Görüntü İşleme Dersi-MEK

Bilgisayar Mühendisliği, Yıldız Teknik Üniversitesi

Ödev 1

Bengi Yurdusever

19011014

Creating and Applying Gauss Filters

Filter sizes of 3x3, 5x5 and 7x7 and standard deviation(s) values of 1.0, 2.0 and 4.0 were sent to the function as parameters and the calculated filters were applied to the image.


```
Gauss filter for kernel size: 3 and sigma : 1.000000
0.075114 0.123841 0.075114
0.123841 0.204180 0.123841
0.075114 0.123841 0.075114
```



```
Gauss filter for kernel size: 3 and sigma : 2.000000
0.101868 0.115432 0.101868
0.115432 0.130801 0.115432
0.101868 0.115432 0.101868
```



```
Gauss filter for kernel size: 3 and sigma : 4.000000
0.108797 0.112250 0.108797
0.112250 0.115813 0.112250
0.108797 0.112250 0.108797
```



```
Gauss filter for kernel size: 5 and sigma : 1.000000
0.002969 0.013306 0.021938 0.013306 0.002969
0.013306 0.059634 0.098320 0.059634 0.013306
0.021938 0.098320 0.162103 0.098320 0.021938
0.013306 0.059634 0.098320 0.059634 0.013306
0.002969 0.013306 0.021938 0.013306 0.002969
```



```
Gauss filter for kernel size: 5 and sigma : 2.000000
0.023247 0.033824 0.038328 0.033824 0.023247
0.033824 0.049214 0.055766 0.049214 0.033824
0.038328 0.055766 0.063191 0.055766 0.038328
0.033824 0.049214 0.055766 0.049214 0.033824
0.023247 0.033824 0.038328 0.033824 0.023247
```
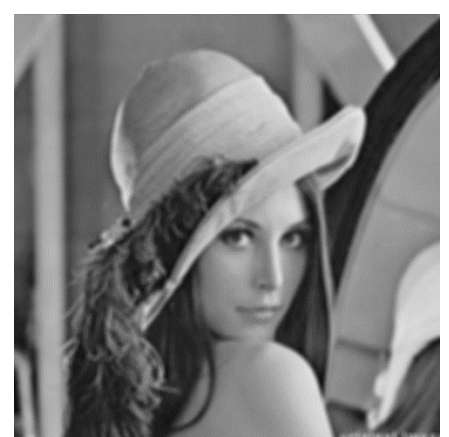


```
Gauss filter for kernel size: 5 and sigma : 4.000000
0.035204 0.038664 0.039891 0.038664 0.035204
0.038664 0.042464 0.043812 0.042464 0.038664
0.039891 0.043812 0.045203 0.043812 0.039891
0.038664 0.042464 0.043812 0.042464 0.038664
0.035204 0.038664 0.039891 0.038664 0.035204
```
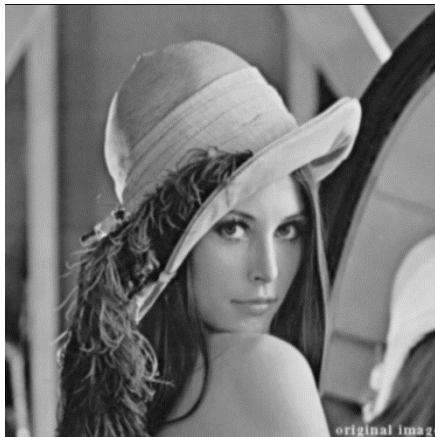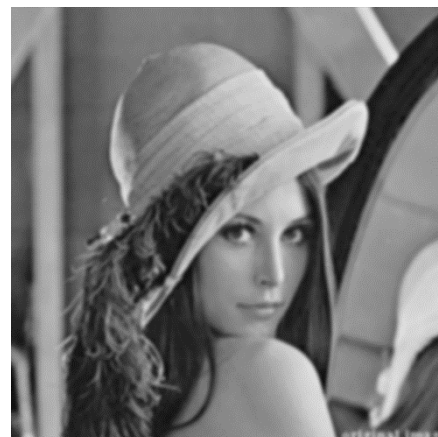


```
Gauss filter for kernel size: 7 and sigma : 1.000000
0.000020 0.000239 0.001073 0.001769 0.001073 0.000239 0.000020
0.000239 0.002917 0.013071 0.021551 0.013071 0.002917 0.000239
0.001073 0.013071 0.058582 0.096585 0.058582 0.013071 0.001073
0.001769 0.021551 0.096585 0.159241 0.096585 0.021551 0.001769
0.001073 0.013071 0.058582 0.096585 0.058582 0.013071 0.001073
0.000239 0.002917 0.013071 0.021551 0.013071 0.002917 0.000239
0.000020 0.000239 0.001073 0.001769 0.001073 0.000239 0.000020
```



```
Gauss filter for kernel size: 7 and sigma : 2.000000
0.004922 0.009196 0.013380 0.015162 0.013380 0.009196 0.004922
0.009196 0.017181 0.024998 0.028326 0.024998 0.017181 0.009196
0.013380 0.024998 0.036371 0.041214 0.036371 0.024998 0.013380
0.015162 0.028326 0.041214 0.046702 0.041214 0.028326 0.015162
0.013380 0.024998 0.036371 0.041214 0.036371 0.024998 0.013380
0.009196 0.017181 0.024998 0.028326 0.024998 0.017181 0.009196
0.004922 0.009196 0.013380 0.015162 0.013380 0.009196 0.004922
```
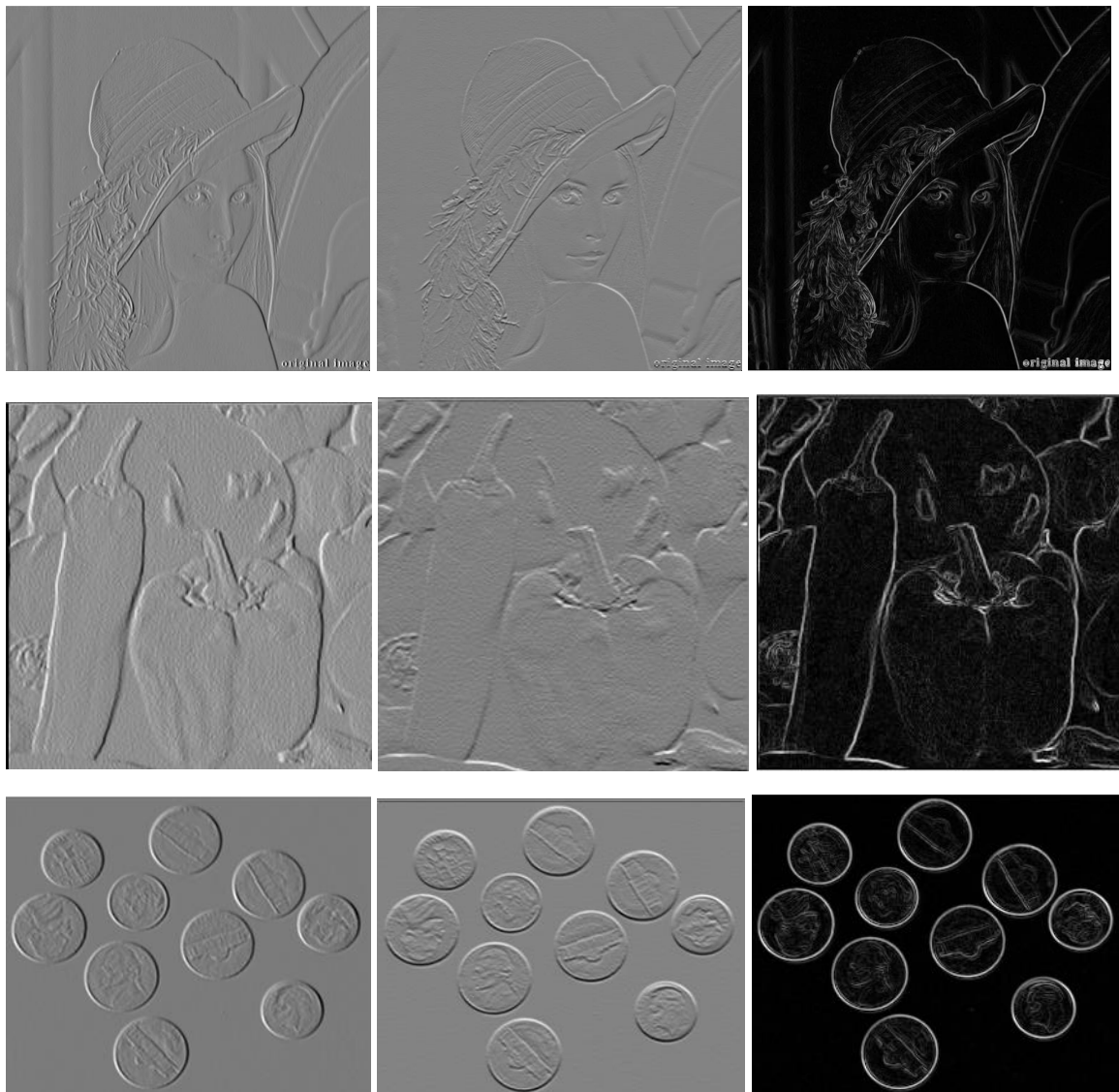


```
Gauss filter for kernel size: 7 and sigma : 4.000000
0.014760 0.017256 0.018952 0.019554 0.018952 0.017256 0.014760
0.017256 0.020175 0.022157 0.022861 0.022157 0.020175 0.017256
0.018952 0.022157 0.024335 0.025108 0.024335 0.022157 0.018952
0.019554 0.022861 0.025108 0.025909 0.025108 0.022861 0.019554
0.018952 0.022157 0.024335 0.025108 0.024335 0.022157 0.018952
0.017256 0.020175 0.022157 0.022861 0.022157 0.020175 0.017256
0.014760 0.017256 0.018952 0.019554 0.018952 0.017256 0.014760
```

At this stage, it has been observed that increasing the kernel size increases the blurring of the image. Larger kernel sizes will spread the blur over a wider region because each pixel is replaced by more pixels around it.

The same is true for Sigma. Since the σ value determines the extent of the blur effect around a pixel, the blurring increases as σ increases.

Applying Sobel Filter to Original Images

Edge detection was performed by applying a Sobel filter to the original images. The images of the result of the total edge detection process, which give the change in the x direction, y direction and in each direction, respectively, for each image are given below.

Three example .pgm formatted images are first blurred and then sobel filter is applied.

Kernel: 3x3 ve Sigma: 1

Kernel: 3x3 ve Sigma: 2

Kernel: 3x3 ve Sigma: 4

Kernel: 5x5 ve Sigma: 1

Kernel: 5x5 ve Sigma: 2

Kernel: 5x5 ve Sigma: 4

Kernel: 7x7 ve Sigma: 1

Kernel: 7x7 ve Sigma: 2

Kernel: 7x7 ve Sigma: 4

Kernel: 3x3 ve Sigma: 1
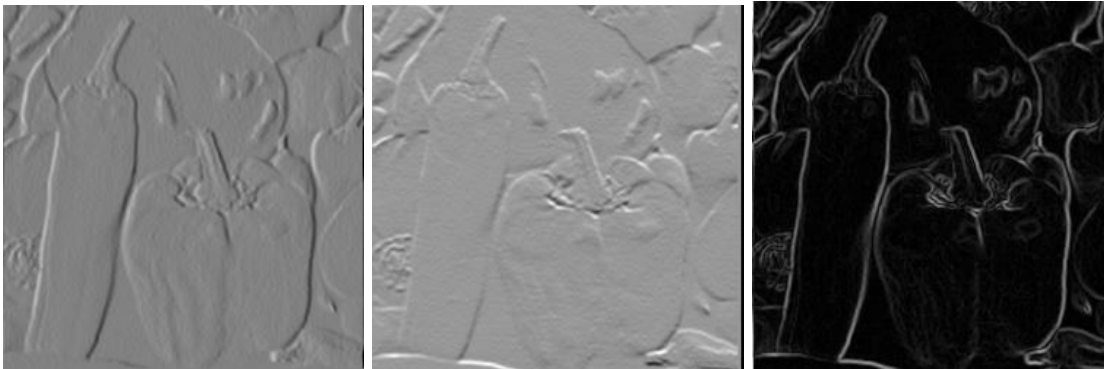
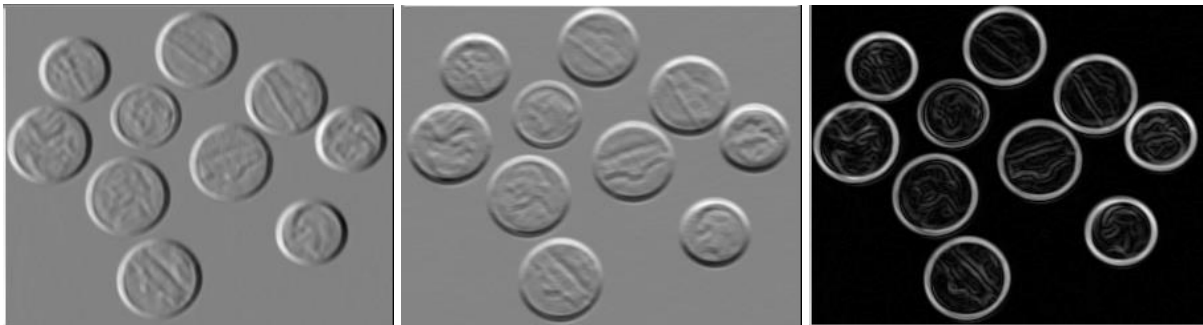Kernel: 3x3 ve Sigma: 2
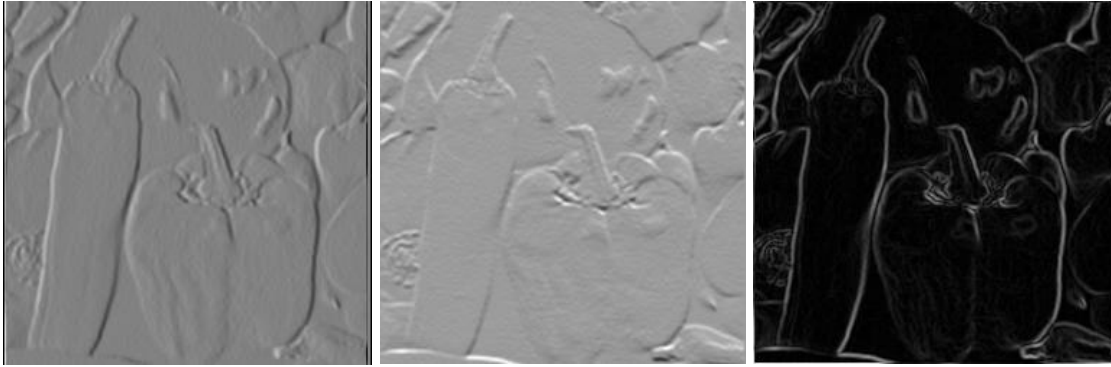
Kernel: 3x3 ve Sigma: 4
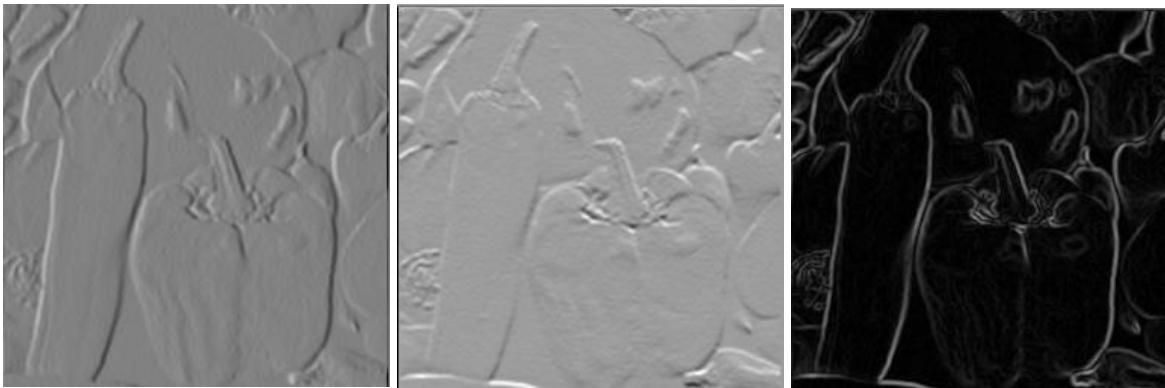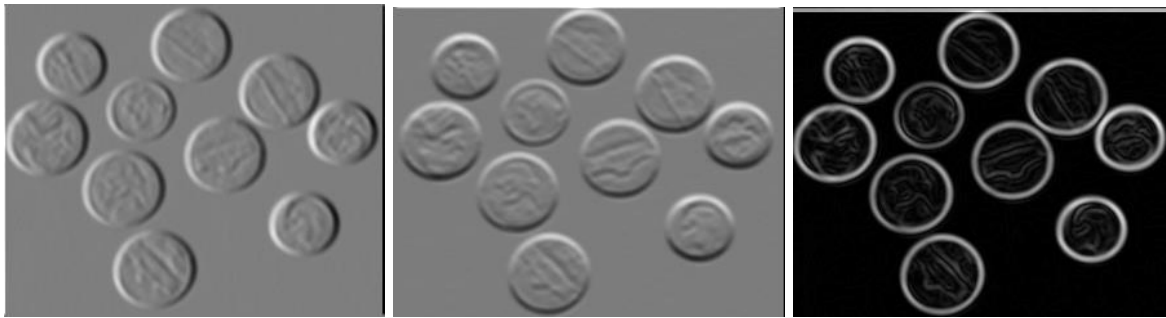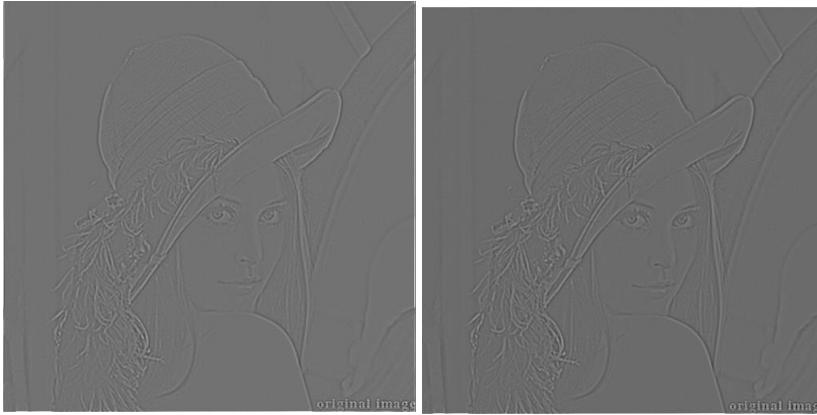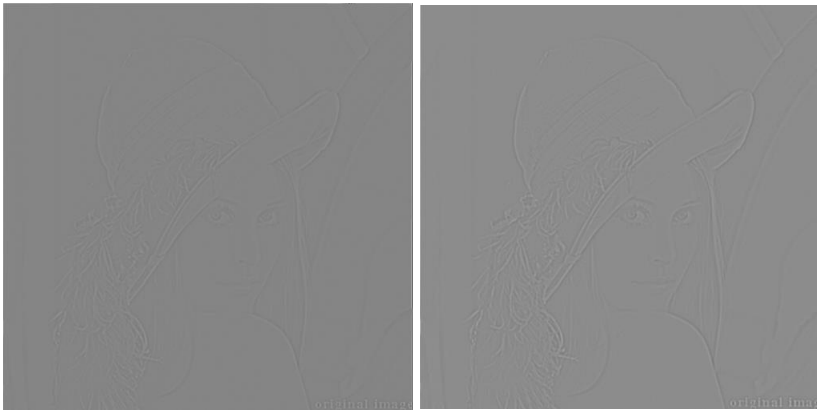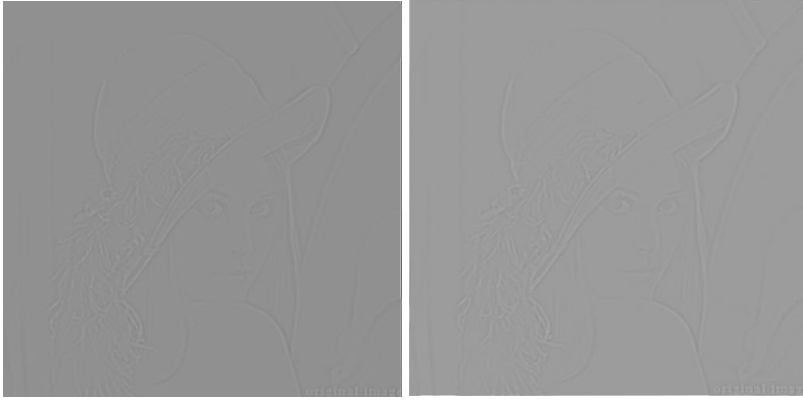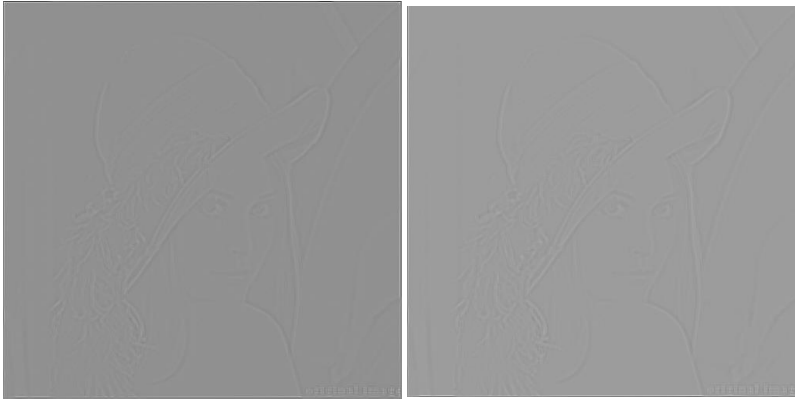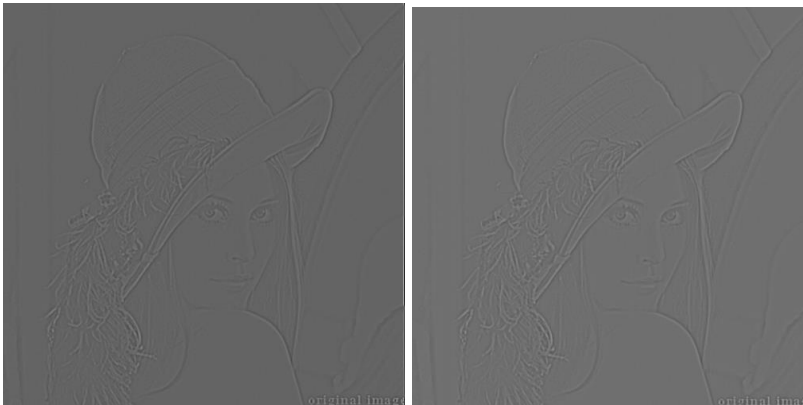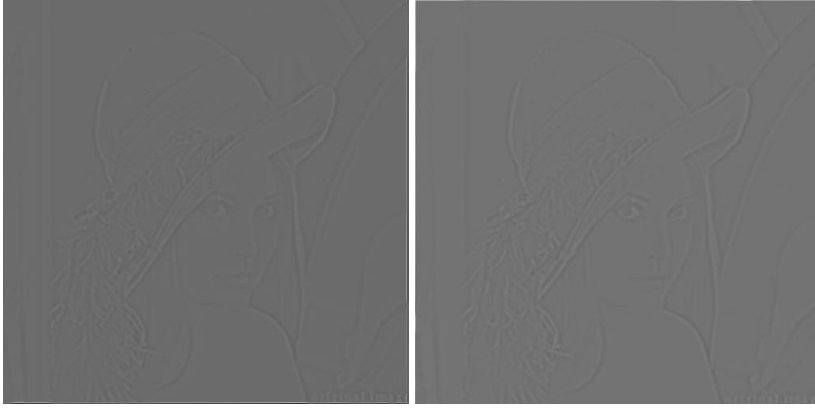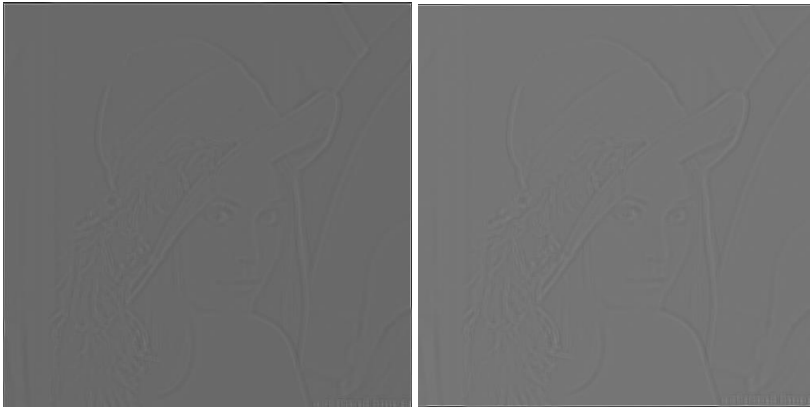

Kernel: 5x5 ve Sigma: 1
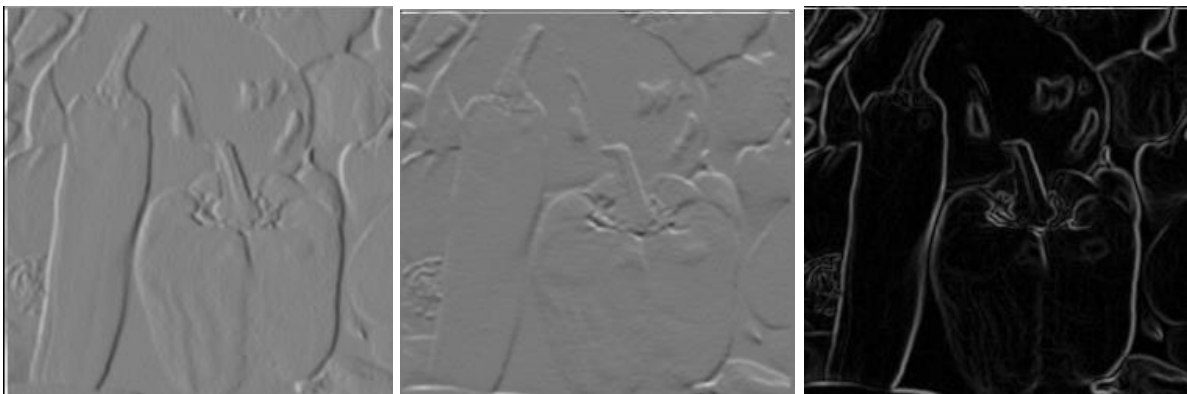
Kernel: 5x5 ve Sigma: 2
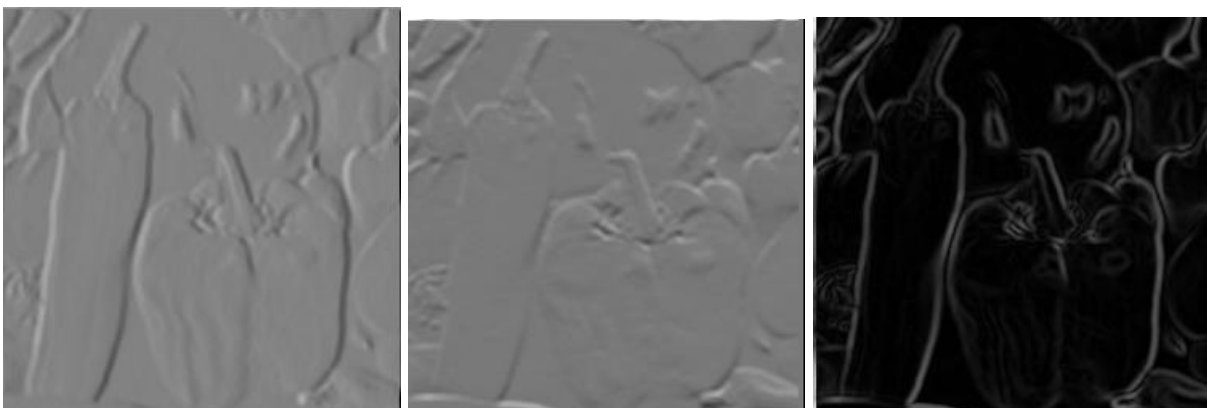
Kernel: 5x5 ve Sigma: 4
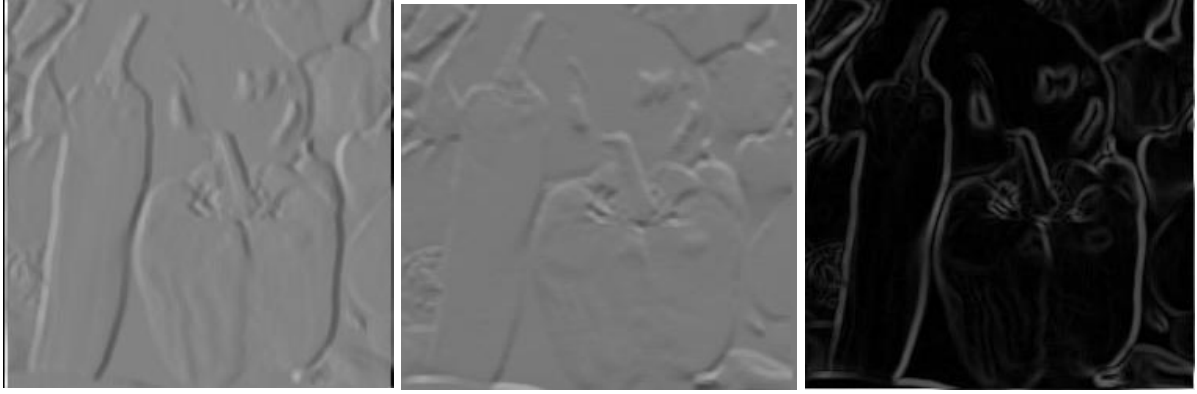
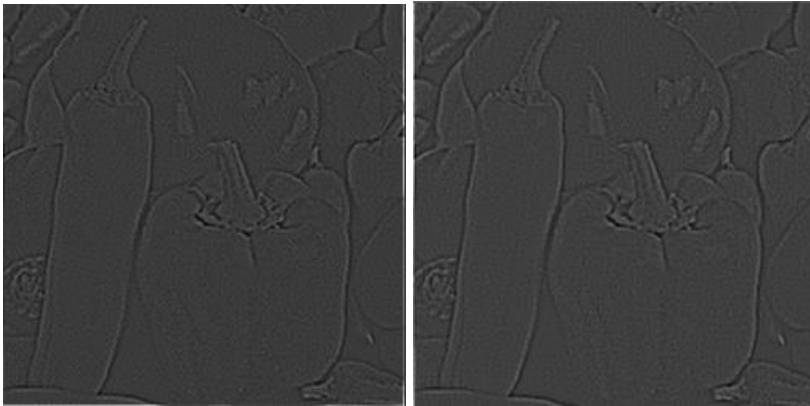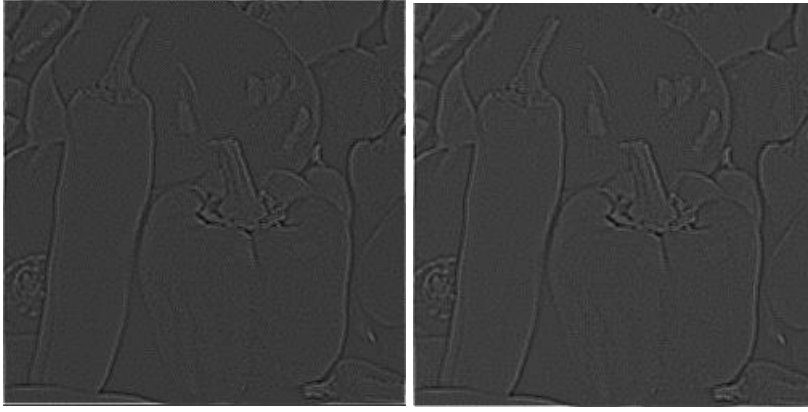Kernel: 7x7 ve Sigma: 1
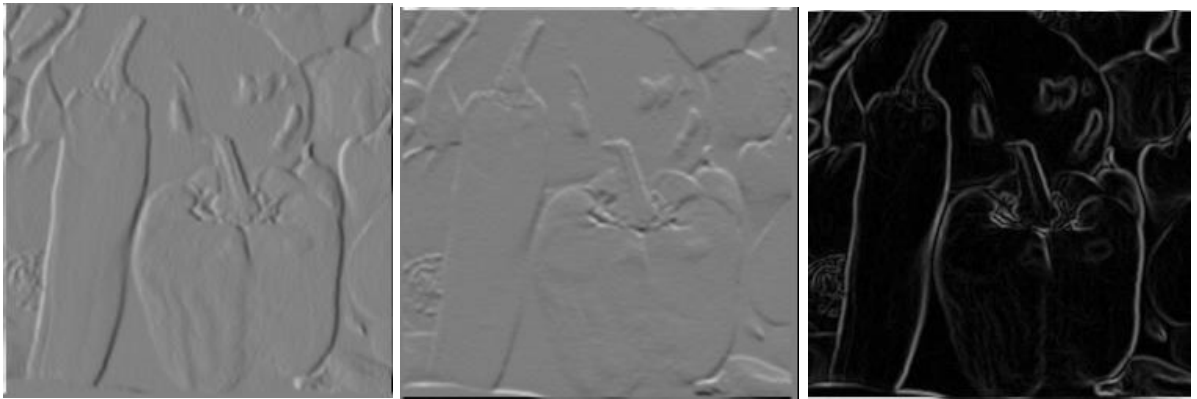
Kernel: 7x7 ve Sigma: 2
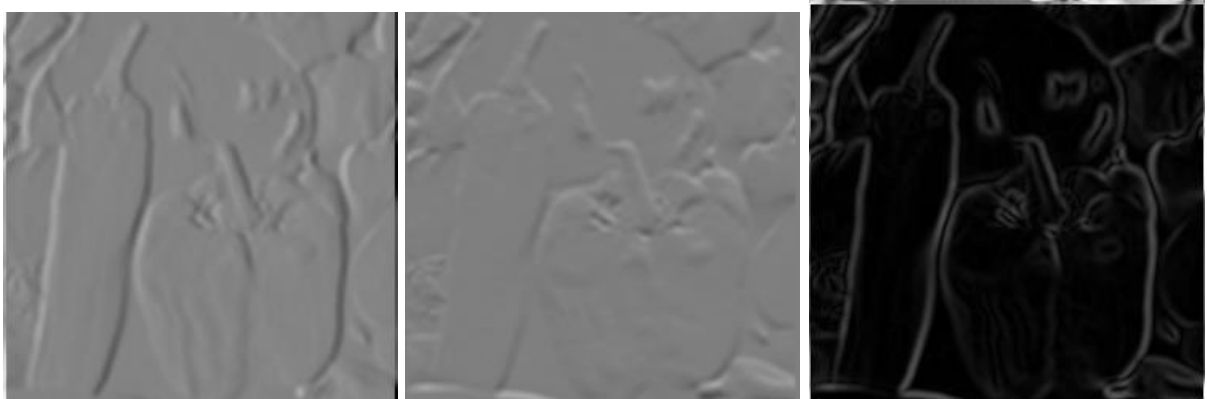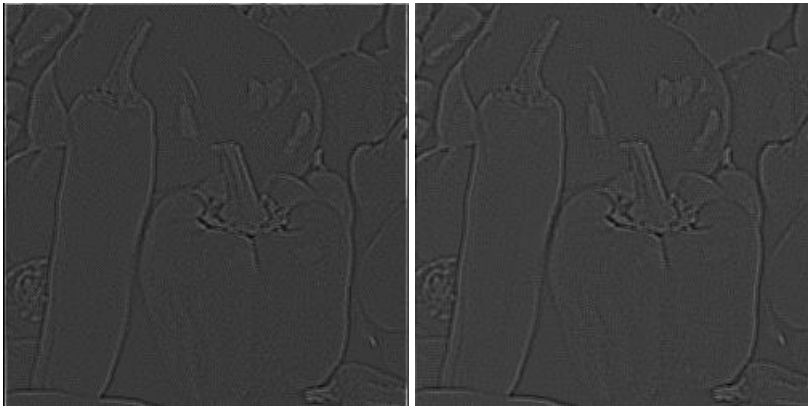

Kernel: 7x7 ve Sigma: 4
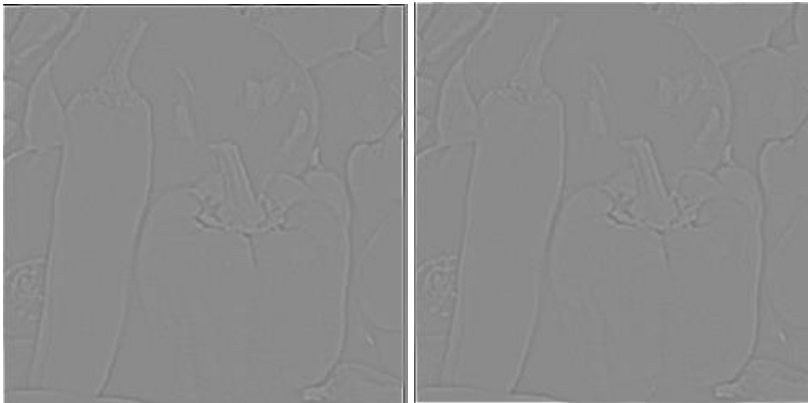
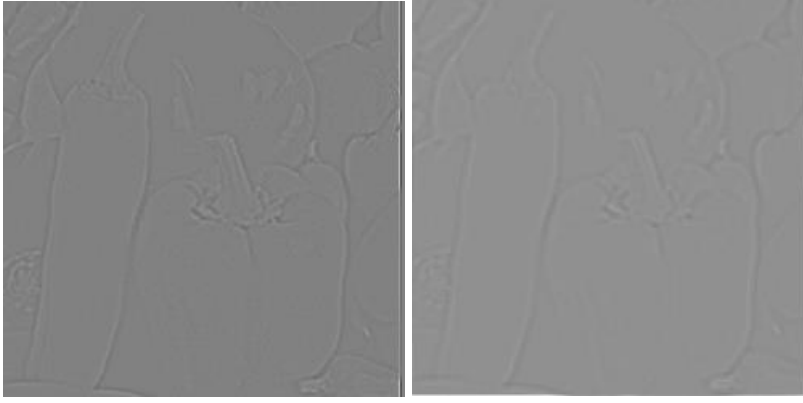Kernel: 3x3 ve Sigma: 1

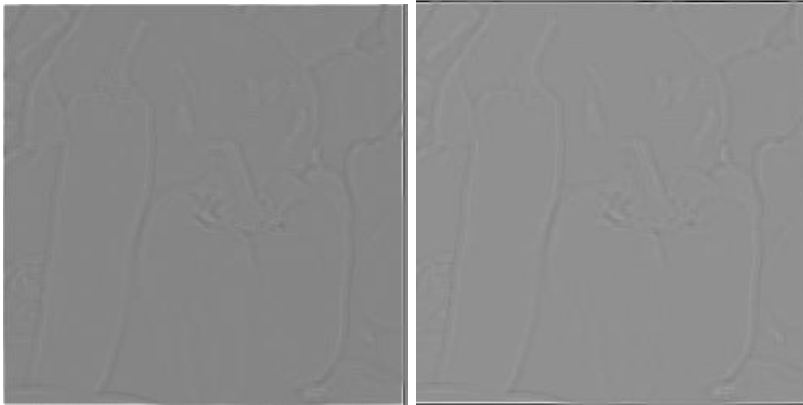
Kernel: 3x3 ve Sigma: 2

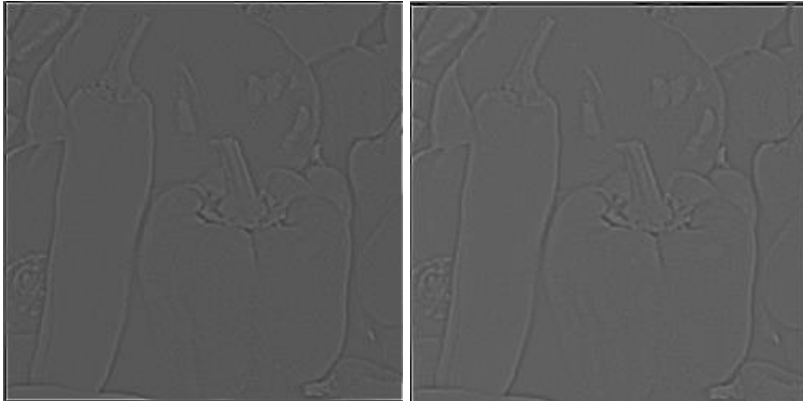Kernel: 3x3 ve Sigma: 4

Kernel: 5x5 ve Sigma: 1

Kernel: 5x5 ve Sigma: 2
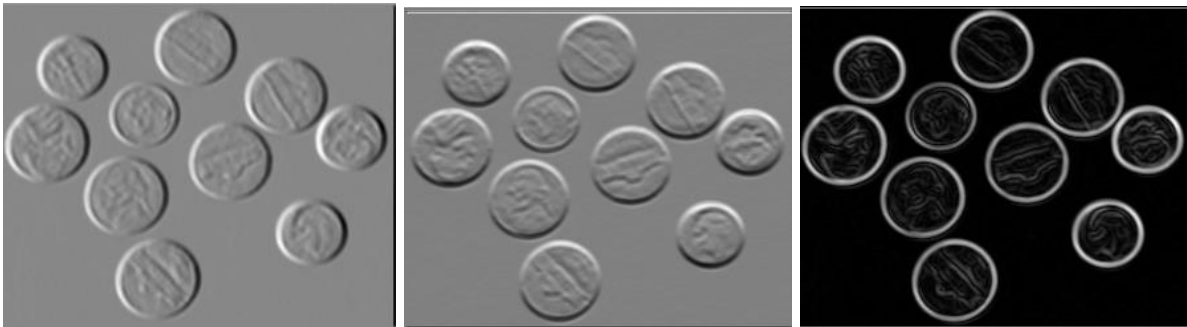
Kernel: 5x5 ve Sigma: 4

Kernel: 7x7 ve Sigma: 1

Kernel: 7x7 ve Sigma: 2

Kernel: 7x7 ve Sigma: 4

At the end of this stage, it was observed that the edge detection was more precise in blurred images. Blurring reduces noise and information we don't need in the image, allowing us to find edges and determine image gradients more accurately.

Applying Laplacian to Blurred Images

Three example .pgm formatted images are first blurred and then laplacian filter is applied.

Kernel: 3x3 ve Sigma: 1

Kernel: 3x3 ve Sigma: 2

Kernel: 3x3 ve Sigma: 4

Kernel: 5x5 ve Sigma: 1
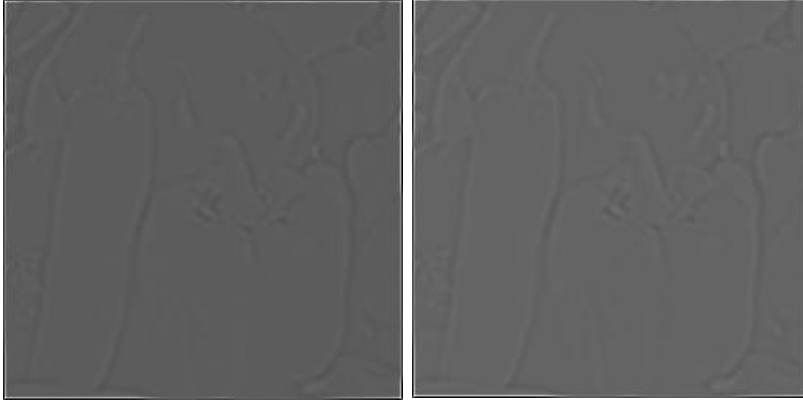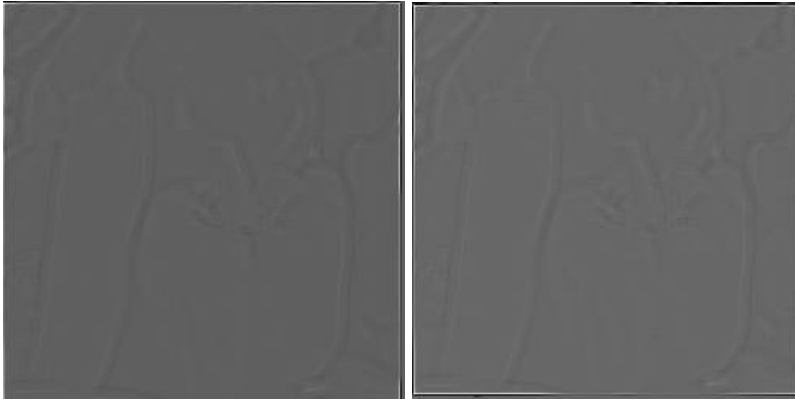

Kernel: 5x5 ve Sigma: 2

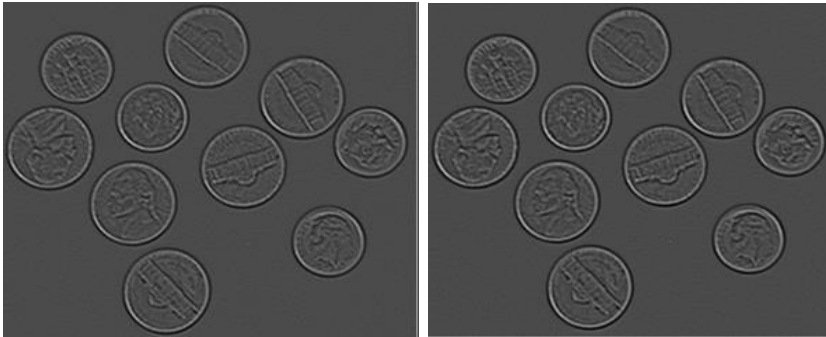Kernel: 5x5 ve Sigma: 4

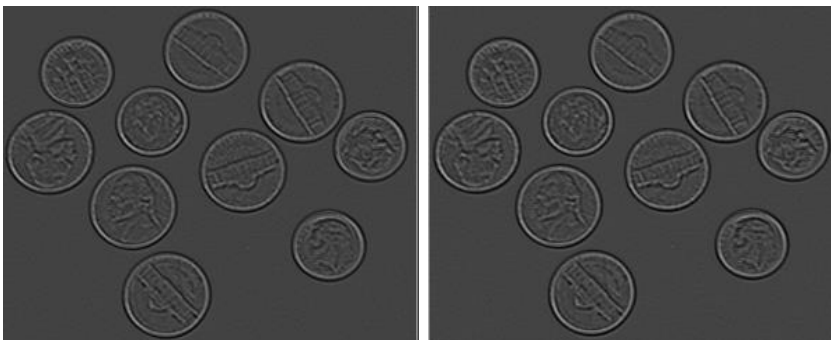Kernel: 7x7 ve Sigma: 1
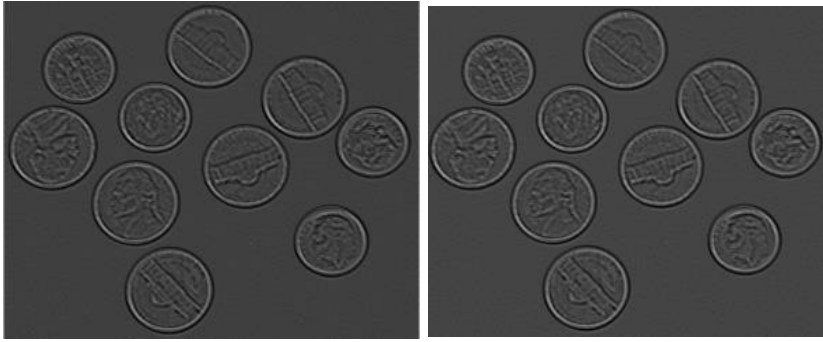

Kernel: 7x7 ve Sigma: 2

Kernel: 7x7 ve Sigma: 4
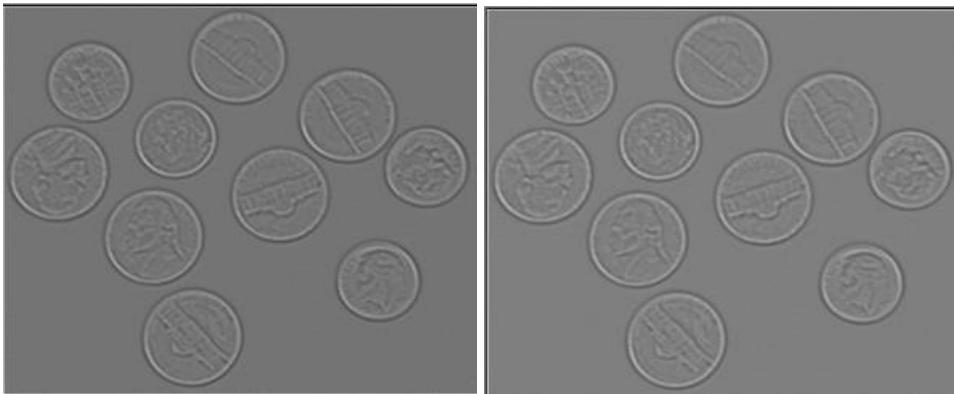

Kernel: 3x3 ve Sigma: 1
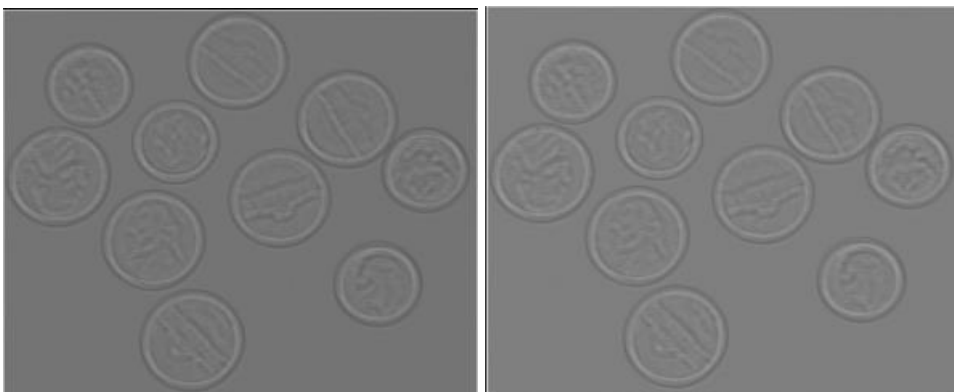

Kernel: 3x3 ve Sigma: 2

Kernel: 3x3 ve Sigma: 4

Kernel: 5x5 ve Sigma: 1

Kernel: 5x5 ve Sigma: 2

Kernel: 5x5 ve Sigma: 4

Kernel: 7x7 ve Sigma: 1

Kernel: 7x7 ve Sigma: 2

Kernel: 7x7 ve Sigma: 4

Kernel: 3x3 ve Sigma: 1

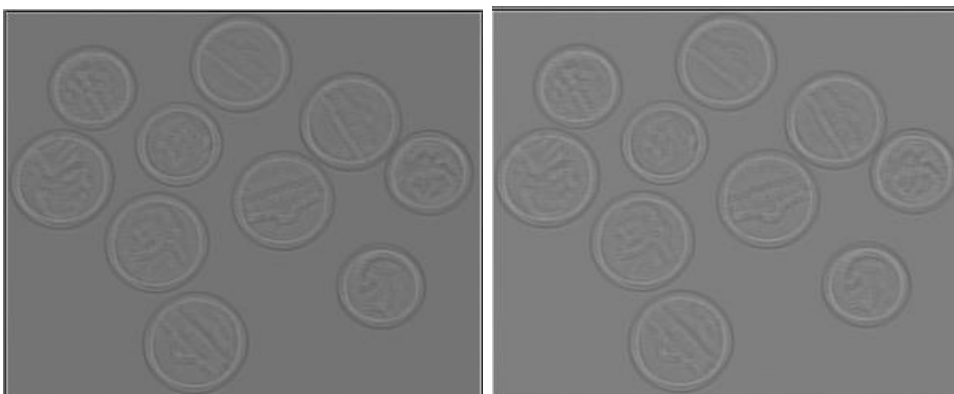Kernel: 3x3 ve Sigma: 2

Kernel: 3x3 ve Sigma: 4

Kernel: 5x5 ve Sigma: 1

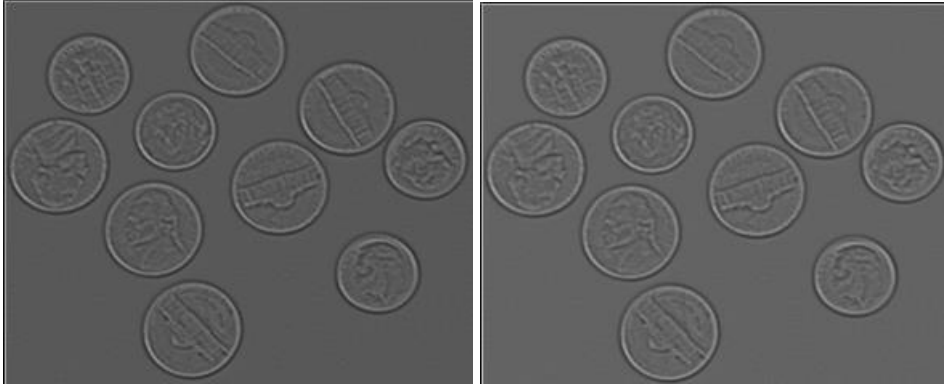Kernel: 5x5 ve Sigma: 2
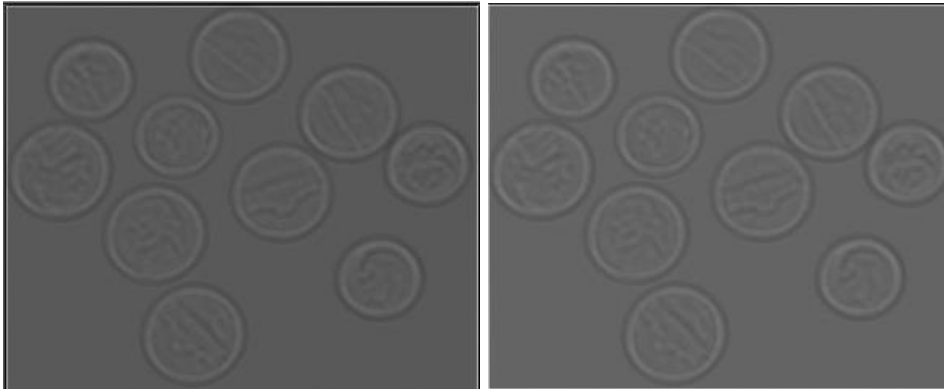

Kernel: 5x5 ve Sigma: 4
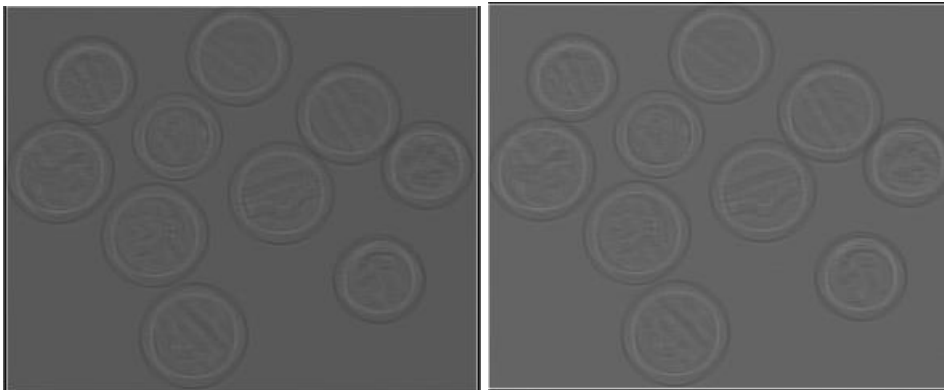

Kernel: 7x7 ve Sigma: 1

Kernel: 7x7 ve Sigma: 2

Kernel: 7x7 ve Sigma: 4

As a result of this step, Laplacian and Sobel filters were applied to the blurred images. It has been observed that the Laplacian filter is more sensitive to noise.