

# 영상처리 실습 보고서

6주차: Derivative of Gaussian(DoG)

학번	201802170
이름	하 상 호

## 1. 과제의 내용

- get\_DoG\_filter(fsize,sigma=1) 함수 구현
- main()
- get\_DoG\_filter 함수를 통해서 mask 생성
- dst\_x, dst\_y를 사용하여 magnitude를 계산

## 2. 과제의 해결 방법

과제의 내용을 해결하기 위해 어떠한 방법을 사용했는지, 자세하게 기술한다.

본 과제는 실습 , 이론 내용을 바탕으로 구현 하였다.

### • DoG 수행 절차

( $\frac{1}{2\pi\sigma^2}$ 는 없어도 됨)

1. 가우시안 분포 함수 :  $G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$  (원본 이미지 :  $I$ )

2.  $\frac{d}{dx}G(x, y) = -\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \frac{d}{dy}G(x, y) = -\frac{y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

3.  $DoG(x) = \frac{d}{dx}G(x, y) * I$

4.  $DoG(x), DoG(y)$ 의 magnitude를 계산

위 구현 과정을 진행한다.

```

y, x = np.mgrid[-fsize:fsize + 1, -fsize:fsize + 1]

mask = x ** 2 + y ** 2
sig2 = sigma ** 2

factor = np.e ** (-(mask / (2 * sig2)))

DoG_x = (-x / sig2) * factor
DoG_y = (-y / sig2) * factor

```

2번) 먼저 가우시안 분포함수를 미분한 값을 통해 Dog\_x , DoG\_y를 선언

```

x = ((x - np.min(x)) / (np.max(x) - np.min(x))).astype(np.float32) |
y = ((y - np.min(y)) / (np.max(y) - np.min(y))).astype(np.float32)

```

x , y 는 필터를 시각화 하기 위한 변수이다.  
필터를 시각화 하기 위해선 정규화 과정이 필요하다.  
본 정규화 과정은 공식을 통해 구현 하였다.  
minmax scale을 사용하였다.

3번) 은 my\_filtering 함수를 통해 conv를 진행 해 준다.

출처 : <https://hwiyong.tistory.com/91>

### MinMax(Normalization)

일정 범위로 scaling을 적용해줍니다. 대표적으로 [0, 1]을 많이 사용합니다.

$$X = \frac{x - x_{min}}{x_{max} - x_{min}}$$

```

dst = np.sqrt(dst_x**2 + dst_y**2)

```

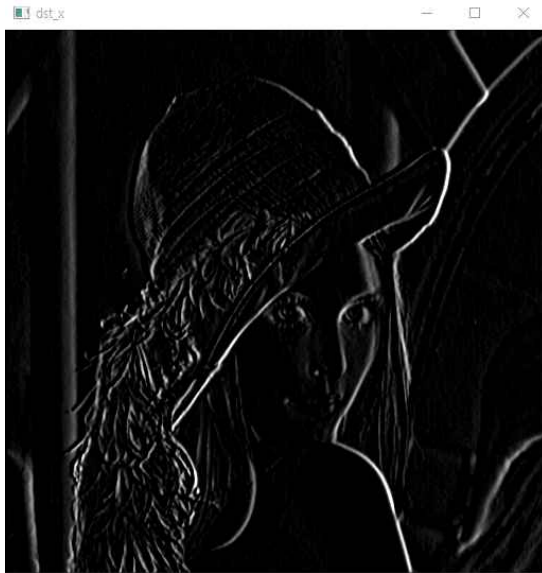
4번의 경우 conv 한 dst\_x 와 dst\_y를 통해 식을 구성한다.

dst 의 결과가 pdf 와 달라 예외처리를 포함한 filtering을 사용하였을 때 다르다는 것을 발견 하였다. 따라서 예외처리를 제외하여 실습pdf 와 같은 결과를 도출하였다.

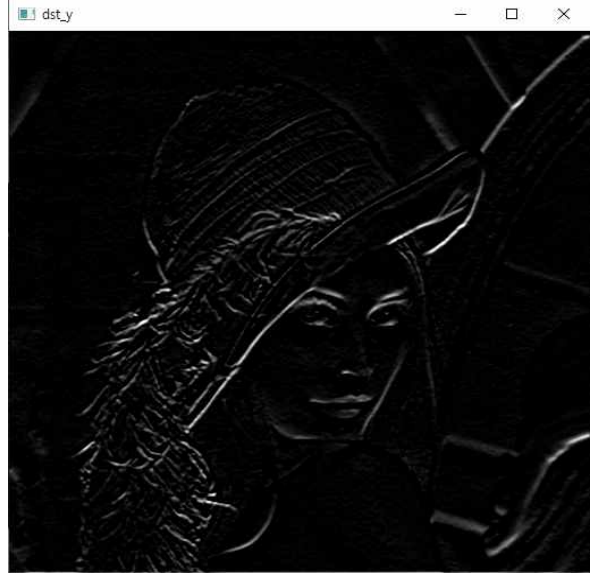
### 3. 결과물

결과물이 잘 보이도록 화면을 캡처해 보고서에 올린다.

dst\_x



dst\_y



dst

