

[illegible]

- my_average_filter

```
38 def my_average_filter(src, fshape, verbose=False):
39     h, w = src.shape
40     h, w = fshape
41     if verbose:
42         print('average filtering')
43
44     #####
45     # TODO #
46     # average filter 생성 #
47     #####
48
49     filter = np.full(fshape, 1/(h*w))
50
51     if verbose:
52         print('<average filter> - shape:', fshape)
53         print(filter)
54
55     dst = my_filtering(src, filter)
56     return dst
```

- my_get_Gaussian_filter

```
74 def my_get_Gaussian_filter(fshape, sigma=1):
75     (f_h, f_w) = fshape
76
77     y, x = np.mgrid[-(f_h//2):(f_h//2)+1, -(f_w//2):(f_w//2)+1]
78     gaus_x = (1/(math.sqrt(2*math.pi)*sigma))*np.exp(-(x**2)/(2*(sigma**2)))
79     gaus_y = (1/(math.sqrt(2*math.pi)*sigma))*np.exp(-(y**2)/(2*(sigma**2)))
80     if f_w == 1 and f_h != 1:
81         gaus_x = 1
82     elif f_h == 1 and f_w != 1:
83         gaus_y = 1
84     filter_gaus = gaus_x * gaus_y
85     filter_gaus = filter_gaus/np.sum(filter_gaus)
86
87     return filter_gaus
```

2. 코드 설명

- my_filtering

첫 번째 for문은 pad_img의 열을 돌고, 두 번째 for문은 pad_img의 행을 돈다. 세 번

째 for문은 filter의 열을 돌고, 네 번째 for문은 filter의 행을 돈다. cut_img는 pad_img에서 필터의 크기만큼 잘라온 이미지이다. cut_img와 filter를 convolution 연산해서 새로운 이미지 dst를 만들어낸다.

- my_average_filter

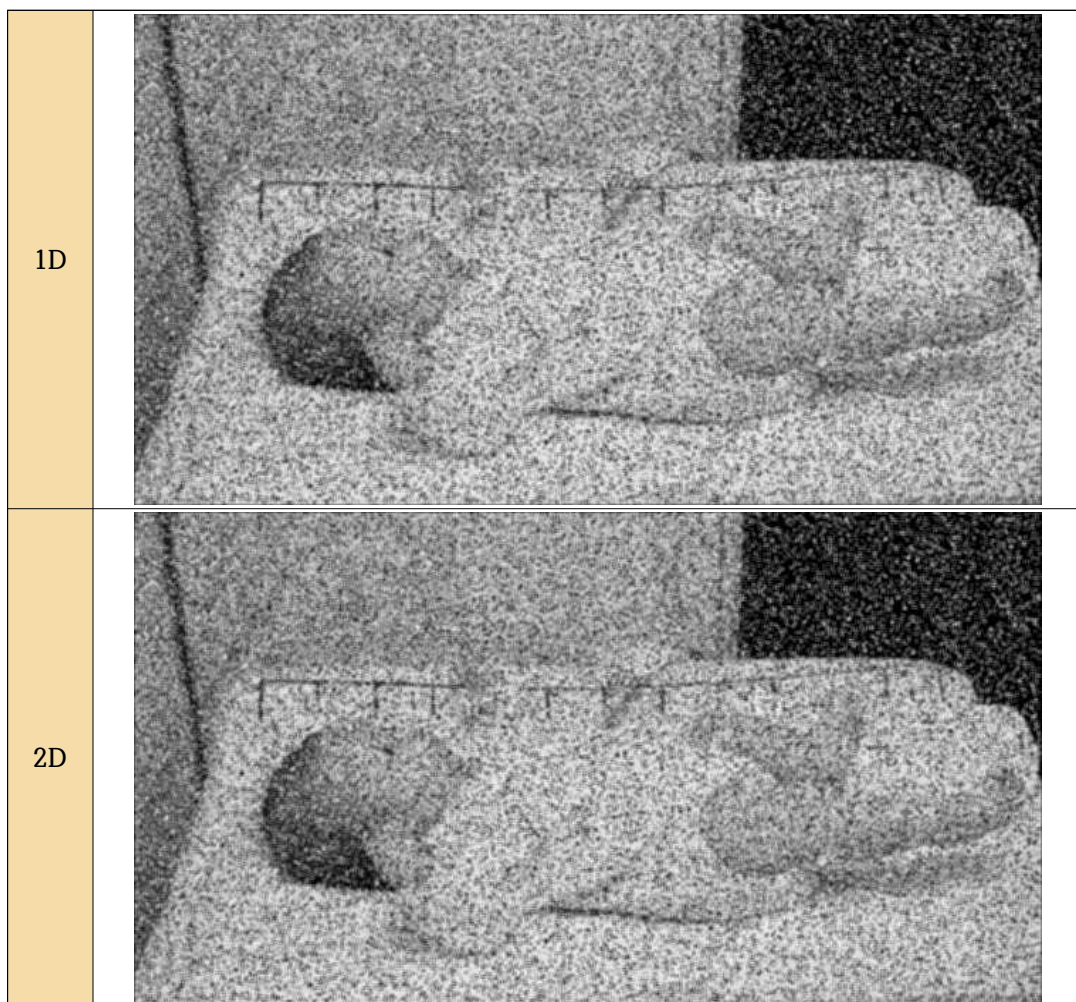
필터의 행 수(h)와 열 수(w)를 곱해 필터의 픽셀 개수를 구하고 픽셀 개수로 1을 나눈 값으로 필터를 채운다.

- my_get_Gaussian_filter

2차 가우시안 수식을 각 축에 대해 분리한 수식을 이용한다. 분리된 수식을 각각 gaus_x, gaus_y로 만들고 둘을 곱한다. 이때, 필터의 행의 수가 1이면 gaus_y를 1로, 열의 수가 1이면 gaus_x를 1로 만들어주어 곱했을 때 값에 영향을 주지 않도록 한다.

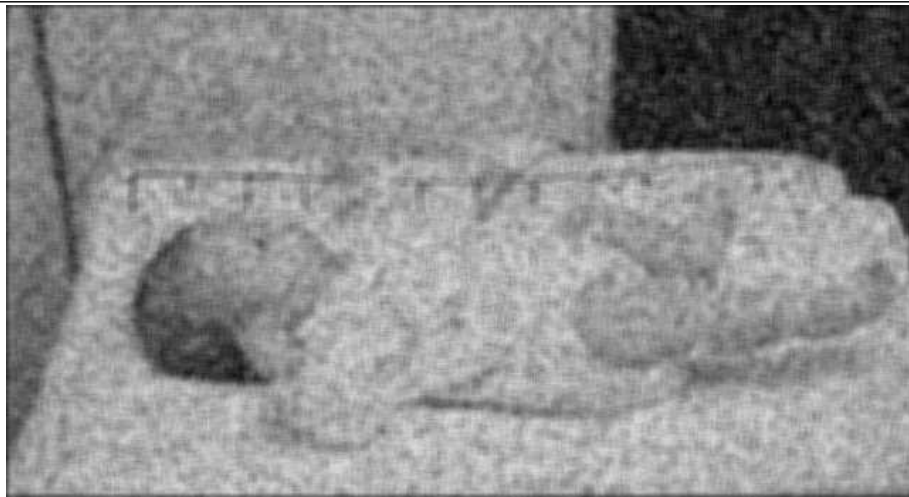
3. 이미지

① 3x3 average filter

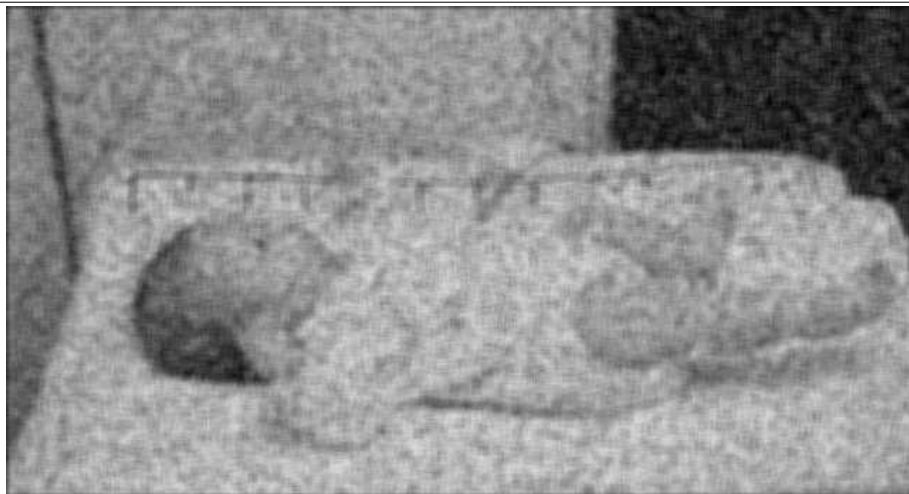


② 7x7 average filter

1D

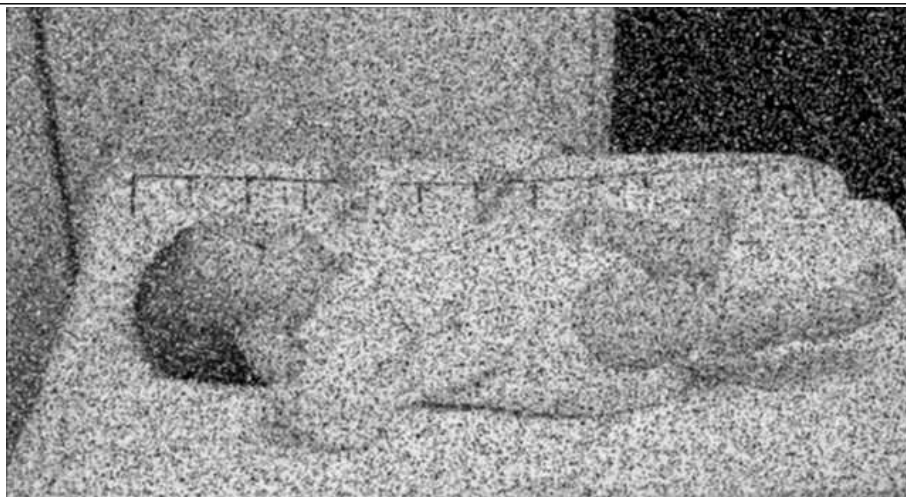


2D



③ 3x3 gaussian filter

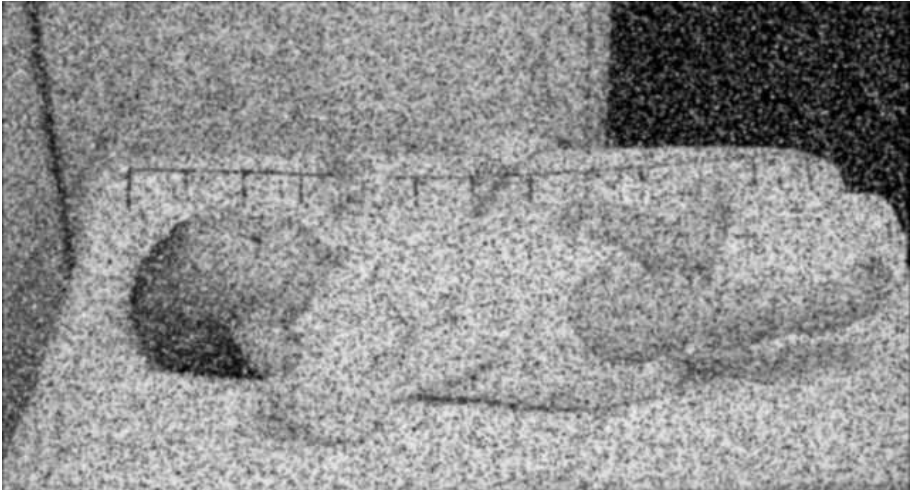
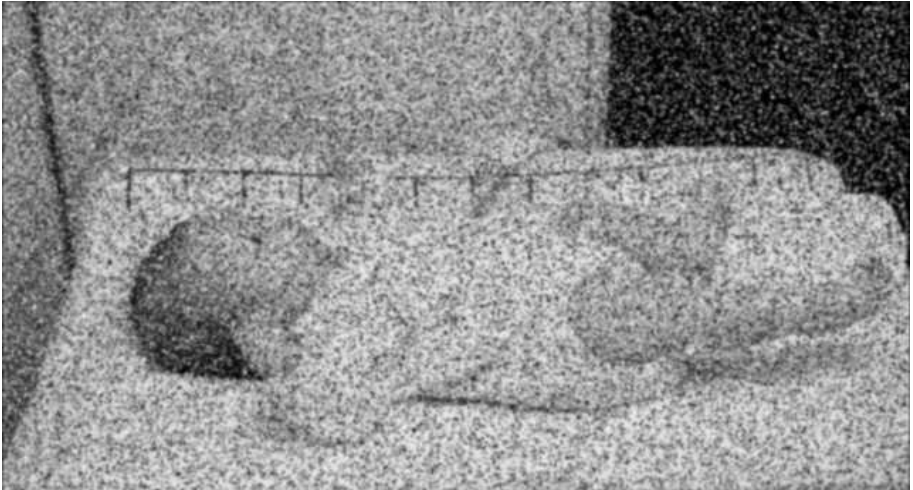
1D



2D



④ 7x7 gaussian filter

1D	
2D	

→ 필터 크기에 따른 사진 변화 : 두 필터 모두 필터 크기가 커졌을 때 더 뿌연 사진이 출력된다. average 필터의 경우 필터의 크기에 따라 큰 변화가 나타났지만 가우시안 필터는 필터의 크기만 키웠을 때 큰 변화가 나타나진 않았다.

→ 1D 필터와 2D 필터의 시간 차이 : 필터의 크기가 작을 땐 시간이 많이 차이 나지 않았지만 필터의 크기가 커지면 1D 필터의 시간이 훨씬 적게 나왔다.

→ 시그마 값에 따른 사진 변화 : 가우시안 필터는 시그마와 필터의 크기 둘 다 키웠을 때 큰 변화가 나타났다.

4. 느낀점

이론 수업을 들을 땐 다 이해했다고 생각했는데 코드로 구현하다보니 내가 잘못 생각하고 있던 부분들이 꽤 있었다. 2D의 필터를 2개의 1D 필터로 바꿀 때 1D 필터 모양대로 2D 필터 내부의 값만 가져오느라 제대로 된 값이 나오지 않았다. 가우시안 수식을 잘 이해하지 못해서 필터의 값이 이상하게 나왔다. 잘못된 부분들을 고치면서 자연스럽게 이해하게 되었고 공부에 많은 도움이 되었다.

5. 과제 난이도

적당한 것 같다.