- **Explanation: Black color font**
- **Headlines: Blue color Font**
- **Notes: Red color font**

Task 1

A-

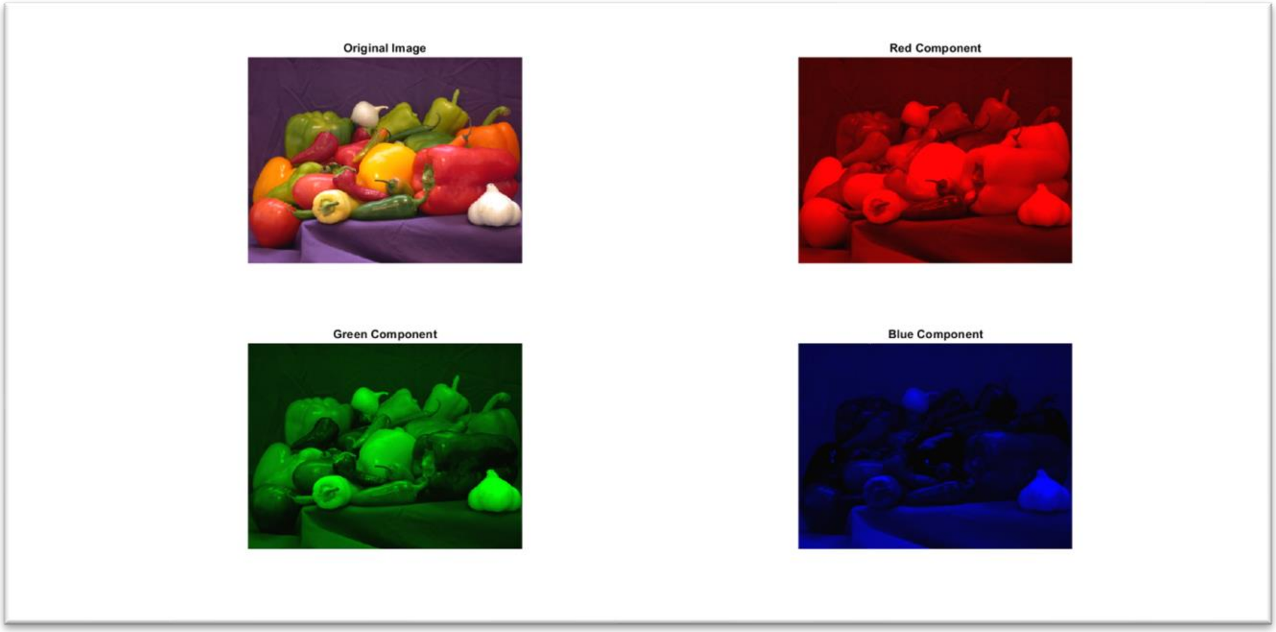


Image 0

I read the image then separate the 3 channels.

By making 3 new variable R,G,B in each variable I delete the two other channels .

B1- Edge detection kernel:

1	1	1
0	0	0
-1	-1	-1

Note:

-when dose the edge happen?

- **it happens when there Is different color -> big different in intensity value -> sharp transition -> high frequency -> sharp edge**

- **I choose this filter because it takes the difference in the intensity around the center pixel.**
- **When I do this, I try to know where the higher transition (higher frequency) in intensity value.**
- **If the difference is small or =0 -> that mean, there is low frequency -> smooth transition -> no edge detected -> black color in output image**
- **If the difference is Big -> that mean , there is High frequency -> Sharp transition -> edge detected -> white color in output image (the color of the edge in the output image is white)**

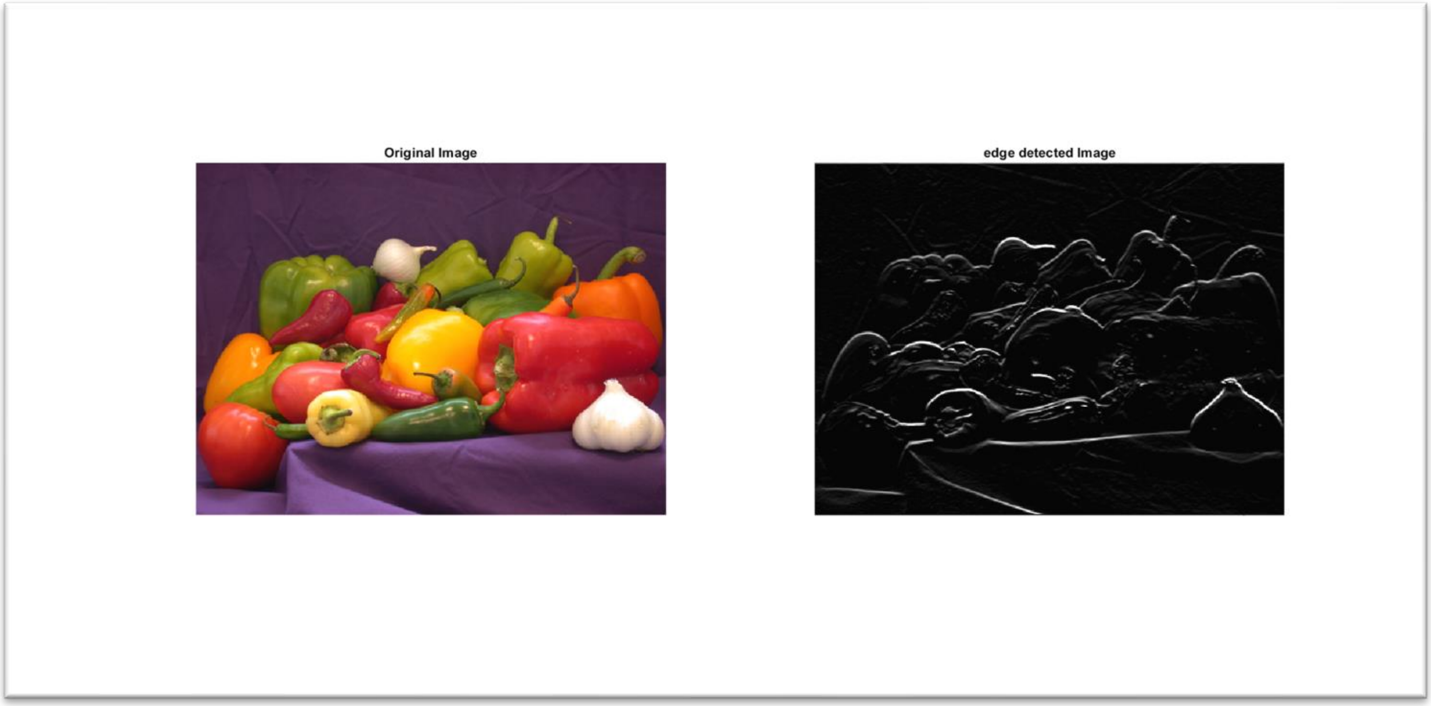


Image 1

B2- Image sharpening kernel:

1	1	1
0	1	0
-1	-1	-1

Note: it’s nearly the same filter as the edge detection filter except the center pixel =1

- **the concept is to add the original photo to the edge detected photo to make photo clearer and enhance the image details and edges as it’s required**
- **the centered value in the filer it’s role to maintain the original value in original photo while the other value in the filter edge detected the original photo and then the**

result will be the addition of the original photo and edge detected photo.

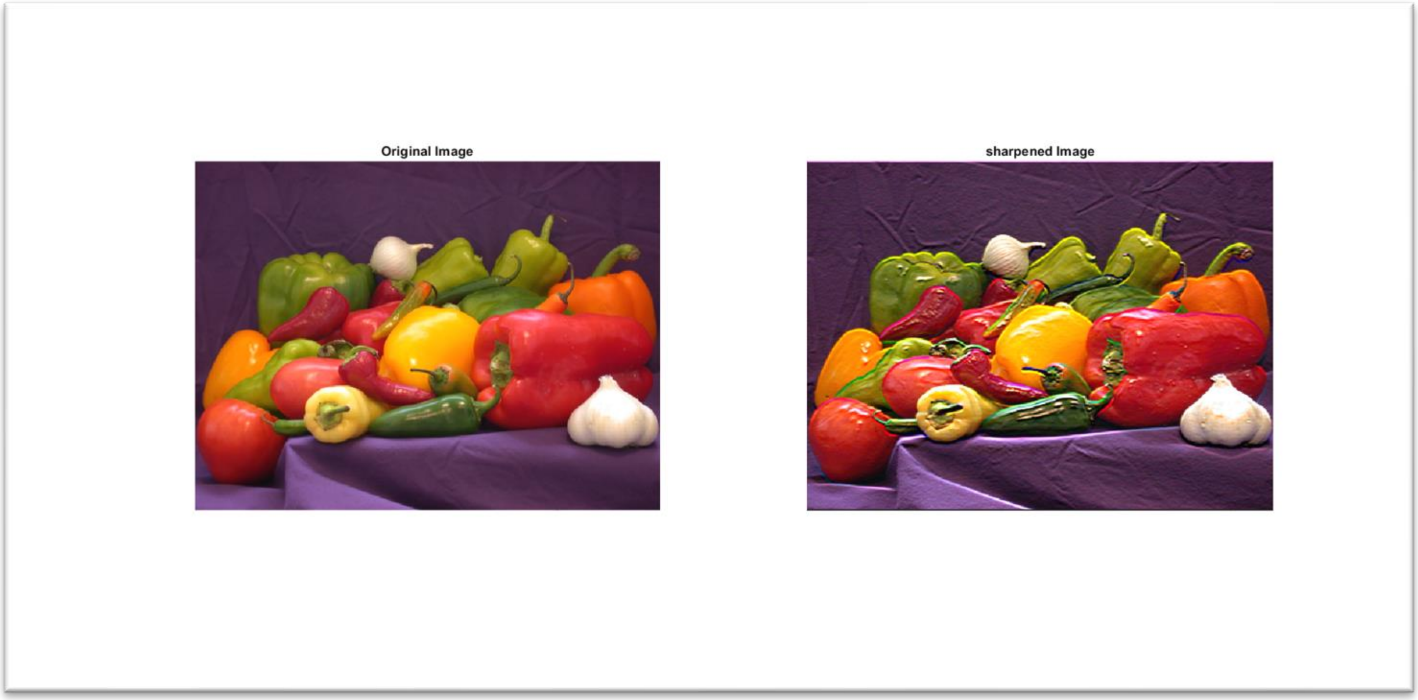


Image 2

B3- blur kernel:

1/9 *

1	1	1
1	1	1
1	1	1

I choose this filter because it takes an average of the intensity around the center pixel.

When I do this, I try to reduce the higher transition (higher frequency) between the value of the pixel's intensity, so I try to hide the details (make the photo blurred)

Note: here I choose the dimension of the filter 3*3 but the effect may be not strong so I will increase the size of the filter to be 5*5 and take the average and by doing that I hide more details of the photo (more blurred)

1/25 *

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

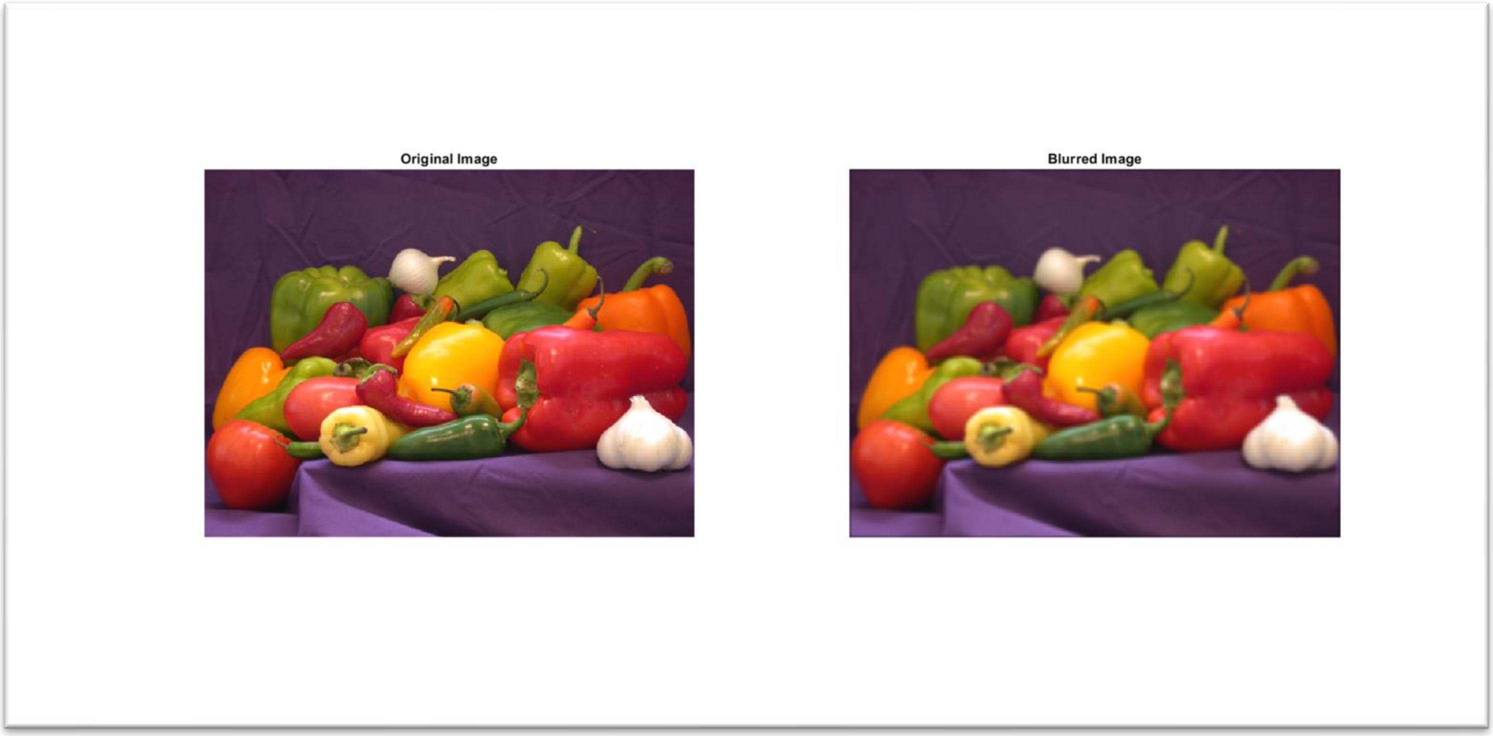


Image 3

B-4 Motion blurring:

1/7 *

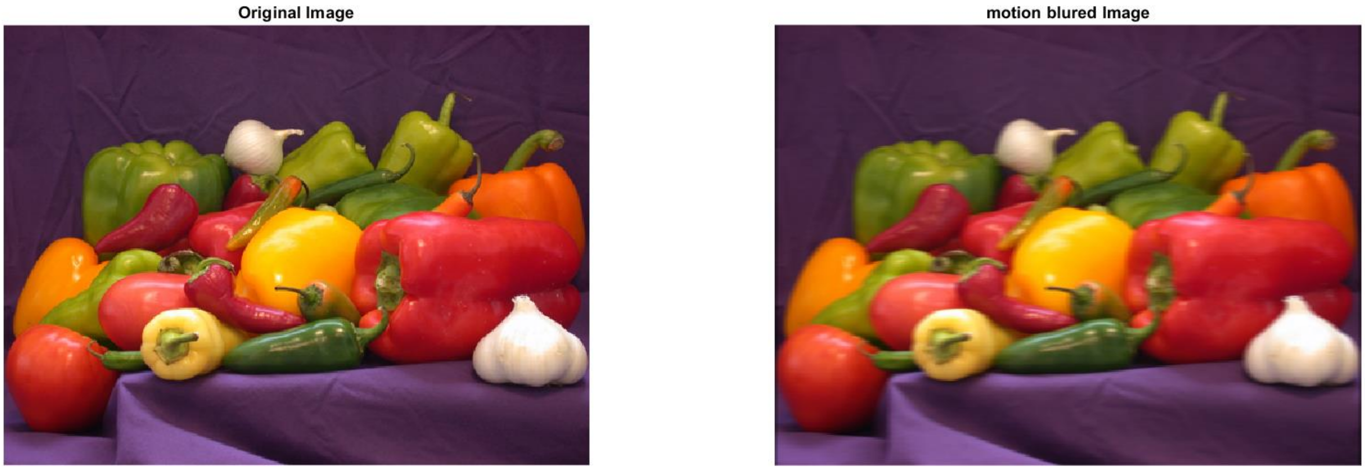
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	1	1	1	1	1	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

- **It’s look like the blurring (average) but only on one row and taking the average so the center pixel take the average only in the same raw while the blur in previous filter takes the average in all directions**
- **Note : if we want to make vertical motion blur we will also take the average but the one will be only on one column**

1/7 *

0	0	0	0	1	0	0
0	0	0	0	1	0	0
0	0	0	0	1	0	0
0	0	0	0	1	0	0
0	0	0	0	1	0	0
0	0	0	0	1	0	0
0	0	0	0	1	0	0

- Note : if we want to make the filter stronger we can increase the dimension of the filter and talking the average but on other hand when we increase the size of the filter it makes it harder to restore the details of the image in the next task


Original Image

motion blured Image

Image 4

C- Restor original image:

Steps:

- **I will transform the blurred image into frequency domain using FFT.**
- **I will transform the kernel into frequency domain using FFT.**
- **I will divide the FFT of blurred image on FFT of the kernel.**
- **I will transform the result back to space domain to get the original image.**

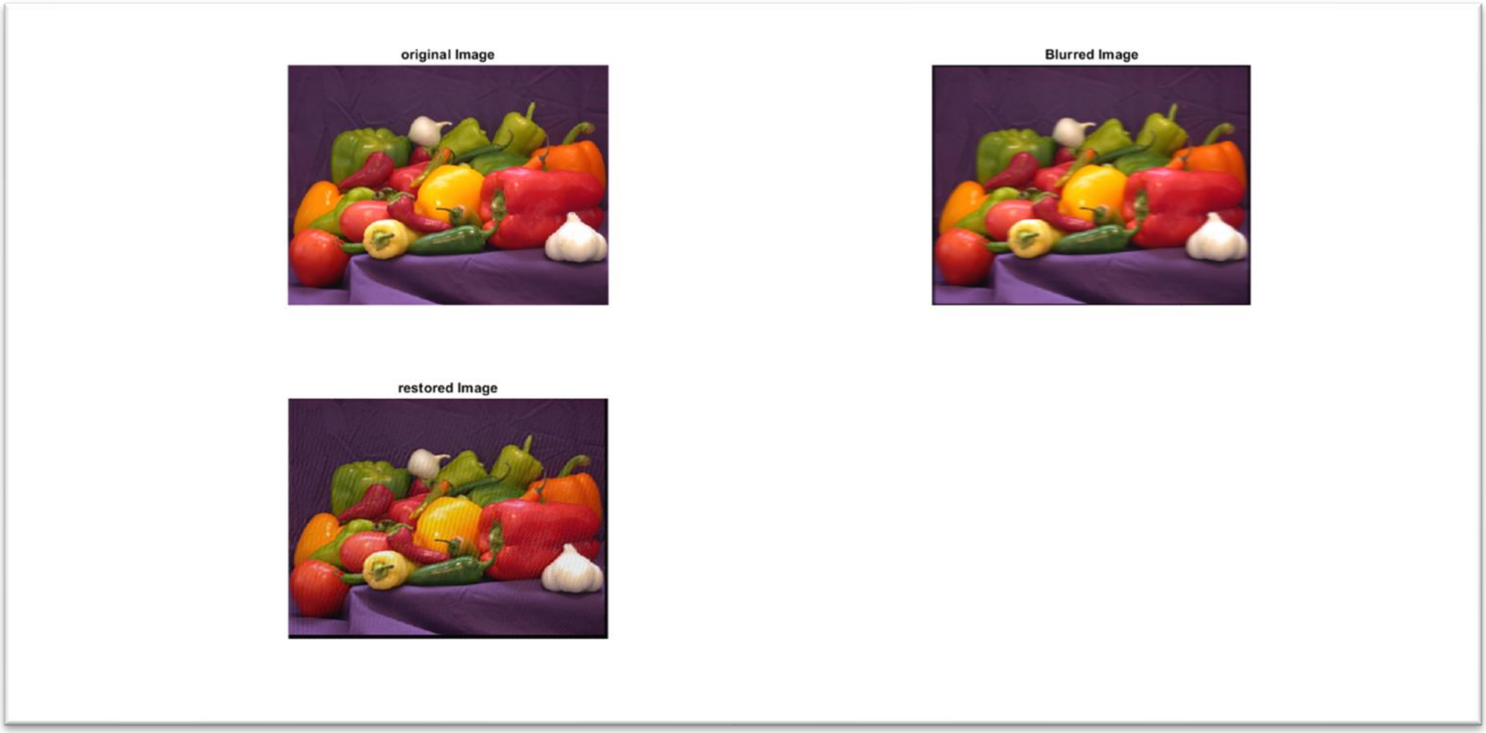


Image 5