EEE - 321: Signals and Systems

Lab Assignment 4

Yiğit Narter

22102718

Section 02

Part 1

As expected, the image was displayed successfully using the required commands.

Part 2

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

Now write an input signal $x[m,n]$, which is a two dimensional signal as:

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

We can think of $x[k,\ell]$ as the coefficient of the impulse $\delta[m-k, n-\ell]$, which is shifted by k units in m-axis, and shifted by ℓ units in n-axis.

Since $\delta[m-k,n-\ell]=1$ for only $m=k$, $n=\ell$ (and 0 otherwise), by adding for all $k,\ell \in \mathbb{Z}$, we can obtain $x[m,n]$, as a sum of $x[k,\ell] \cdot \delta[m-k, n-\ell]$'s.

Now, since system is time-variant, we know that the impulse response (shifted) to $\delta[m-k, n-\ell]$ should be $h[m-k, n-\ell]$.

Using the linearity of the system, the input-output relation can be obtained as:

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

or making variable change as $a=m-k$, $b=n-\ell$

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

which is denoted by:

$$\boxed{y[m,n] = x[m,n] ** h[m,n]}$$

Part 3

Opening up eqn. 5:

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

For $y[m,n]$ to be non-zero, we need $h[k,l]$ and $x[m-k,n-l]$ to be non-zero:

- For $h[k,l] \neq 0$, we need $\boxed{0 \leq k \leq M_h-1, \quad 0 \leq l \leq N_h-1}$ ①
- For $x[m-k,n-l] \neq 0$, we need $0 \leq m-k \leq M_x-1 \Rightarrow k \leq m \leq M_x-1+k$
  $0 \leq n-l \leq N_x-1 \Rightarrow l \leq n \leq N_x-1+l$ ②

Combining 1 and 2:

- We know $0 \leq k \leq M_h-1$ and $k \leq m \leq M_x-1+k$.
  Based on interval of k:
  $\Rightarrow 0 \leq m \leq M_x-1+M_h-1 \Rightarrow \boxed{0 \leq m \leq M_x+M_h-2}$
  similarly: $0 \leq l \leq N_h-1$ and $l \leq n \leq N_x-1+l$:
  $\Rightarrow 0 \leq n \leq N_x-1+N_h-1 \Rightarrow \boxed{0 \leq n \leq N_x+N_h-2}$

Hence $y[m,n]$ is non-zero for: $\boxed{\begin{array}{l} 0 \leq m \leq M_x+M_h-2 \\ 0 \leq n \leq N_x+N_h-2 \end{array}}$ ⊛

As given in manual, $0 \leq m \leq M_y-1$ and $0 \leq n \leq N_y-1$. Ⓐ
Equating the limits (of ⊛ and Ⓐ), we obtain:

$$\boxed{M_y = M_x+M_h-1}, \quad \boxed{N_y = N_x+N_h-1}$$

and $y[m,n] \neq 0$ for $0 \leq m \leq M_y-1$, $0 \leq n \leq N_y-1$.

The function DSLSI2D:

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

Mh = size(h,1);
Nh = size(h,2);
Mx = size(x,1);
Nx = size(x,2);

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

To test the code, the inputs x and h are given according to the lab manual, and the obtained output also conforms to that in the lab manual:

```
x = [1 0 2; -1 3 1 ; -2 4 0];
h = [1 -1; 0 2];

y = DSLSI2D(h, x)
```

```
>> part3

y =

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

Figure 1: The output matrix y is as expected

Part 4

```
x = ReadMyImage('Part4.bmp');
subplot(2,2,1);
DisplayMyImage(x);
title('Original image');

d7 = rem(22102718, 7);

Mh = 30 + d7;
Nh = 30 + d7;
B = 0.7;

h = zeros(Mh, Nh);

for m=0:Mh-1
    for n=0:Nh-1
        k = B * (m - (Mh-1)/2);
        l = B * (n - (Nh-1)/2);
        h(m+1, n+1) = sinc(k) * sinc(l);
    end
end

y=DSLSI2D(h, x);
subplot(2,2,2);
DisplayMyImage(y);
title('B = 0.7');
```

The code above generates h[m,n], then processes the image using the previously defined function. The same process is repeated for B=0.4 and B=0.1, and the images are all given under a subplot. The command subplot() is used such that they will be presented in 2 rows and 2 columns in the same figure, since there are 4 images in total. The resulting images are given below.

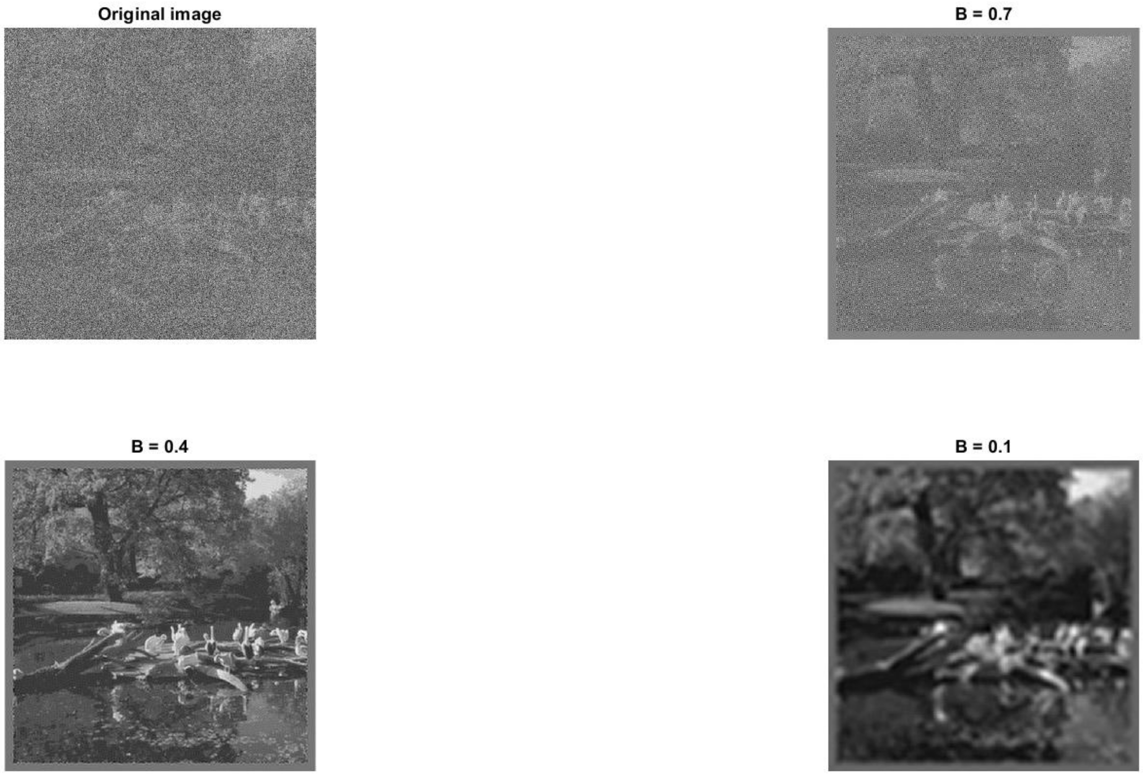

Figure 2: The images for B=0.7, 0.4 and 0.1

The original photo x is very noisy, while the processed images (y) can be observed more clearly compared to x. For B=0.7, the noise is reduced but still exists. For B=0.1, the image is very blurry, even though the noise is canceled. It can be seen that the clearest image is obtained for B=0.4, where there is no noise or blur. Hence, 0.4 seems to be the most appropriate value to choose. This is because as the bandwidth is used to cancel high frequency noises, if B is 0.7, some noises can still pass since the bandwidth is wider. If B is 0.1, on the other hand, along with the noise, some details are also eliminated, resulting in a blurry image. This is caused by the narrow band-width this time, with the smallest value for B.

Part 5

The code for part 5:

```
x = ReadMyImage('Part5.bmp');
DisplayMyImage(x);
title('Original image');

h1 = [0.5 -0.5];
y1 = DSLSI2D(h1,x);
s1 = y1.^2;
DisplayMyImage(s1);
title('s1[m,n]');

h2 = [0.5 ; -0.5];
y2 = DSLSI2D(h2,x);
s2 = y2.^2;
DisplayMyImage(s2);
title('s2[m,n]');

h3 = 0.5*h1 + 0.5*h2;
y3 = DSLSI2D(h3,x);
s3 = y3.^2;
DisplayMyImage(s3);
title('s3[m,n]');
```

The original image, also the resulting images for $y_1[m,n]$, $y_2[m,n]$, $s_1[m,n]$ and $s_2[m,n]$ are presented in Figures 3-7.

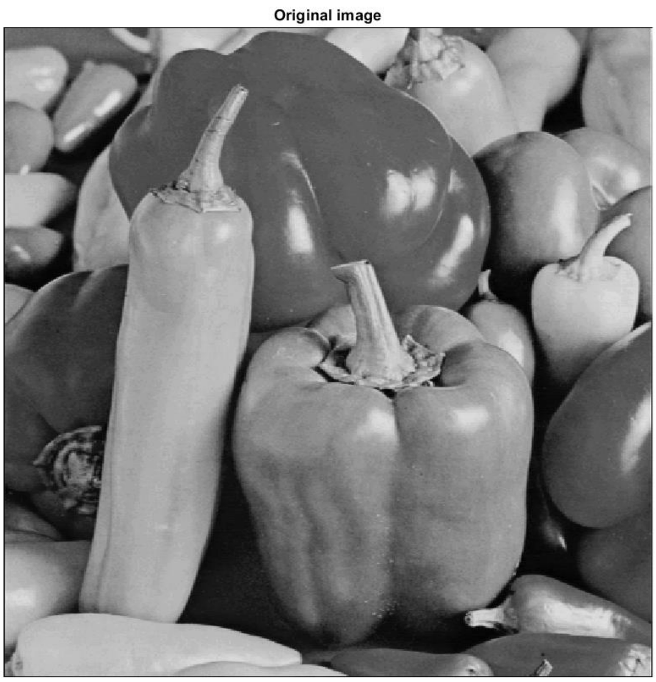

Figure 3: Original image of part 5

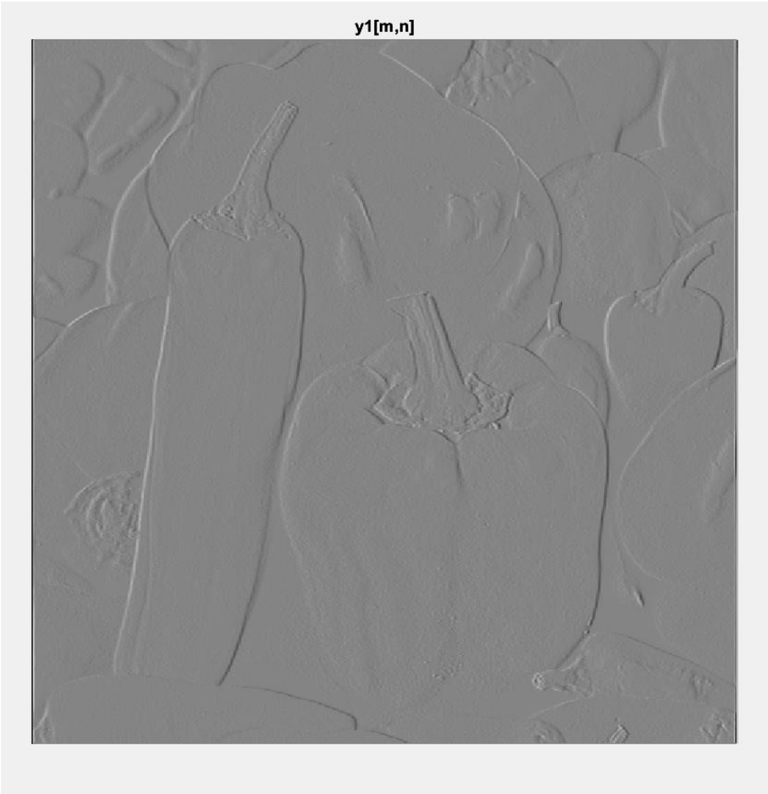

Figure 4: $y_1[m,n]$ is displayed

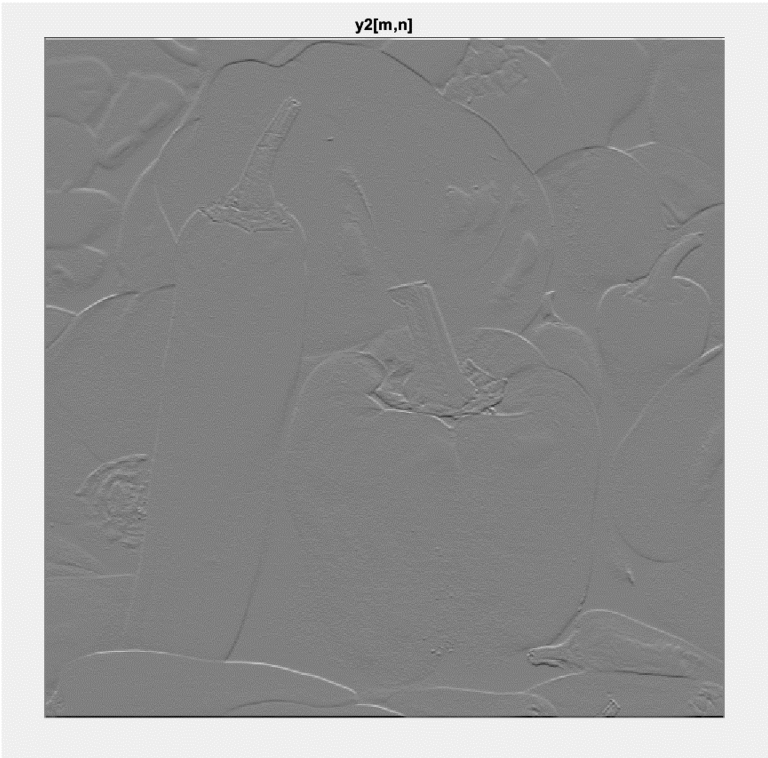

Figure 5: $y_2[m,n]$ is displayed

Figure 6: $s_1[m,n]$ is displayed

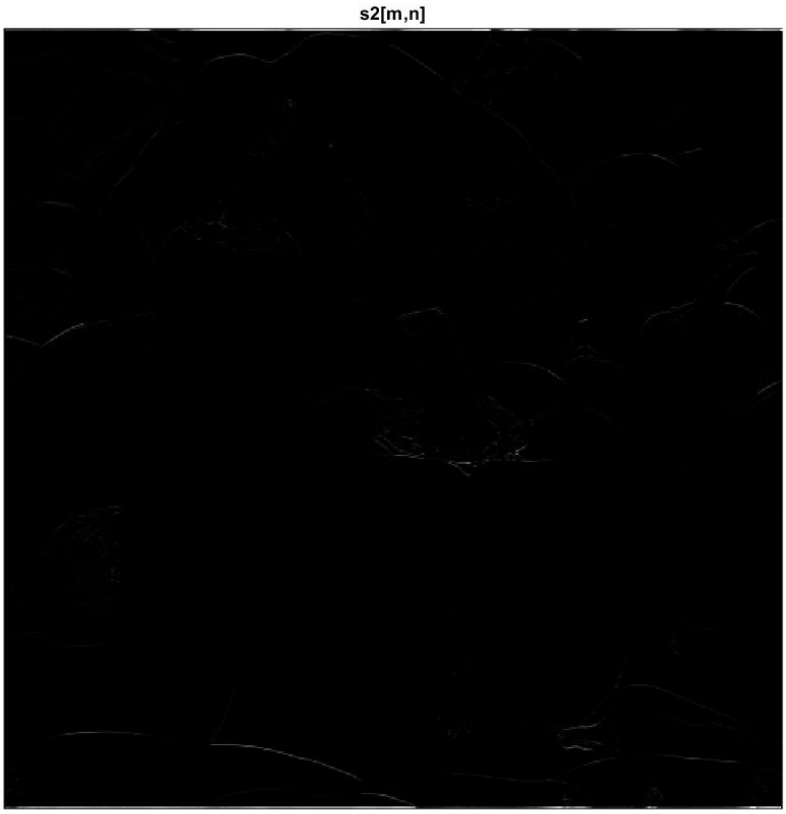

Figure 7: $s_2[m,n]$ is displayed

In s_1, the vertical edges of the photos are emphasized and are visible, while in s_2 the horizontal edges are emphasized. The difference is that since the image is processed with different h's, the output emphasizes differently oriented edges of the image. Taking the square of y, denoted as s, also puts more emphasis on the edges, making them more distinct compared to y. After that, the final image is processed, given in Figure 8.

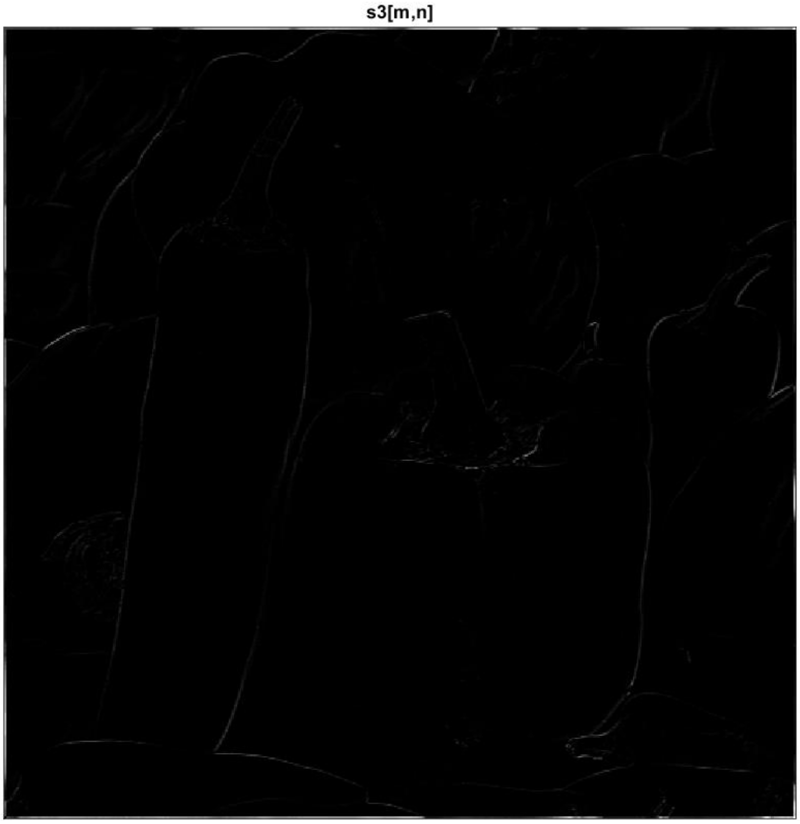

Figure 8: $s_3[m,n]$ is displayed

In s_3, all the edges are visible and distinguishable. This image is like a combination of s_1 and s_2, the first emphasizing the vertical edges and the other emphasizing the horizontal edges. In s_1, the horizontal edges are not visible while for s_2 the vertical edges are not distinguishable. But in s_3, all of the edges are now emphasized and therefore visible. The image for y_3 is also given in Figure 9.

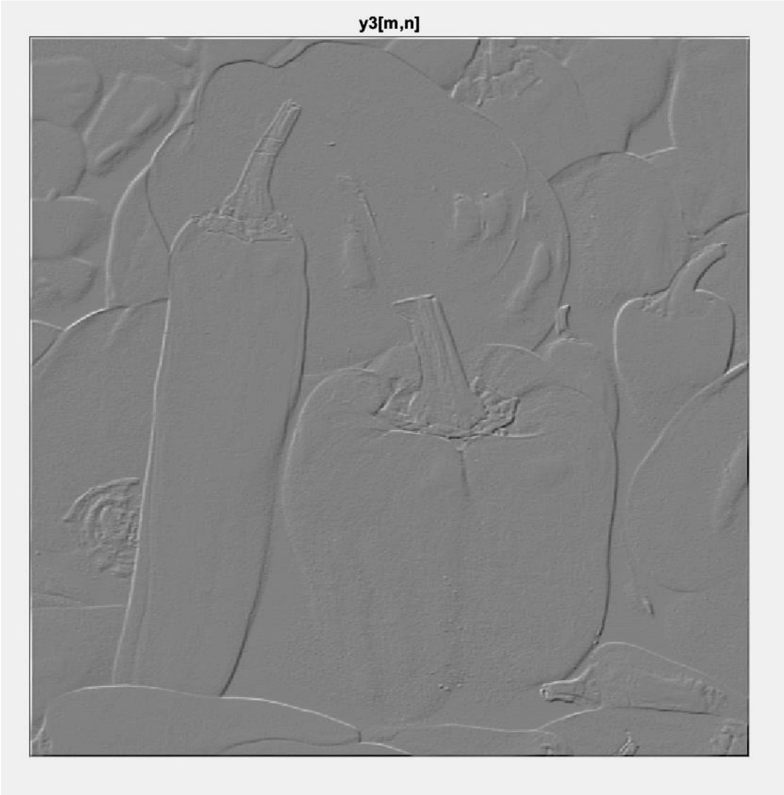

Figure 9: $y_3[m,n]$ is displayed

Part 6

The code for part 6:

```
x = ReadMyImage('Part6x.bmp');
DisplayMyImage(x);
title('Original image');

h = ReadMyImage('Part6h.bmp');
DisplayMyImage(h);
title('Impulse response');

y = DSLSI2D(h,x);

DisplayMyImage(abs(y));
title('|y[m,n]|');

DisplayMyImage(abs(y).^3);
title('|y[m,n]|^3');

DisplayMyImage(abs(y).^5);
title('|y[m,n]|^5');
```

Figure 10: Original image (x)

Figure 11: Impulse response (h)

The resulting image, which is the absolute value of y, is given in Figure 12. The bright points occur on the faces of the players, but also in other places like their uniform (around their chest) or the background. Hence it is possible to find bright points where there is no face, meaning that I don't always see a face at the location that there is a bright point. Still the brightest points are on the faces, though not the only ones.

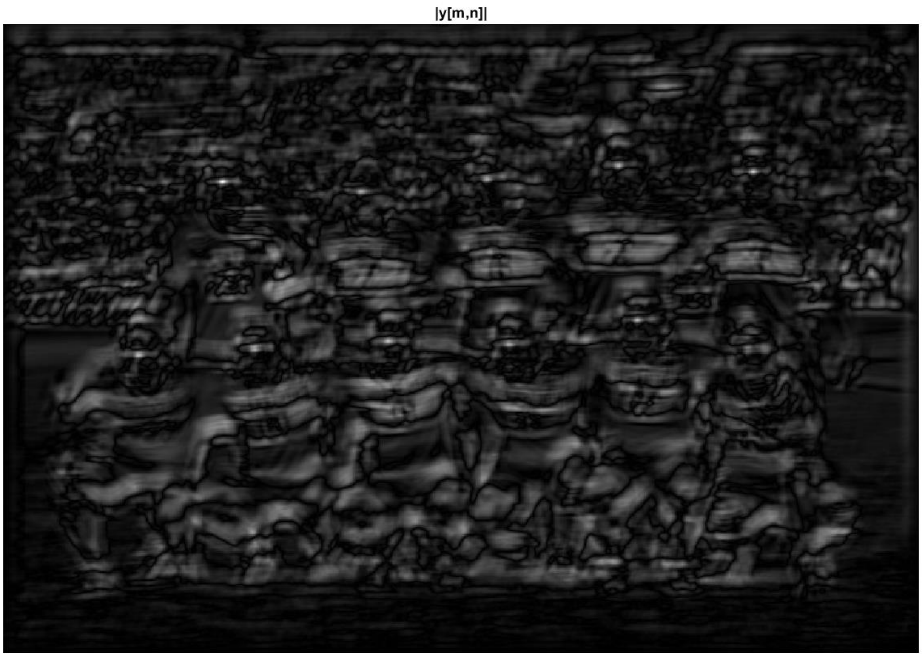

Figure 12: Output image |y[m,n]|

To make bright points more visible, the powers of |y[m,n]| are taken, namely its 3rd and 5th powers. The resulting images are shown in Figures 13-14.

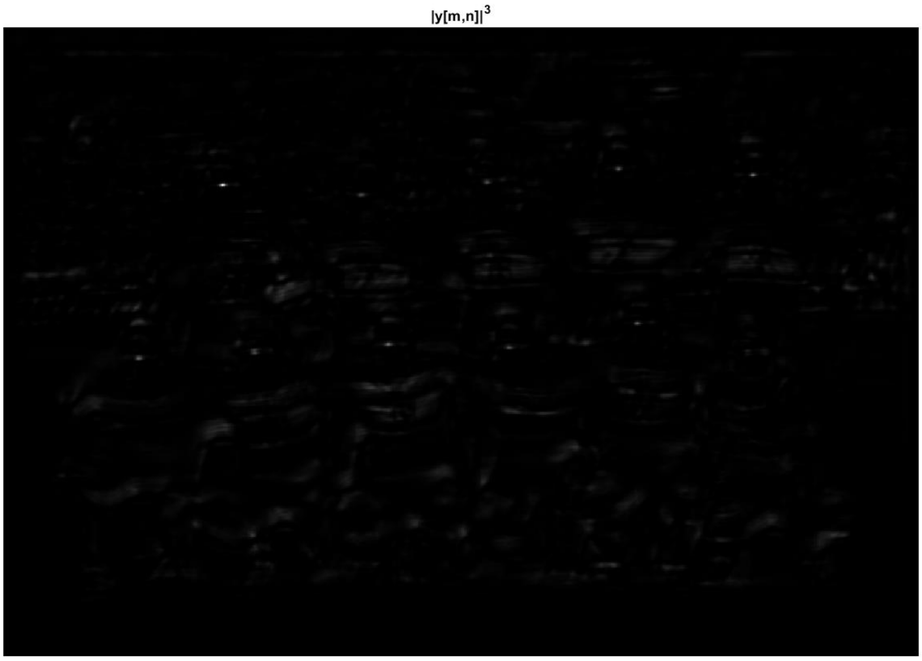

Figure 13: Output image $|y[m,n]|^3$

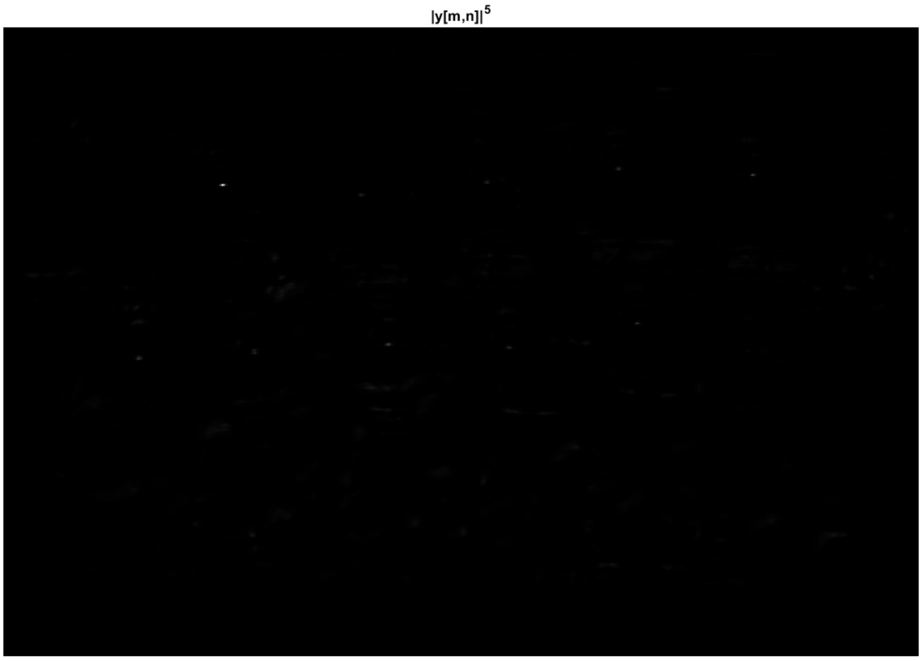

Figure 14: Output image $|y[m,n]|^5$

Taking the powers helps distinguish the brighter points. As the power is increased, the bright points that are not on the faces diminish and disappear, while the ones on the faces become more distinct. Taking the 3rd power, there were still minor bright points on the players' uniforms; however, when the 5th power is taken the bright points are only on their faces and nowhere else. Hence, taking the 5th power was sufficient to detect the faces without any confusion. Hence, this method was successful in terms of detecting the faces, since the bright points indicated the faces' locations on the photo.