	-0.0159	-0.0278	-0.0198	-0.0278	-0.0278	0.0040	0.0040	0.0397
19	-0.5000	-0.0476	0.0238	0.0278	0.0595	0.0556	0.0516	0.0278	0.0516	0.0516	0.0476	0.0675	0.0833	0.0794	0.0992
20	-0.5000	0.0079	0.0833	0.0873	0.1071	0.1032	0.0992	0.0952	0.0952	0.0952	0.0913	0.0913	0.1230	0.1230	0.1270
21	-0.5000	0.0714	0.1548	0.1468	0.1627	0.1587	0.1548	0.1508	0.1270	0.1468	0.1468	0.1468	0.1468	0.1587	0.1587
22	-0.5000	0.1071	0.2103	0.1786	0.1706	0.1667	0.1667	0.1508	0.1667	0.1468	0.1468	0.1468	0.1786	0.1786	0.1786
23	-0.5000	0.1071	0.2103	0.1786	0.1706	0.1548	0.1508	0.1825	0.1825	0.1786	0.1786	0.1786	0.1786	0.1786	0.1786
24	-0.5000	0.1429	0.2103	0.1786	0.1944	0.1865	0.1627	0.1825	0.1786	0.1667	0.1786	0.1786	0.1786	0.1786	0.1786
25	-0.5000	0.1111	0.1865	0.1746	0.1706	0.1627	0.1825	0.1786	0.1667	0.1786	0.1786	0.1786	0.1667	0.1667	0.1667
26	-0.5000	0.1151	0.2063	0.1746	0.1548	0.1865	0.1825	0.1825	0.1984	0.1984	0.1786	0.1984	0.1984	0.1786	0.1786
27	-0.5000	0.1151	0.2183	0.1746	0.1905	0.1865	0.1825	0.1825	0.2024	0.1786	0.1786	0.1984	0.1984	0.2024	0.1984
28	-0.5000	0.1468	0.2183	0.1746	0.1865	0.1825	0.2024	0.2024	0.2024	0.1905	0.1984	0.1984	0.1984	0.1905	0.1905
29	-0.5000	0.1389	0.2183	0.1746	0.1865	0.1865	0.2024	0.2024	0.1984	0.1984	0.2024	0.2024	0.1984	0.1905	0.2024
30	-0.5000	0.1389	0.2183	0.1746	0.1905	0.2063	0.2024	0.2024	0.2024	0.1984	0.2024	0.2024	0.2024	0.2262	0.2024
31	-0.5000	0.1151	0.2183	0.1944	0.1905	0.2063	0.2063	0.2024	0.2024	0.2143	0.2262	0.2262	0.2262	0.2262	0.2302
32	-0.5000	0.1389	0.2183	0.2063	0.1905	0.1865	0.2063	0.2024	0.2262	0.2262	0.2262	0.2262	0.2262	0.2302	0.2302
33	-0.5000	0.1389	0.2183	0.2103	0.2222	0.2063	0.2063	0.2302	0.2262	0.2143	0.2302	0.2302	0.2302	0.2302	0.2302
34	-0.5000	0.1389	0.2381	0.1984	0.1905	0.2103	0.2063	0.2063	0.2302	0.2302	0.2302	0.2302	0.2341	0.2341	0.2341
35	-0.5000	0.1389	0.2222	0.1984	0.2143	0.2103	0.2103	0.2302	0.2302	0.2302	0.2302	0.2222	0.2341	0.2341	0.2341
36	-0.5000	0.1389	0.2421	0.1984	0.2143	0.2103	0.2341	0.2341	0.2302	0.2302	0.2302	0.2341	0.2341	0.2540	0.2381
37	-0.5000	0.1508	0.2421	0.1984	0.2143	0.2103	0.2341	0.2341	0.2341	0.2341	0.2222	0.2222	0.2341	0.2540	0.2341
38	-0.5000	0.1508	0.2421	0.1984	0.2143	0.2103	0.2103	0.2341	0.2341	0.2341	0.2540	0.2579	0.2579	0.2579	0.2579

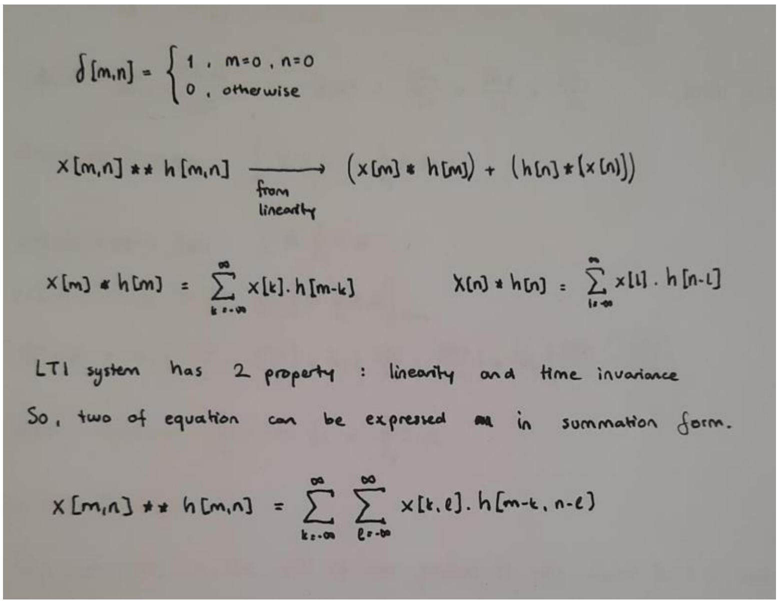
Figure 1 – 512 x 512 matrix of the image

This is the matrix representation of the image signals.

Figure 2 – Result of DisplayMyImage()

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

Part 2



$$\delta[m,n] = \begin{cases} 1, & m=0,\ n=0 \\ 0, & \text{otherwise} \end{cases}$$

$$x[m,n] ** h[m,n] \xrightarrow[\text{linearity}]{\text{from}} (x[m] * h[m]) + (h[n] * (x[n]))$$

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

LTI system has 2 property : linearity and time invariance

So, two of equation can be expressed in summation form.

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

Figure 3 – 2D convolution

By using the linearity and time-invariance property, we proved that 2D convolution is the sum of the 1D convolutions.

Part 3

Photos are 512 x 512 in this case. So, the boundaries of the x[m,n] M_x and N_x are 512.To convolve the whole photo, impulse functions h[m,n] should have the same boundaries. At the example, we are asked to take x[] as 3 x 3 and h[] is 2 x 2.

Matlab Code

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

    [Mx,Nx] = size(x);
    [Mh,Nh] = size(h);
    Y = zeros(Mx+Mh-1,Nx+Nh-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
```

And the main code part is given at the below.

Matlab Code

```
Mx=3;
Nx=3;
Mh=2;
Nh=2;

X=zeros(Mx,Nx);
X(1:Mx,1:Nx)=[1 0 2;
             -1 3 1;
             -2 4 0];

H=zeros(Mh,Nh);

H(1:Mh,1:Nh)=[1 -1;
              0 2];

y = DSLSI2D(H,X)
```

```
y =

     1    -1     2    -2
    -1     6    -2     3
    -2     4     2     2
     0    -4     8     0
```

Figure 4 – Result of the convolution x and h

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

Part 4

At this part, we are asked to clean the noise in an image. Pictures consist of signals at various frequencies. If we remove the high frequency parts, the transitions will become softer and photos seem softer. So, the noise is cleaned. At the figures below, I used imshow() so the images are darker than what it should be. When I tried with given function, picture is clearer. The clear picture shown at Figure 6-9.

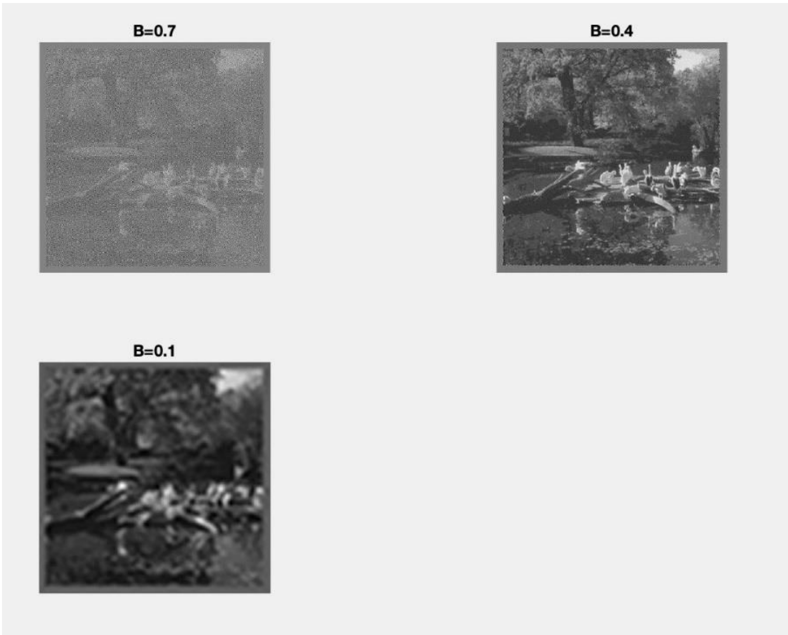


Figure 5 – Cleaned images

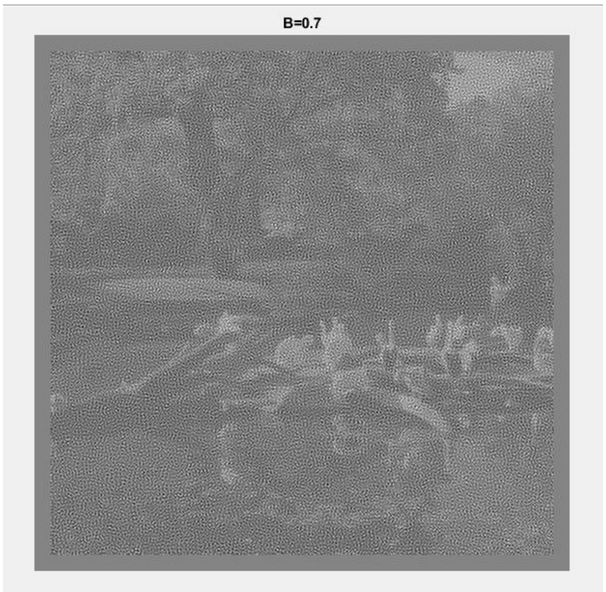


Figure 6 – B = 0.7

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

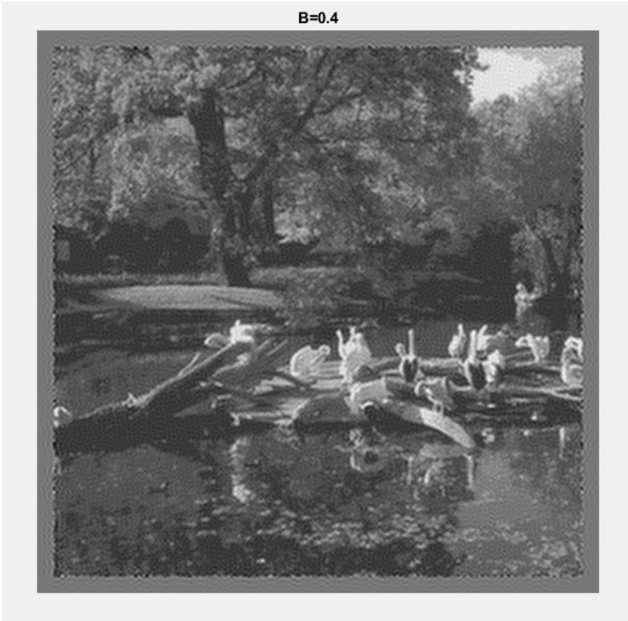


Figure 7 – B = 0.4

Figure 8 – B =0.1

Figure 9 – Original Photo

Matlab Code

```
ID = 22002075;
D7 = rem(ID,7);
Mh = 30 + D7;
Nh = 30 + D7;

H=zeros(Mh,Nh);
K=zeros(Mh,Nh);
L=zeros(Mh,Nh);

X = ReadMyImage('Part4.bmp');
DisplayMyImage(X);

B = 0.7;
C = 0.4;
D = 0.1;

for m = 0:Mh-1
    for n = 0:Nh-1
    H0(m+1,n+1) = sinc(B*(m-((Mh-1)/2))) * sinc(B*(n-((Mh-1)/2)));
    H1(m+1,n+1) = sinc(C*(m-((Mh-1)/2))) * sinc(C*(n-((Mh-1)/2)));
    H2(m+1,n+1) = sinc(D*(m-((Mh-1)/2))) * sinc(D*(n-((Mh-1)/2)));
    end
end

result0=DSLSI2D(H0,X);
result1=DSLSI2D(H1,X);
result2=DSLSI2D(H2,X);
```

```
hold on

figure;
imshow(result0)
title('B=0.7')

figure;
imshow(result1)
title('B=0.4')

figure;
imshow(result2)
title('B=0.1')
```

Part 5

At this part, we are asked to find the edges. In a photo, an edge can be found in the fast change at the signal. Because of the color signal changes dramatically, these points are required high frequency components. So, if we use high pass filter, only high frequency parts remain and shades can be seemed better. With changing h function, we highlighted different edges of the same picture. Then we sum them up and result was clear edge detected picture.

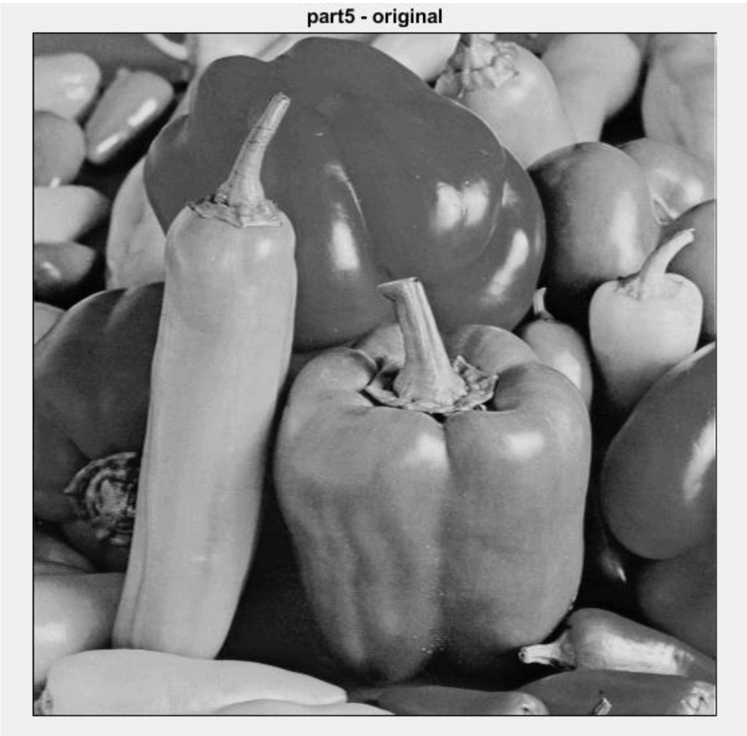


Figure 10 – Original image

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

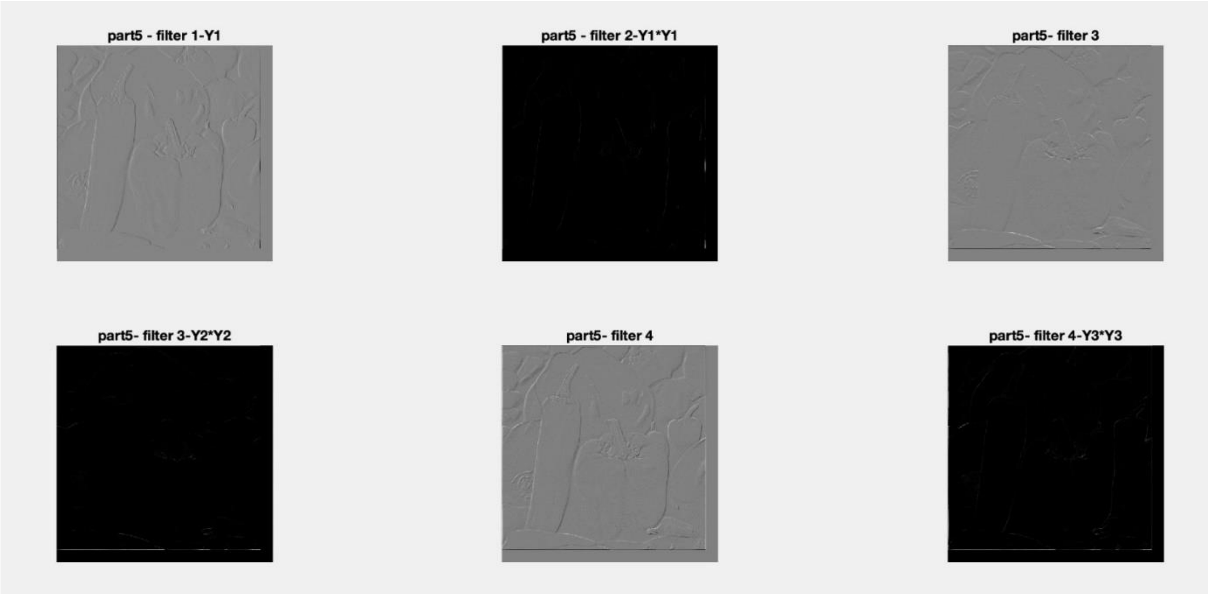


Figure 11 – Edge detected images

```
ID = 22002075;
D7 = rem(ID,7);

Mh = 30 + D7;
Nh = 30 + D7;

H=zeros(Mh,Nh);
K=zeros(Mh,Nh);
L=zeros(Mh,Nh);

B=0.7;
C=0.4;
D=0.1;

for m=0:Mh-1
    for n=0:Nh-1
    H0(m+1,n+1) = sinc(B*(m-((Mh-1)/2))) * sinc(B*(n-((Mh-1)/2)));
    H1(m+1,n+1) = sinc(C*(m-((Mh-1)/2))) * sinc(C*(n-((Mh-1)/2)));
    H2(m+1,n+1) = sinc(D*(m-((Mh-1)/2))) * sinc(D*(n-((Mh-1)/2)));
    end
end

result0=DSLSI2D(H,X);
result1=DSLSI2D(K,X);
result2=DSLSI2D(L,X);
```

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

```
X = ReadMyImage('Part5.bmp');

DisplayMyImage(X)

Horiz = zeros(Mh,Nh);
Horiz(1,1)=0.5;
Horiz (1,2)=-0.5;
Y1=DSLSI2D(Horiz,X1);
Y4=Y1.^2;

figure

subplot(2, 3, 1);
imagesc(Y1)
title('Y1')

subplot(2, 3, 2);
imagesc(Y4)
title('Y1^2')

Verti = zeros(Mh,Nh);
Verti (1,1)=0.5;
Verti (2,1)=-0.5;
Y2=DSLSI2D(Verti,X1);
Y5=Y2.^2;

subplot(2, 3, 3);
imagesc(Y2)
title(‘Y2')

subplot(2, 3, 4);
imagesc(Y5)
title('Y2^2')

H3=0.5*H1 + 0.5*H2;
Y3=DSLSI2D(H3,X1);
Y6=Y3.^2;

subplot(2, 3, 5);
imagesc(Y3)
title('Y3')

subplot(2, 3, 6);
imagesc(Y3)
title('Y3^2')
```

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

Part 6

At part 6, we are asked to find the face in the national soccer player photos. In the part 3, 2D convolution function was written. With this function, two photos are convolved each other and the same pattern was tried to find at the photo.

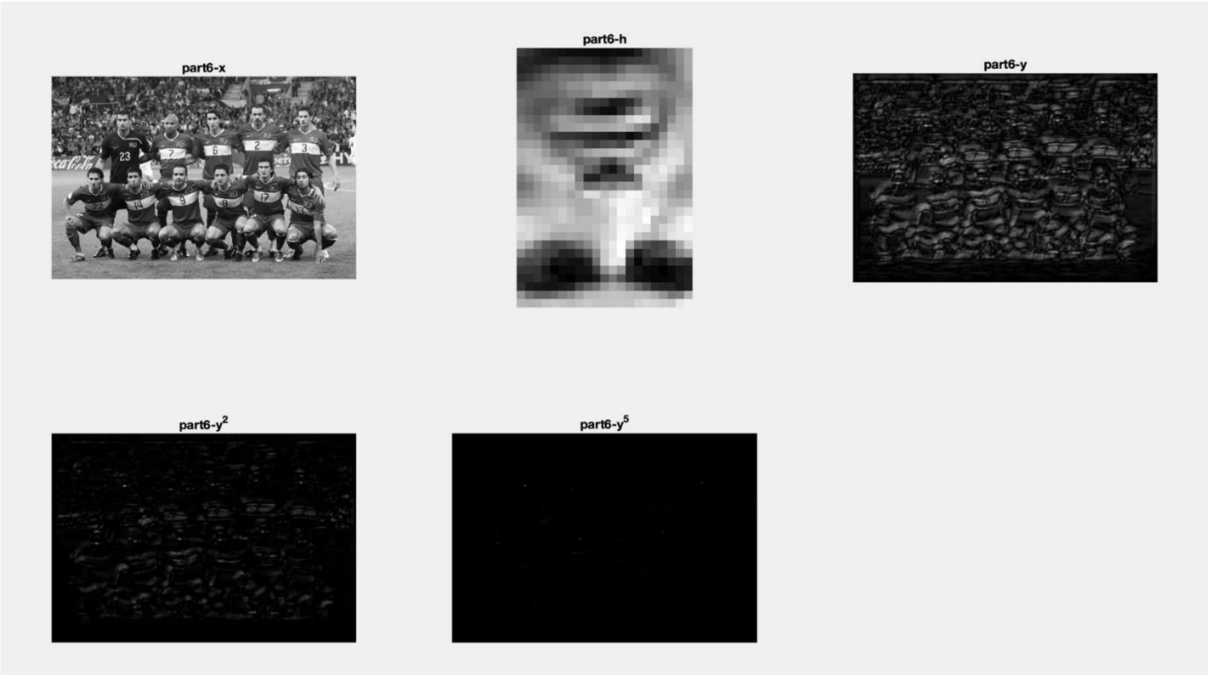


Figure 12 – 2 original images and convolution results

All pixels have a value between -1 and 1. So, when we take higher order of each term if it is not 1, it tends to go zero and getting darker. So only matched point remains white.

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

Figure 13 - Original photo.

Figure 14 – Searched image

Satılmış Furkan Kahraman
22002075

Figure 15- Convolved image

Figure 16 – Squared of the convolved picture

Satılmış Furkan Kahraman
22002075
10/11/2023
Report 4

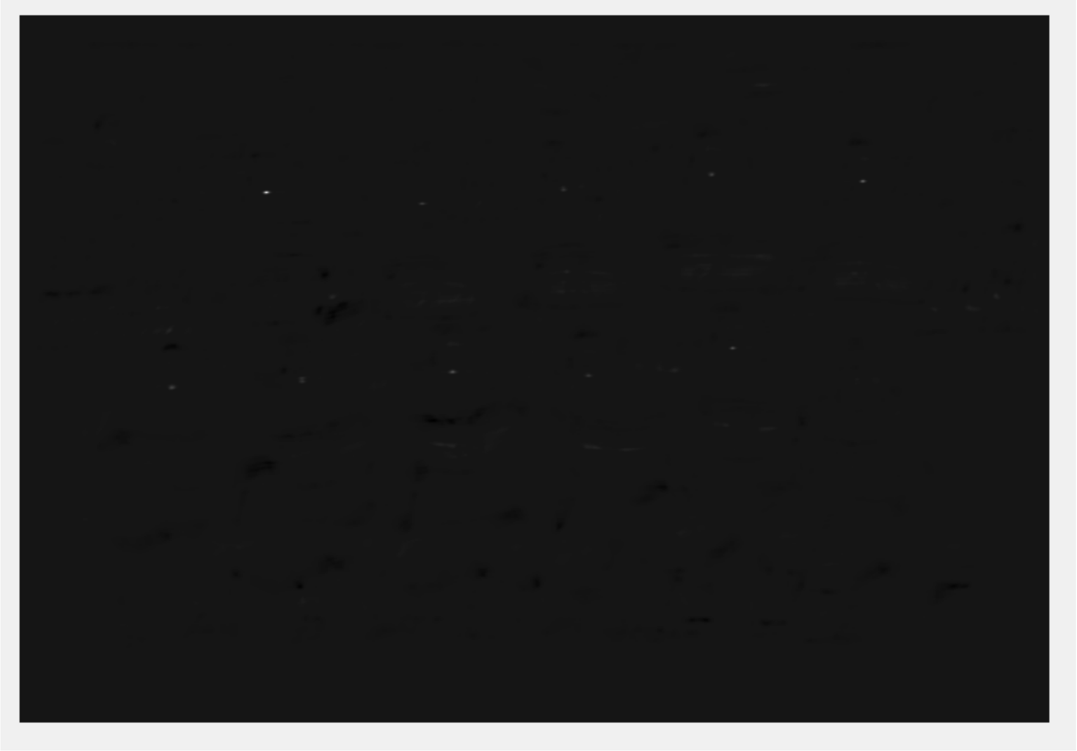


Figure 17 – 5th order degree of the image

```
ID = 22002075;
D7 = rem(ID,7);
Mh = 30 + D7;
Nh = 30 + D7;

X=ReadMyImage('Part4.bmp');
% DisplayMyImage(X);

H0 = zeros(Mh,Nh);
H1 = zeros(Mh,Nh);
H2 = zeros(Mh,Nh);

B = 0.7;
C = 0.4;
D = 0.1;

for m=0:Mh-1
    for n=0:Nh-1
    H0(m+1,n+1) = sinc(B*(m-((Mh-1)/2))) * sinc(B*(n-((Mh-1)/2)));
    H1(m+1,n+1) = sinc(C*(m-((Mh-1)/2))) * sinc(C*(n-((Mh-1)/2)));
    H2(m+1,n+1) = sinc(D*(m-((Mh-1)/2))) * sinc(D*(n-((Mh-1)/2)));
    end
end

result0 = DSLSI2D(H0,X);
result1 = DSLSI2D(H1,X);
result2 = DSLSI2D(H2,X);
```

```
figure;

X6=ReadMyImage('Part6x.bmp');
subplot(2, 3, 1);
imagesc(abs(X6))
title('X')

H6=ReadMyImage('Part6h.bmp');
subplot(2, 3, 2);
imagesc(abs(H6))
title('H')

Y6=DSLSI2D(H6,X6);
subplot(2, 3, 3);
imagesc(abs(Y6))
title('Y')

Y7=Y6.^2;
subplot(2, 3, 4);
imagesc(abs(Y7))
title('Y^2')

Y8=Y6.^5;
subplot(2, 3, 5);
imagesc(abs(Y8))
title('Y^5')

DisplayMyImage(X6)

DisplayMyImage(H6)

DisplayMyImage(Y6)

DisplayMyImage(Y7)

DisplayMyImage(Y8)
```