

# 영상처리 실습 보고서

11주차: Bilateral filtering

학번	201802170
이름	하 상 호

## 1. 과제의 내용

my\_bilateral 함수 및 main 완성

mybilateral 함수 빈 칸 채우기

img\_pad는 padding 된 이미지를 뜻함

보고서에 원본과 결과 사진 비교 및 수행 시간 기록

## 2. 과제의 해결 방법

과제의 내용을 해결하기 위해 어떠한 방법을 사용했는지 자세하게 기술한다.

```
def my_bilateral(src, msize, sigma, sigma_r, pad_type='zero'):  
    #####  
    # TODO #  
    # my_bilateral 완성 #  
    # mask를 만들 때 4중 for문으로 구현 시 감점(if문 fsize*fsize개를 사용해서 구현해도 감점) #  
    #####  
    (h, w) = src.shape  
  
    dst = np.zeros((h, w))  
  
    #mask 를 씌우기 위해 padding  
    img_pad = my_padding(src, (int(msize / 2), int(msize / 2)), pad_type)  
  
    y, x = np.mgrid[-int(msize / 2):int(msize / 2) + 1, -int(msize / 2):int(msize / 2) + 1]  
    #exp1  
    exp1 = np.exp(-((x ** 2 + y ** 2) / (2 * sigma ** 2)))
```

My bilateral 함수 구현

Img 의 h, w 를 받고

받은 img 를 통해 dst 변수의 np.zero 를 통해 zero 로 채워진 마스크를 선언한다.

Img\_pad 변수

Mask를 씌우기 위해 padding 작업을 진행한다.

. y, x 변수

Np.mgrid 를 통해 차원을 나누어 준다. 지금 까지 해온 실습과 동일

Exp1

$$\frac{1}{N} \exp\left(-\frac{(i-i)^2}{2\sigma_x^2} - \frac{(j-j)^2}{2\sigma_y^2}\right) \cdot \exp\left(-\frac{(I(i,j) - I(i,j))^2}{2\sigma_r^2}\right)$$

해당 식 중 exp1 은

$$\exp\left(-\frac{(i-i)^2}{2\sigma_x^2} - \frac{(j-j)^2}{2\sigma_y^2}\right)$$

해당 부분에 해당 하는데

시그마 x 와 y 는 동일 하다는 가정이 들어갔기 때문에 위의 계산식과 동일하다.

해당부분은 좌표를 통한 exp 작업을 진행 한 것이다.

```
for i in range(h):
    print('\r%d / %d ...' % (i, h), end=" ")
    for j in range(w):
        diff = src[i, j] - img_pad[i + int(msize / 2) + y, j + int(msize / 2) + x] # I[i, j] - I[k, l]
        exp2 = np.exp(-(diff ** 2) / (2 * sigma_r ** 2))
        mask = exp1 * exp2

        mask /= np.sum(mask) # EXP1 * EXP2 / N

    dst[i, j] = np.sum(mask * img_pad[i:i+msize, j:j+msize])

return dst
```

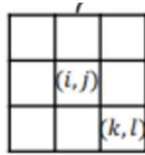
해당 코드는 exp2 를 구현하기 위한 코드이다.

Exp2 는 위 식 중

$$\exp\left(-\frac{(I(i,j) - I(k,l))^2}{2\sigma_r^2}\right)$$

위 부분에 해당한다.

Exp2 는 해당 픽셀 값을 이용한다.



해당 마스크에서  $i, j$  부터 떨어진  $k, l$  의 위치에 해당하는 픽셀 값들을 접근하며 이용해야 하기 때문에 2중 포문을 사용한다.

Diff 변수

오리지널 이미지의 값에서 `img_pad` 값들을 빼준다.

`Img_pad` 의 index가 저렇게 된 이유는

오리지널 이미지의 위치 + pad size +  $k, l$  의 위치를 옮기기 위함이다.

`Exp2` 를 형성 하였다면

`Mask` 를 통해 두 `exp` 를 곱해주고 `Np.sum(mask)` 를 나누어준다.

나누어 주는 이유는 마스크의 값들의 합은 1이 되어야 하기 때문이다.

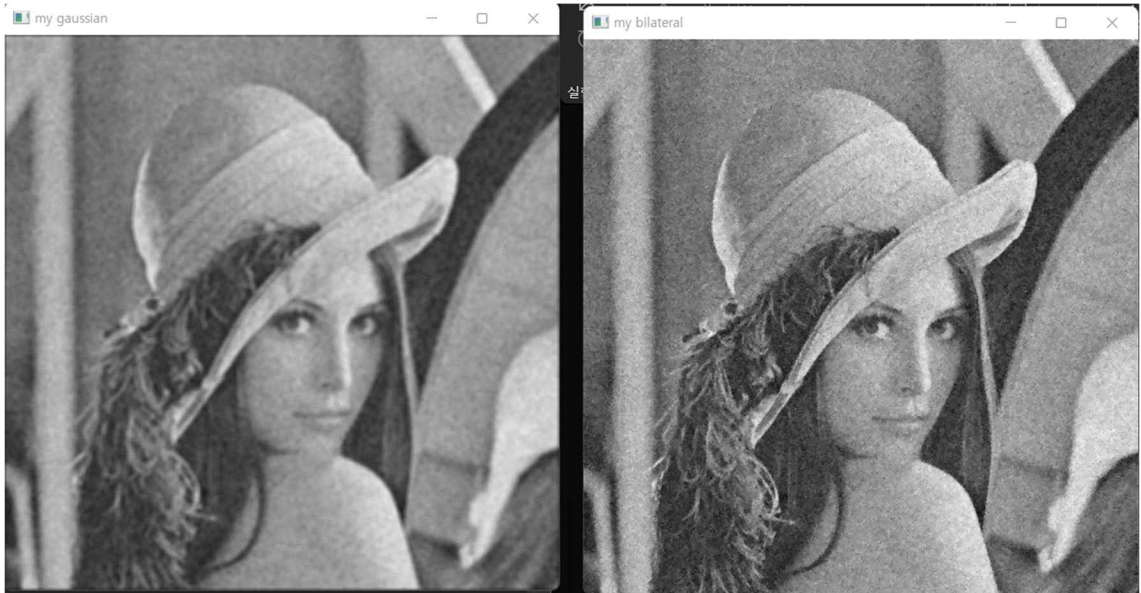
위 작업이 끝난 후 `dst` 변수에 `np.sum`을 통해 `mask` 와 `img_pad` 의 위치를 설정하여 삽입해준다.

### 3. 결과물

결과물이 잘 보이도록 화면을 캡처해 보고서에 올린다.

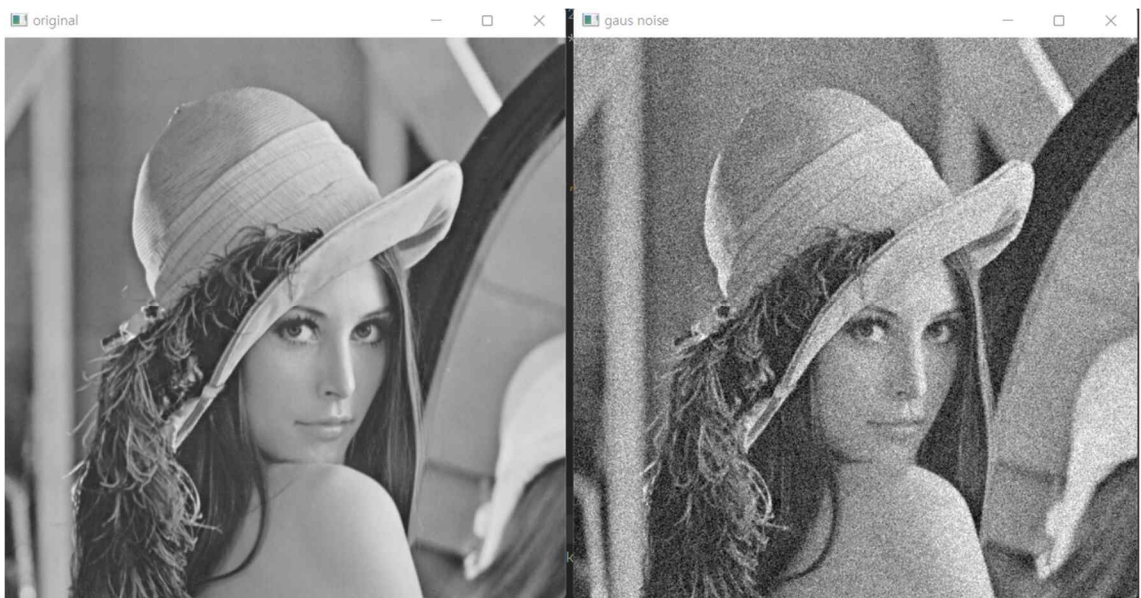
My gaussian

My bilateral



Original

gaus noise



time

```
C:\Users\sangho\anaconda3\python.exe "C:/Users/sangho/OneDrive/Desktop/새 폴더/bila  
511 / 512 ...  
time : 6.863156795501709
```