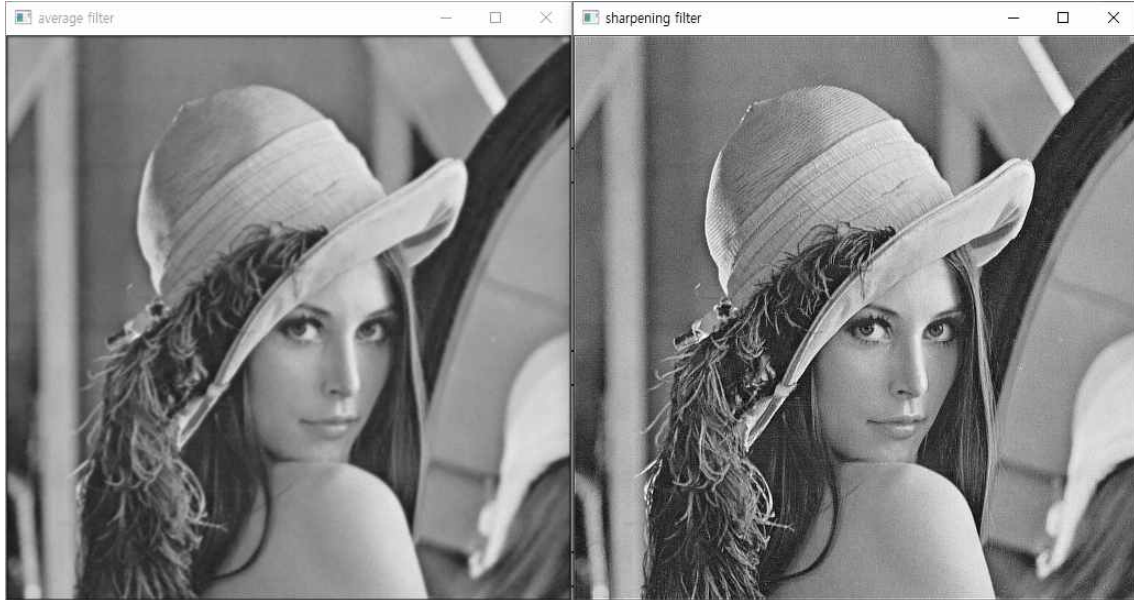


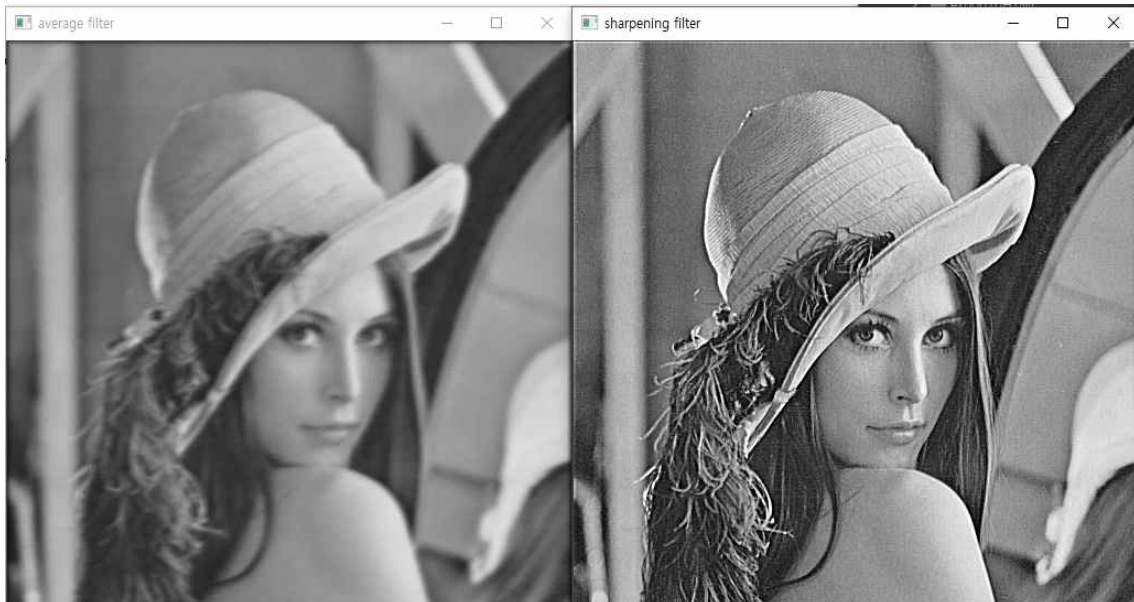
영상처리
- image filtering -

제출일자	2021.04.02
분 반	01
이 름	강인한
학 번	201701969

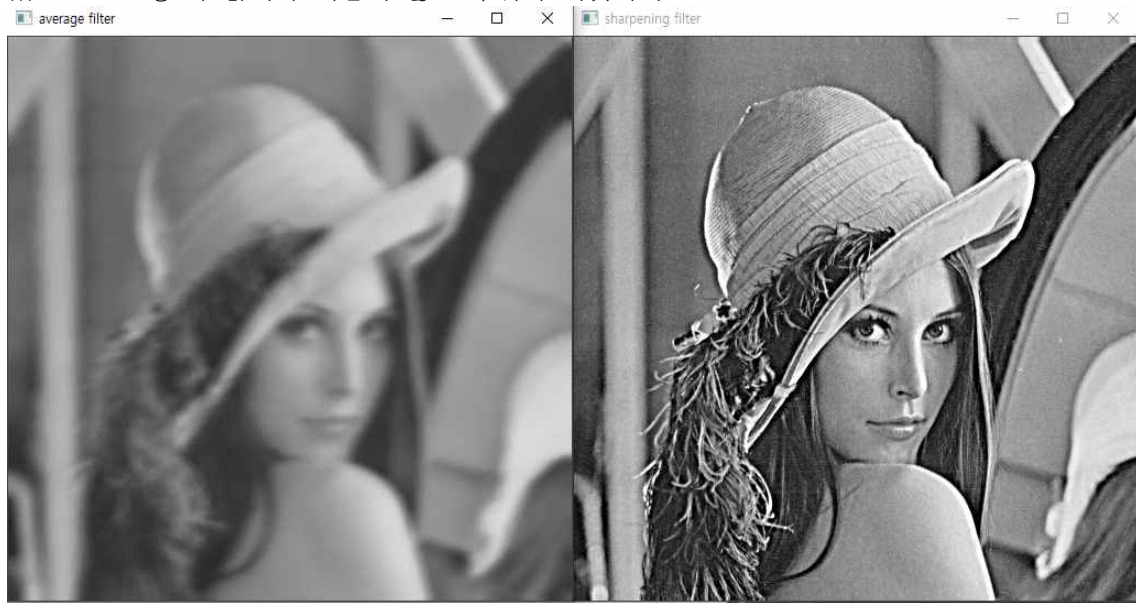
3x3 no repetition : repetition을 하지 않아 average 필터는 edge 부분이 어둡고 sharpening 필터는 edge가 밝다.



5x7 no repetition : 3x3보다 average는 더 흐릿하고 sharpening은 더 선명하다.

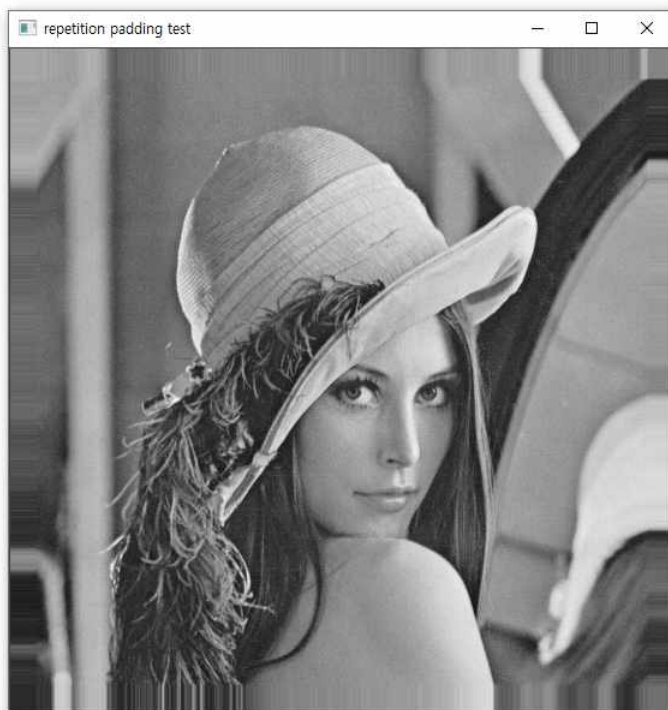


11x13 repetition : average는 더 흐릿하고 sharpening은 더 선명하다. repetition을 사용하였으므로 edge가 밝거나 어둡지 않고 주위와 비슷하다.



```
rep_test = my_padding(src, (20,20), 'repetition')
```

20x20 repetition



my_padding 함수

repetition이 인자로 들어온 경우의 my_padding 함수는 repetition을 해주는 작업이 필요하다.

```
# repetition되도록 상하 좌우 순서로 값을 넣음
for h1 in range(0, p_h):
    for w1 in range(p_w, w+p_w):
        pad_img[h1, w1] = pad_img[p_h, w1]

for h1 in range(h+p_h, (2*p_h)+h):
    for w1 in range(p_w, w+p_w):
        pad_img[h1, w1] = pad_img[h+p_h-1, w1]

for h1 in range(0, 2*p_h+h):
    for w1 in range(0, p_w):
        pad_img[h1, w1] = pad_img[h1, p_w]

for h1 in range(0, 2*p_h+h):
    for w1 in range(p_w+w, w+(2*p_w)):
        pad_img[h1, w1] = pad_img[h1, w+p_w-1]
```

my_filtering 함수

```
if ftype == 'average':
    print('average filtering')

    # np.full 함수를 이용해 numpy를 만들 average이므로 같은 값
    mask = np.full((fshape[0], fshape[1]), 1/(fshape[0]*fshape[1]))
```

average의 경우 1을 필터의 가로*세로로 나눈 값으로 채운다.

```
elif ftype == 'sharpening':
    print('sharpening filtering')
    # -1/(가로*세로) 값을 전체에 넣어주고 중앙 한값에 2- 1/(가로*세로)을 넣는다.
    mask = np.full((fshape[0], fshape[1]), -1 / (fshape[0] * fshape[1]))
    mask[fshape[0]//2][fshape[1]//2] = 2 - 1 / (fshape[0] * fshape[1])
```

sharpening의 경우 -1/가로*세로로 모두 채운 뒤 중앙 값에 2-1/(가로*세로)를 넣는다

이미지에 마스크를 씌워 주는 작업

```
#sum을 이용하여 filter된 값을 구해 저장
for c in range(h):
    for r in range(w):
        image_piece = src_pad[c:c+fshape[0],r:r+fshape[1]]
        if np.sum(mask*image_piece)>=255:
            src_pad[c+(fshape[0]//2)][r+(fshape[1]//2)]=255
        elif np.sum(mask*image_piece)<0:
            src_pad[c+(fshape[0]//2)][r+(fshape[1]//2)]=0
        else:
            src_pad[c+(fshape[0]//2)][r+(fshape[1]//2)]=(np.sum(mask*image_piece))

src_pad = (src_pad+0.5).astype(np.uint8)
return src_pad[fshape[0]//2:h+fshape[0]//2,fshape[1]//2:w+fshape[1]//2]
```

if 문을 이용해 오버플로우와 언더플로를 바로 잡아준다.

return 값에는 my_filtering 함수로 인해 조금 커졌던 이미지를 기존의 이미지와 같은 크기로 출력하기 위해 복잡한 식이 들어간다.

3	난이도 및 느낀 점
---	------------

존재 이유를 알 수 없는 코드가 몇 개 있어 조금 어려웠던 것 같습니다. 그래도 두 번째 과제라 저번 과제보단 익숙하게 진행하였습니다. 느낀 점으로는 마스크의 크기를 큰 필터를 씌울수록 average는 흐릿해지고 sharpening은 선명해지는 걸 알았습니다.