

# Image Processing

실습 3주차

김 현 섭

Department of Computer Science and Engineering

Chungnam National University, Korea



# 실습 소개

- 과목 홈페이지

- DSC 공유대학 사이버 캠퍼스 (<http://ecampus.dscu.ac.kr>)

- TA 연락처

- 메일 보내실 때 [IP]를 제목에 붙여주세요

- 김현섭

- [hyunseop95@gmail.com](mailto:hyunseop95@gmail.com)



# 목차

- 실습
  - Numpy
  - Image filtering
  - Padding
- 과제
  - My filtering
    - Sharpening filter
    - Average filter
    - Gaussian filter



- Numpy
  - Numpy 기초

```
import numpy as np

if __name__ == '__main__':
    v1 = np.full((3, 3), 5)
    v2 = np.ones((3, 3))

    print(v1 + v2)
    print(v1 * v2)
    print(np.sum(v1 * v2))
```

code

```
[[6.  6.  6.]
 [6.  6.  6.]
 [6.  6.  6.]]
[[5.  5.  5.]
 [5.  5.  5.]
 [5.  5.  5.]]
45.0
```

output



- Numpy
  - Numpy 기초

```
import numpy as np
import time

if __name__ == '__main__':
    v1 = np.full((1000, 1000), 3)
    v2 = np.ones((1000, 1000))

    numpy_start = time.time()
    print(np.sum(v1*v2))
    print(time.time() - numpy_start)

    sum = 0
    for_start = time.time()
    for i in range(1000):
        for j in range(1000):
            sum += v1[i, j] * v2[i, j]
    print(sum)
    print(time.time() - for_start)
```

code

```
3000000.0
0.0050258636474609375
3000000.0
2.5103886127471924

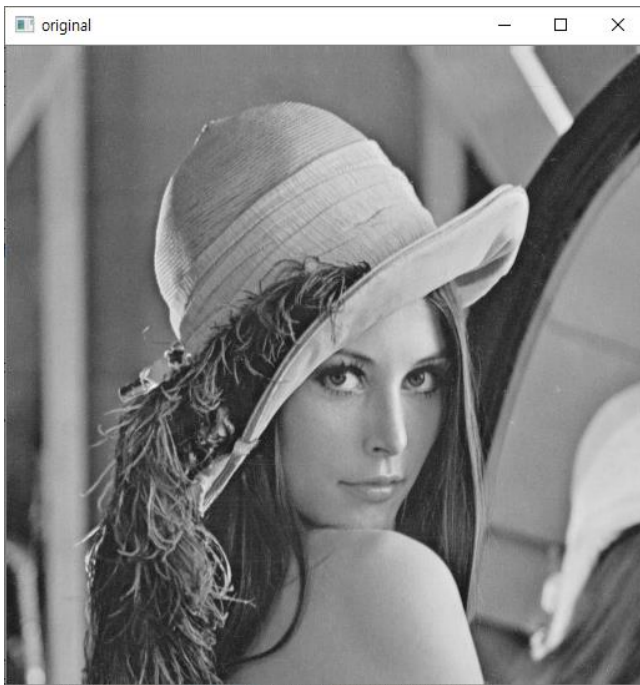
Process finished with e
```

output

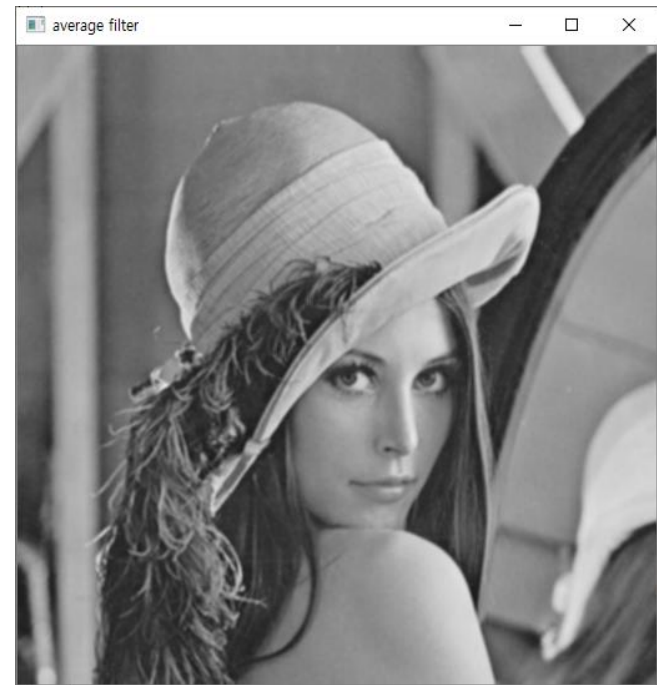


- **Image filtering**

- Average filter(평균값 필터)
  - Image를 부드럽게 해주는 효과
  - 잡음을 제거하는데 사용됨



original



3x3 average filter

## • Image filtering

### – Average filter(평균값 필터)

- Image를 부드럽게 해주는 효과
- 잡음을 제거하는데 사용됨

filter2D(src, ddepth, kernel)

- src : 이미지
- ddepth : 출력 영상 데이터 타입. -1이면 입력과 동일  
(cv2.CV\_8U, cv2.CV\_32F etc.)
- kernel : 커널 행렬

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

mask

```
def my_average_filter_3x3(src):  
    mask = np.array([[1/9, 1/9, 1/9],  
                     [1/9, 1/9, 1/9],  
                     [1/9, 1/9, 1/9]])  
    print(mask)  
  
    dst = cv2.filter2D(src, -1, mask)  
    return dst  
  
if __name__ == '__main__':  
    src = cv2.imread('../Lena.png', cv2.IMREAD_GRAYSCALE)  
    dst = my_average_filter_3x3(src)  
  
    cv2.imshow('original', src)  
    cv2.imshow('average filter', dst)  
    cv2.waitKey()  
    cv2.destroyAllWindows()
```



- Image filtering

- Average filter(평균값 필터)

- Image를 부드럽게 해주는 효과
    - 잡음을 제거하는데 사용됨

filter2D(src, ddepth, kernel)

- src : 이미지
- ddepth : 출력 영상 데이터 타입. -1이면 입력과 동일  
(cv2.CV\_8U, cv2.CV\_32F etc.)
- kernel : 커널 행렬

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

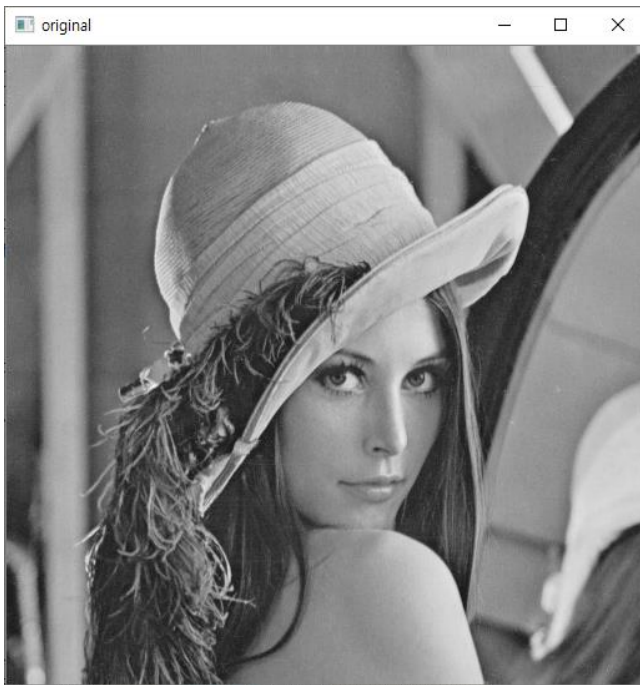
mask

```
def my_average_filter_5x5(src):  
    mask = np.array([[1/25, 1/25, 1/25, 1/25, 1/25],  
                     [1/25, 1/25, 1/25, 1/25, 1/25],  
                     [1/25, 1/25, 1/25, 1/25, 1/25],  
                     [1/25, 1/25, 1/25, 1/25, 1/25],  
                     [1/25, 1/25, 1/25, 1/25, 1/25]])  
  
    print(mask)  
  
    dst = cv2.filter2D(src, -1, mask)  
    return dst
```

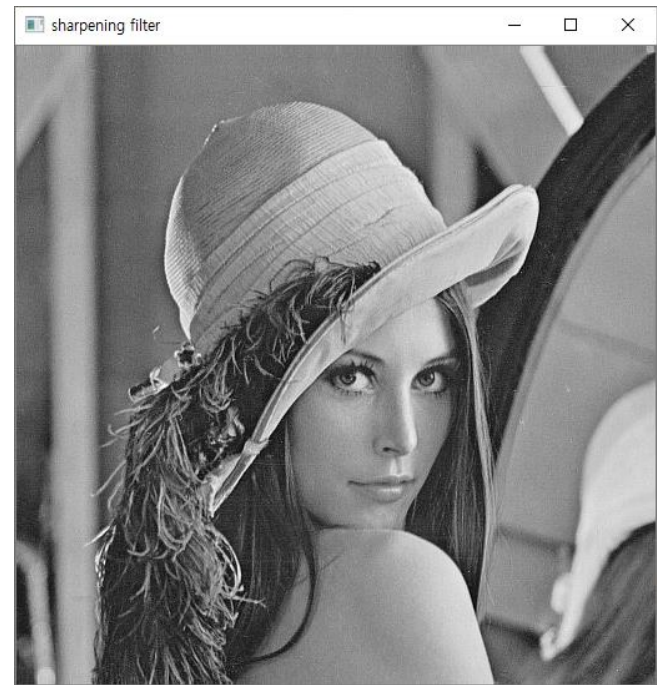




- **Image filtering**
  - Sharpening filter
    - Image를 선명하게 해주는 효과



original



3x3 sharpening filter

- **Image filtering**
  - Sharpening filter
    - Image를 선명하게 해주는 효과

0	0	0
0	2	0
0	0	0

 $-$ 

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

 $=$ 

-1/9	-1/9	-1/9
-1/9	17/9	-1/9
-1/9	-1/9	-1/9

- Image filtering
  - Sharpening filter
    - Image를 선명하게 해주는 효과

-1/9	-1/9	-1/9
-1/9	17/9	-1/9
-1/9	-1/9	-1/9

mask

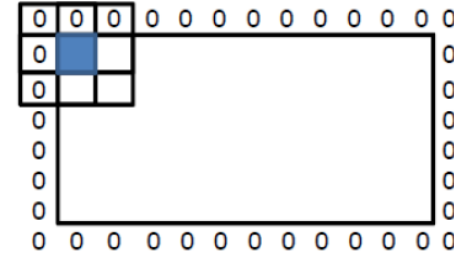
```
def my_sharpening_filter_3x3(src):  
    mask = np.array([[ -1/9, -1/9, -1/9],  
                     [ -1/9, 17/9, -1/9],  
                     [ -1/9, -1/9, -1/9]])  
    print(mask)  
  
    dst = cv2.filter2D(src, -1, mask)  
    return dst
```



- **Padding**

- Zero padding

- 이미지의 가장자리를 0으로 채움



original

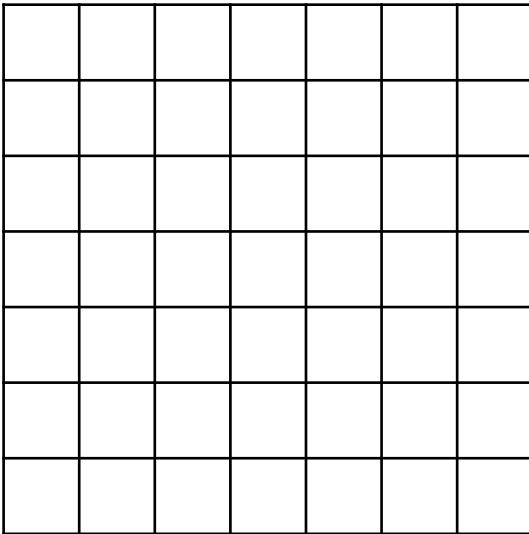


zero padding

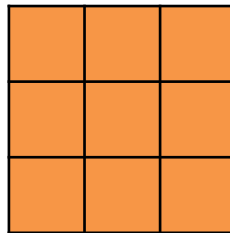
- **Padding**

- Padding을 하는 이유

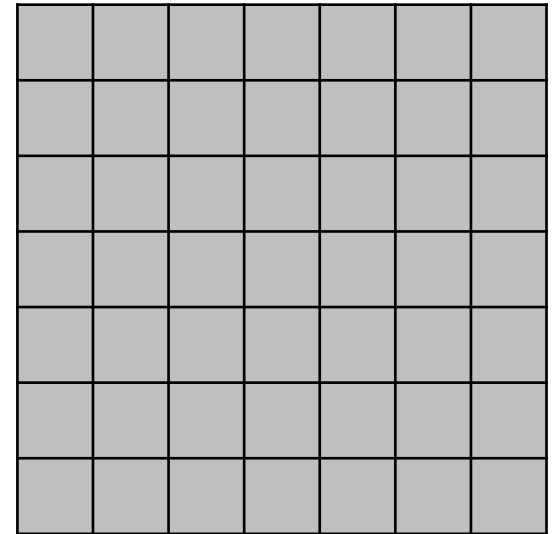
- Filtering 과정에서 생기는 문제를 막기위해



입력 이미지 (7x7)



Filter(3x3)



출력 이미지 (7x7)

- **Padding**

- Padding을 하는 이유

- Filtering 과정에서 생기는 문제를 막기위해

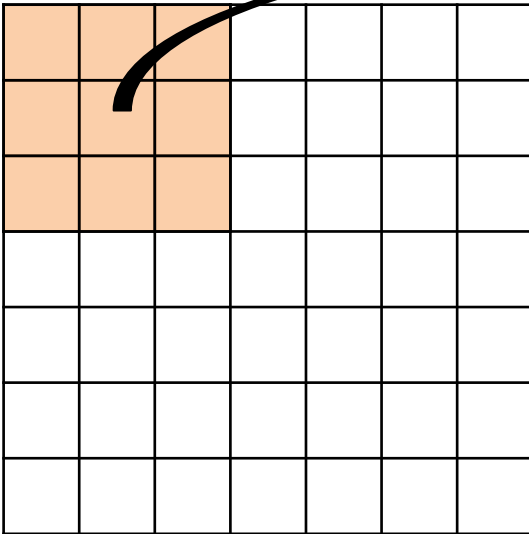
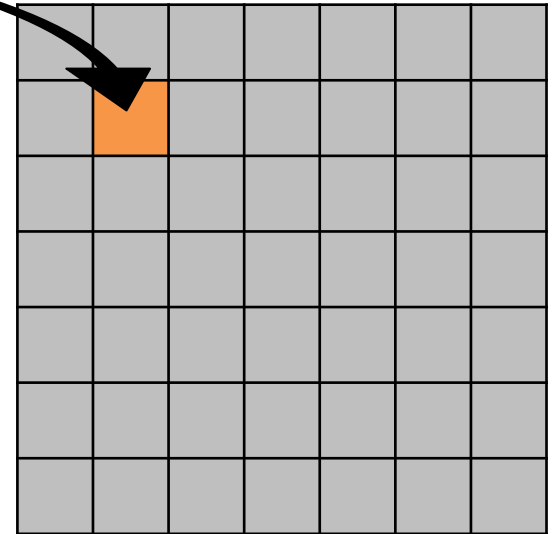


Image filtering



출력 이미지 (7x7)

- **Padding**

- Padding을 하는 이유

- Filtering 과정에서 생기는 문제를 막기위해

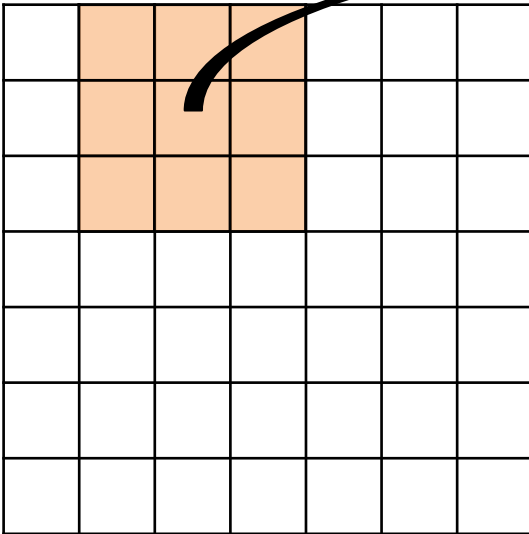
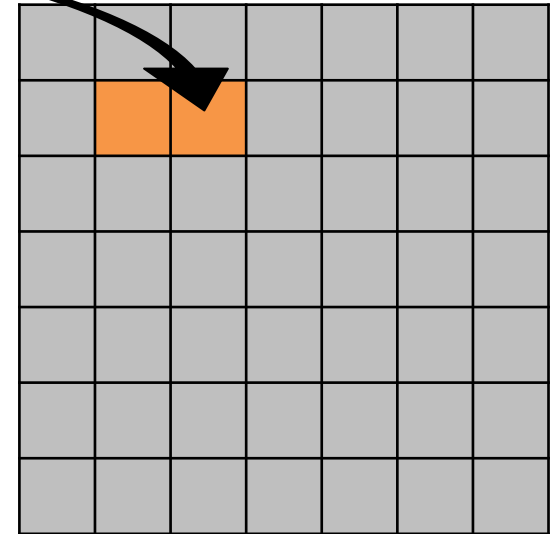


Image filtering



출력 이미지 (7x7)

- **Padding**

- Padding을 하는 이유

- Filtering 과정에서 생기는 문제를 막기위해

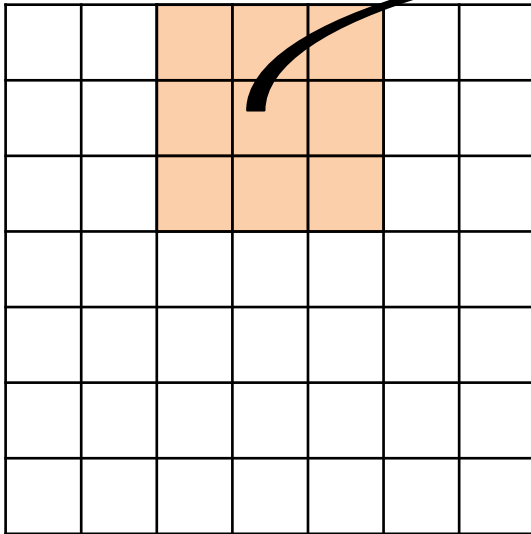
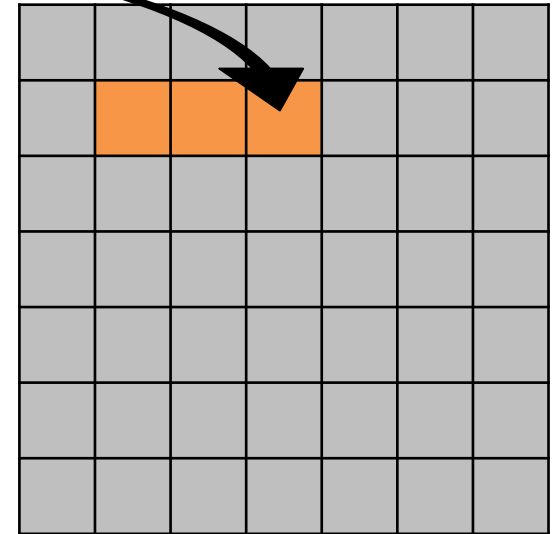


Image filtering



출력 이미지 (7x7)



- **Padding**

- Padding을 하는 이유

- Padding을 하는 경우 이런 문제를 해결할 수 있음

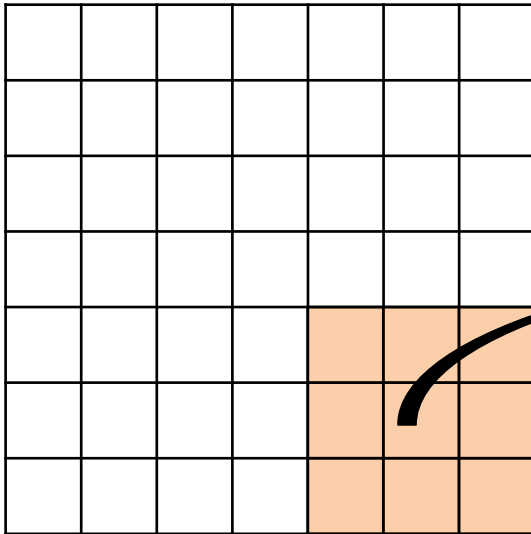
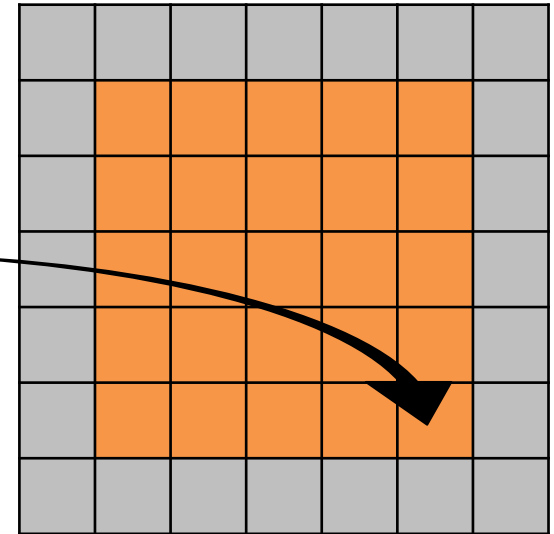
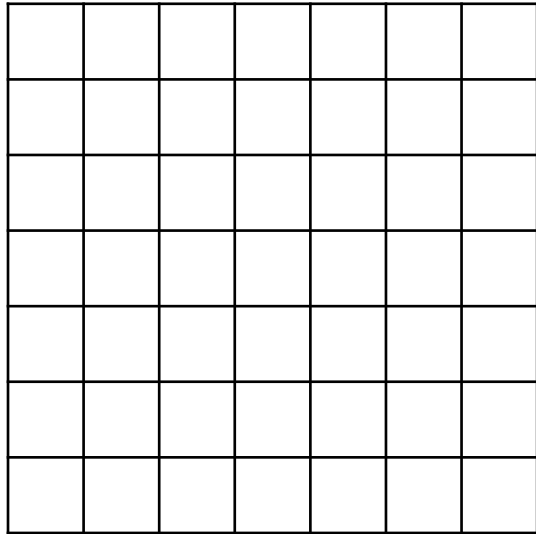


Image filtering

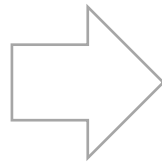


출력 이미지 (7x7)

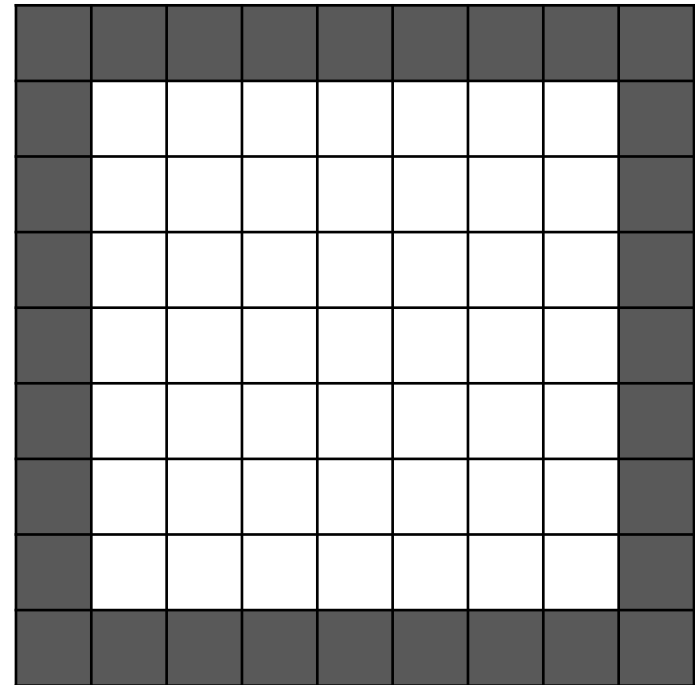
- **Padding**
  - Padding 후 filtering 시



입력 이미지 (7x7)



Zero padding



패딩 이미지 (9x9)

- **Padding**
  - Padding 후 filtering 시

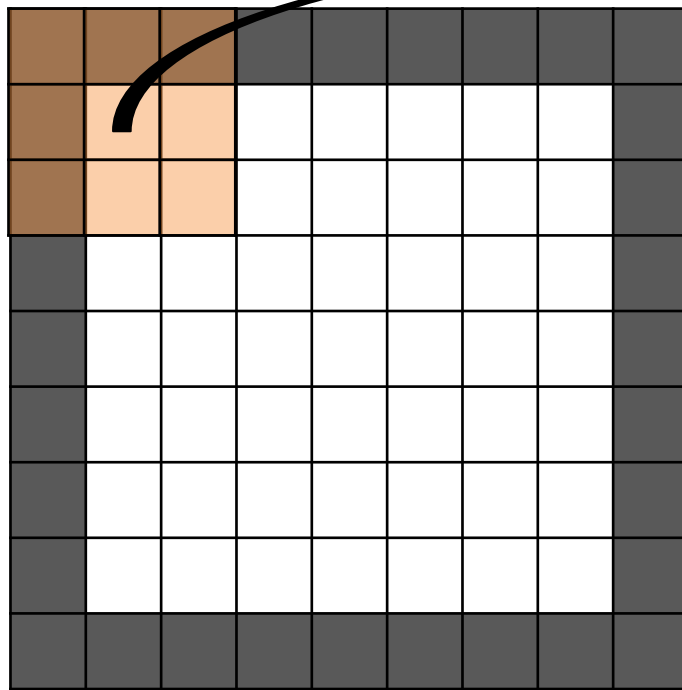
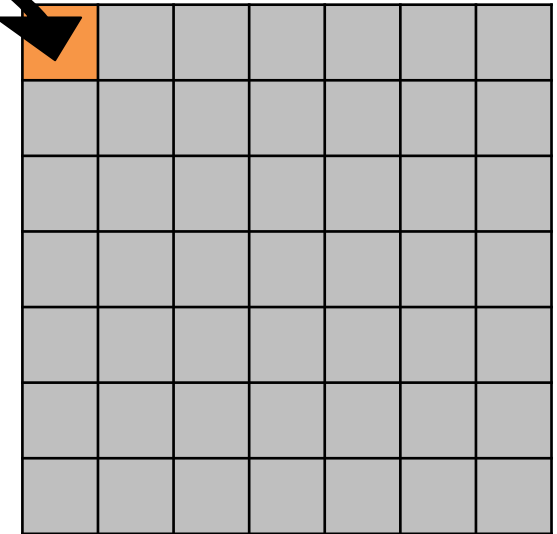


Image filtering



출력 이미지 (7x7)

- **Padding**
  - Padding 후 filtering 시

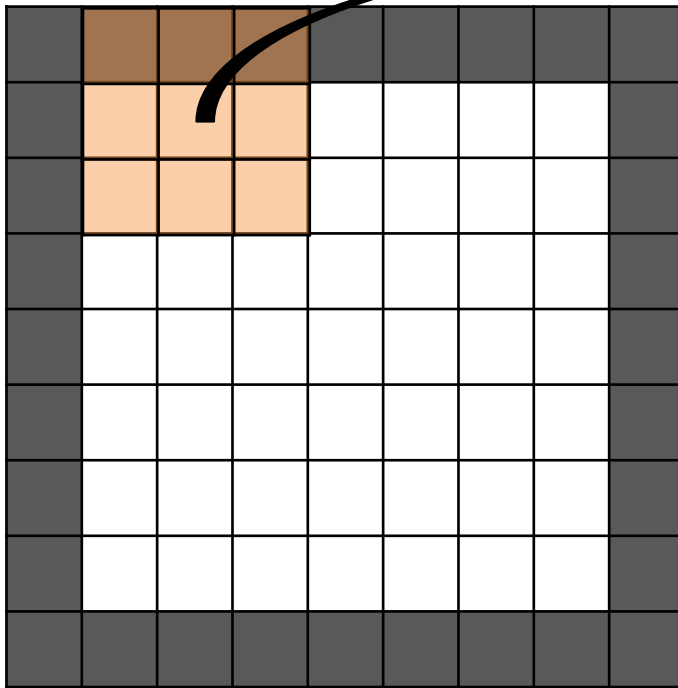
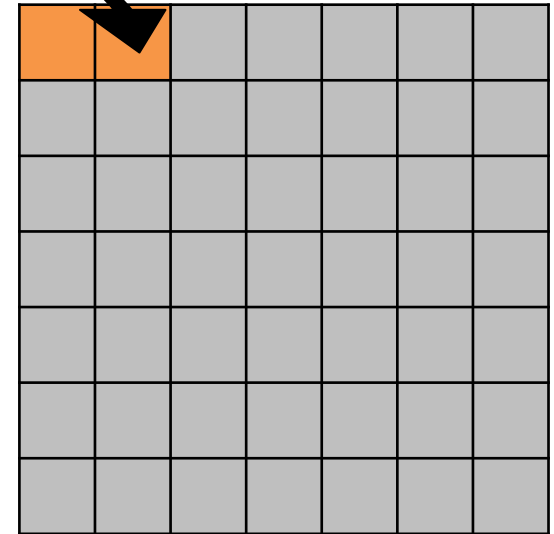


Image filtering



출력 이미지 (7x7)

- **Padding**
  - Padding 후 filtering 시

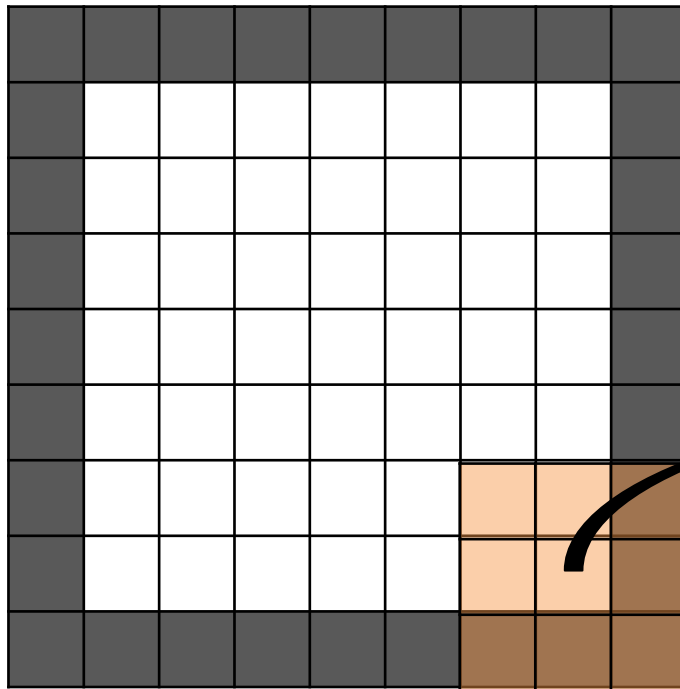
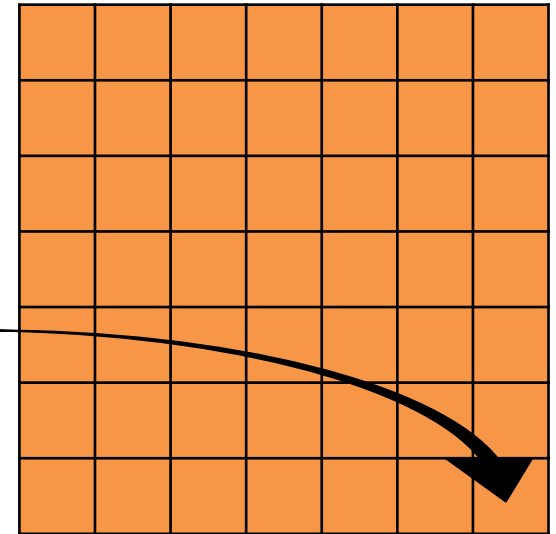


Image filtering



출력 이미지 (7x7)

- **Padding**
  - Zero padding
    - 이미지의 가장자리를 0으로 채움

```
import cv2
import numpy as np

def my_zero_padding(src, pad_shape):
    (h, w) = src.shape
    (p_h, p_w) = pad_shape
    pad_img = np.zeros((h+2*p_h, w+2*p_w))
    pad_img[p_h:p_h+h, p_w:p_w+w] = src
    return pad_img

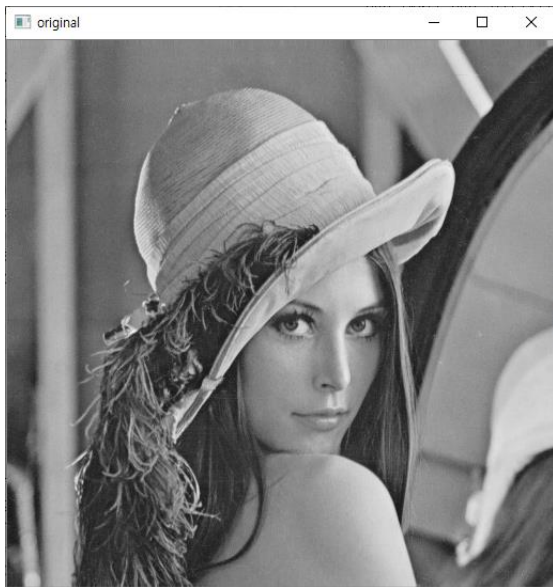
if __name__ == '__main__':
    src = cv2.imread('../Lena.png', cv2.IMREAD_GRAYSCALE)

    pad_src = my_zero_padding(src, (5, 5))

    cv2.imshow('original', src)
    cv2.imshow('pad image', pad_src.astype(np.uint8))
    cv2.waitKey()
    cv2.destroyAllWindows()
```



- **Padding**
  - Zero padding
    - 이미지의 가장자리를 0으로 채움



original



zero padding

# 과제

- **My filtering**

- my\_filtering(src, mask)

- cv2.filter2D를 쓰지 않고 이미지를 필터링하는 것을 구현하기

- Parameter

- » src: 입력 이미지
        - » mask: filter를 의미

- Return

- » dst: filtering이 된 이미지

- my\_mask(ftype, fshape, sigma)

- Average, sharpening, gaussian2D, gaussian1D mask를 출력하는 함수 구현

- Parameter:

- » ftype: 생성할 mask 종류 ex) “average”, “sharpening”
        - » fshape: mask의 크기 ex) (3,3), (5,5)
        - » sigma: default는 1이며, gaussian의 sigma값을 의미

- Return:

- » mask: 생성된 mask를 의미





# 과제

- **My filtering**

- my\_getGaussian2D\_mask(msize, sigma)

- 2D gaussian mask를 생성하는 함수를 구현

- Parameter:

- » msize: gaussian mask의 크기 ex) (3,3), (5,5)

- » Sigma: gaussian sigma 값

- Return:

- » gaus2D: 생성된 2D gaussian filter

- my\_getGaussian1D\_mask(msize, sigma)

- 1D gaussian mask를 생성하는 함수를 구현

- Parameter:

- » msize: gaussian mask의 크기 ex) 3, 5

- » Sigma: gaussian sigma 값

- Return

- » gaus1D: 생성된 1D gaussian filter



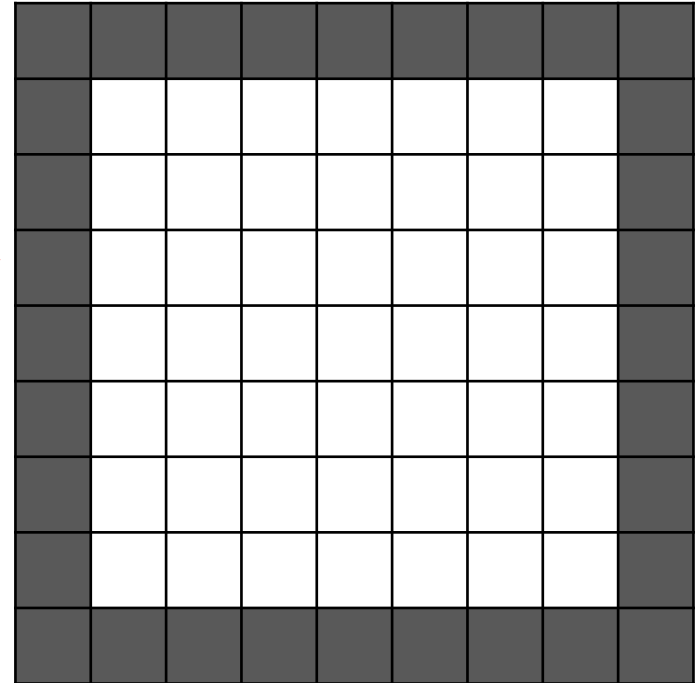
- My filtering
  - my\_filtering(src, mask)

```
def my_filtering(src, mask):
    #####
    # TODO #
    # dst 완성 #
    # dst : filtering 결과 image #
    #####
    h, w = src.shape
    m_h, m_w = mask.shape
    pad_img = my_zero_padding(src, (m_h//2, m_w//2))
    dst = ???

    """
    반복문을 이용하여 filtering을 완성하기
    """
    for row in range(h):
        for col in range(w):
            val = ???
            val = np.clip(val, 0, 255) #범위를 0~255로 조정
            ??? = val

    dst = (dst+0.5).astype(np.uint8) #uint8의 형태로 조정

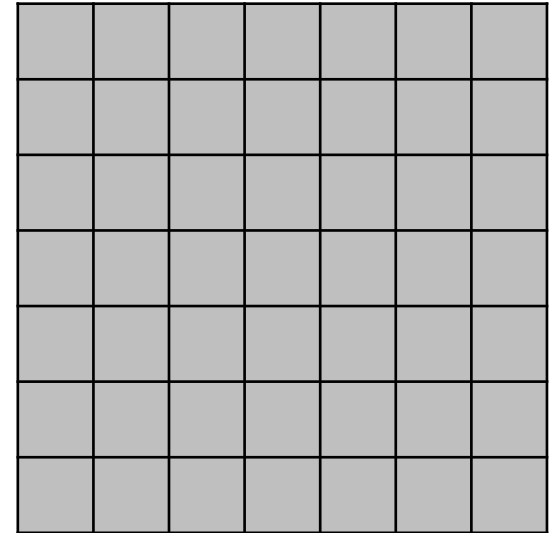
    return dst
```



Pad\_img

- My filtering
  - my\_filtering(src, mask)

```
def my_filtering(src, mask):  
    #####  
    # TODO #  
    # dst 완성 #  
    # dst : filtering 결과 image #  
    #####  
    h, w = src.shape  
    m_h, m_w = mask.shape  
    pad_img = my_zero_padding(src, (m_h//2, m_w//2))  
    dst = ???  
  
    """  
    반복문을 이용하여 filtering을 완성하기  
    """  
    for row in range(h):  
        for col in range(w):  
            val = ???  
            val = np.clip(val, 0, 255) #범위를 0~255로 조정  
            ??? = val  
  
    dst = (dst+0.5).astype(np.uint8) #uint8의 형태로 조정  
  
    return dst
```

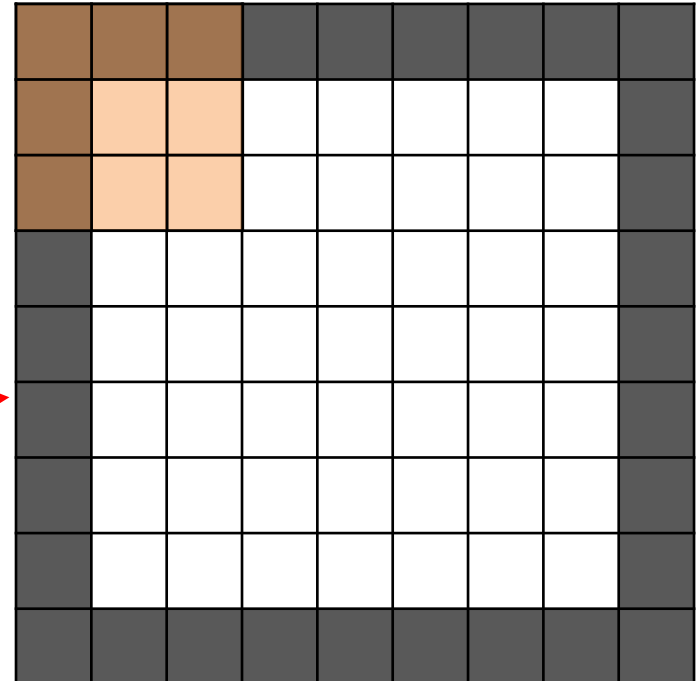


dst



- My filtering
  - my\_filtering(src, mask)

```
def my_filtering(src, mask):  
    #####  
    # TODO #  
    # dst 완성 #  
    # dst : filtering 결과 image #  
    #####  
    h, w = src.shape  
    m_h, m_w = mask.shape  
    pad_img = my_zero_padding(src, (m_h//2, m_w//2))  
    dst = ???  
  
    """  
    반복문을 이용하여 filtering을 완성하기  
    """  
    for row in range(h):  
        for col in range(w):  
            val = ???  
            val = np.clip(val, 0, 255) #범위를 0~255로 조정  
            ??? = val  
  
    dst = (dst+0.5).astype(np.uint8) #uint8의 형태로 조정  
    return dst
```



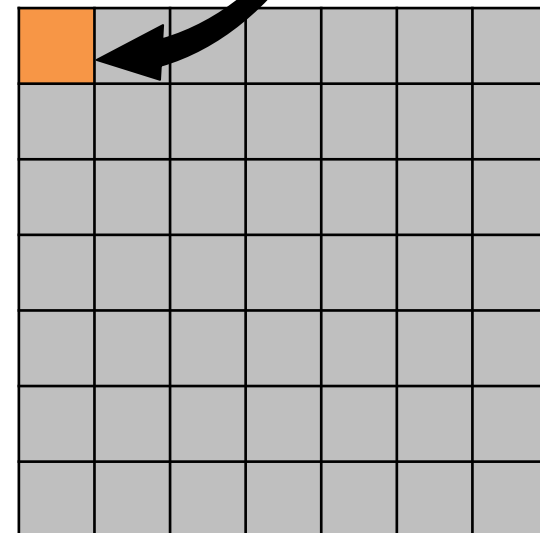
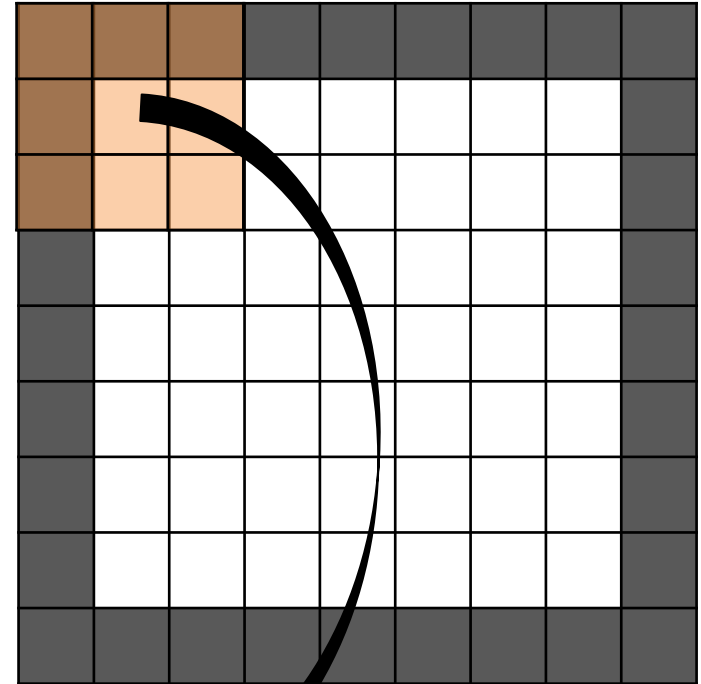
- My filtering
  - my\_filtering(src, mask)

```
def my_filtering(src, mask):
    #####
    # TODO #
    # dst 완성 #
    # dst : filtering 결과 image #
    #####
    h, w = src.shape
    m_h, m_w = mask.shape
    pad_img = my_zero_padding(src, (m_h//2, m_w//2))
    dst = ???

    """
    반복문을 이용하여 filtering을 완성하기
    """
    for row in range(h):
        for col in range(w):
            val = ???
            val = np.clip(val, 0, 255) #범위를 0~255로 조정
            ??? = val

    dst = (dst+0.5).astype(np.uint8) #uint8의 형태로 조정

    return dst
```



# 과제

## • My filtering

– my\_mask(ftype, fshape, sigma)

```
def my_mask(ftype, fshape, sigma=1):
    if ftype == 'average':
        print('average filtering')
        #####
        # TODO #
        # mask 완성 #
        #####
        mask = ???
        mask = ???

        #mask 확인
        print(mask)

    elif ftype == 'sharpening':
        print('sharpening filtering')
        #####
        # TODO #
        # mask 완성 #
        #####

        base_mask = np.zeros(fshape)
        base_mask[fshape[0]//2, fshape[1]//2] = 2
        aver_mask = ???
        aver_mask = ???
        mask = ???
```

```
elif ftype == 'gaussian2D':
    print('gaussian filtering')
    #####
    # TODO #
    # mask 완성 #
    #####
    mask = my_get_Gaussian2D_mask(fshape, sigma=sigma)
    #mask 확인
    print(mask)
```

```
elif ftype == 'gaussian1D':
    print('gaussian filtering')
    #####
    # TODO #
    # mask 완성 #
    #####
    mask = my_get_Gaussian1D_mask(fshape, sigma=sigma)
    #mask 확인
    print(mask)
```

```
return mask
```



- **My filtering**

- `my_mask(fstype, fshape, sigma)`

- Average filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3x3

1/15	1/15	1/15	1/15	1/15
1/15	1/15	1/15	1/15	1/15
1/15	1/15	1/15	1/15	1/15

3x5

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

5x5



# 과제

## • My filtering

– my\_mask(ftype, fshape, sigma)

- Sharpening filter

0	0	0
0	2	0
0	0	0

 $-$ 

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

 $=$ 

-1/9	-1/9	-1/9
-1/9	17/9	-1/9
-1/9	-1/9	-1/9

3x3

0	0	0	0	0
0	0	0	0	0
0	0	2	0	0
0	0	0	0	0
0	0	0	0	0

 $-$ 

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

 $=$ 

-1/25	-1/25	-1/25	-1/25	-1/25
-1/25	-1/25	-1/25	-1/25	-1/25
-1/25	-1/25	49/25	-1/25	-1/25
-1/25	-1/25	-1/25	-1/25	-1/25
-1/25	-1/25	-1/25	-1/25	-1/25

5x5





# 과제

- My filtering

- my\_getGaussian2D\_mask(msize, sigma)

```
def my_get_Gaussian2D_mask(msize, sigma=1):
    #####
    # ToDo
    # 2D gaussian filter 만들기
    #####
    y, x = np.mgrid[???:???, ???:???]
    ...

    y, x = np.mgrid[-1:2, -1:2]
    y = [[-1, -1, -1],
         [ 0,  0,  0],
         [ 1,  1,  1]]
    x = [[-1,  0,  1],
         [-1,  0,  1],
         [-1,  0,  1]]
    ...

    # 2차 gaussian mask 생성
    gaus2D = ???
    # mask의 총 합 = 1
    gaus2D /= ???

    return gaus2D
```



# 과제


## • My filtering

– my\_getGaussian2D\_mask(msize, sigma)

- 5×5 gaussian filter

–  $\sigma = 1$

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



(-2,-2)	(-2,-1)	(-2,0)	(-2,1)	(-2,2)
(-1,-2)	(-1,-1)	(-1,0)	(-1,1)	(-1,2)
(0,-2)	(0,-1)	(0,0)	(0,1)	(0,2)
(1,-2)	(1,-1)	(1,0)	(1,1)	(1,2)
(2,-2)	(2,-1)	(2,0)	(2,1)	(2,2)

$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^0$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$

# 과제

## • My filtering

– my\_getGaussian2D\_mask(msize, sigma)

- 5×5 gaussian filter

–  $\sigma = 1$

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

(-2,-2)	(-2,-1)	(-2,0)	(-2,1)	(-2,2)
(-1,-2)	(-1,-1)	(-1,0)	(-1,1)	(-1,2)
(0,-2)	(0,-1)	(0,0)	(0,1)	(0,2)
(1,-2)	(1,-1)	(1,0)	(1,1)	(1,2)
(2,-2)	(2,-1)	(2,0)	(2,1)	(2,2)

$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^0$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$

# 과제

## • My filtering

- my\_getGaussian2D\_mask(msize, sigma)
  - Sum: 구한 gaussian mask의 총합

$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^0$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$



$\frac{1}{sum}$

$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^0$	$\frac{1}{2\pi} e^{-\frac{1+0}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+0}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+1}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+1}{2}}$
$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{0+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{1+4}{2}}$	$\frac{1}{2\pi} e^{-\frac{4+4}{2}}$

총 합을 1로 만들기



# 과제

## • My filtering


- my\_getGaussian2D\_mask(msize, sigma)
  - np.mgrid를 이용하여 각각 값을 채움
    - msize와 관계를 생각하여 구현

```
def my_get_Gaussian2D_mask(msize, sigma=1):
    #####
    # ToDo
    # 2D gaussian filter 만들기
    #####
    y, x = np.mgrid[???:???, ???:???]
    ...

    y, x = np.mgrid[-1:2, -1:2]
    y = [[-1,-1,-1],
         [ 0, 0, 0],
         [ 1, 1, 1]]
    x = [[-1, 0, 1],
         [-1, 0, 1],
         [-1, 0, 1]]
    ...

    # 2차 gaussian mask 생성
    gaus2D = ???
    # mask의 총 합 = 1
    gaus2D /= ???

    return gaus2D
```



(-2,-2)	(-2,-1)	(-2,0)	(-2,1)	(-2,2)
(-1,-2)	(-1,-1)	(-1,0)	(-1,1)	(-1,2)
(0,-2)	(0,-1)	(0,0)	(0,1)	(0,2)
(1,-2)	(1,-1)	(1,0)	(1,1)	(1,2)
(2,-2)	(2,-1)	(2,0)	(2,1)	(2,2)



# 과제

## • My filtering

- my\_getGaussian2D\_mask(msize, sigma)
  - e: np.exp를 사용,  $\pi$ : np.pi 사용
    - np.exp(2) =>  $e^2$

```
def my_get_Gaussian2D_mask(msize, sigma=1):
    #####
    # ToDo
    # 2D gaussian filter 만들기
    #####
    y, x = np.mgrid[???:???, ???:???]
    ...

    y, x = np.mgrid[-1:2, -1:2]
    y = [[-1, -1, -1],
         [ 0,  0,  0],
         [ 1,  1,  1]]
    x = [[-1,  0,  1],
         [-1,  0,  1],
         [-1,  0,  1]]
    ...

    # 2차 gaussian mask 생성
    gaus2D = ???
    # mask의 총 합 = 1
    gaus2D /= ???

    return gaus2D
```

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



# 과제

- My filtering

- my\_getGaussian1D\_mask(msize, sigma)

```
def my_get_Gaussian1D_mask(msize, sigma=1):  
    #####  
    # ToDo  
    # 1D gaussian filter 만들기  
    #####  
    x = np.full((1, msize), [range(-(msize // 2), (msize // 2) + 1)])  
    '''  
    x = np.full((1, 3), [-1, 0, 1])  
    x = [[ -1, 0, 1]]  
  
    x = np.array([[ -1, 0, 1]])  
    x = [[ -1, 0, 1]]  
    '''  
  
    gaus1D = ???  
  
    # mask의 총 합 = 1  
    gaus1D /= ???  
    return gaus1D
```



# 과제

## • My filtering

– my\_getGaussian1D\_mask(msize, sigma)

$$G(x) = \left( \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \right)$$

- 2차 Gaussian 식을, 각 축에 대해서 분리한 식.
- 1x5 Gaussian kernel

$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-4}{2}}$	$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}}$	$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-0}{2}}$	$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}}$	$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-4}{2}}$
--	--	--	--	--

$$G(y) = \left( \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-y^2}{2\sigma^2}} \right)$$

- 5x1 Gaussian kernel

$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-4}{2}}$
$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}}$
$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-0}{2}}$
$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}}$
$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-4}{2}}$





# 과제

- **My filtering**

- `my_getGaussian1D_mask(msize, sigma)`

The filter factors  
into a product of 1D  
filters:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

```
# Gaussian filter
gaussian2d_mask = my_mask('gaussian2D', (5, 5), sigma=1)
gaussian1d_mask = my_mask('gaussian1D', 5, sigma=1)

dst_gaussian2d = my_filtering(src, gaussian2d_mask)

dst_gaussian1d = my_filtering(src, gaussian1d_mask.T)
dst_gaussian1d = my_filtering(dst_gaussian1d, gaussian1d_mask)
```



# 과제

## • 제출

강의실 홈

강의정보 ▲

· 강의계획서

성적/출석관리 ▲

· 온라인출석부

수강생 알림 ▼

기타 관리 ▼

교수화면 보기

학습활동

과제

파일

동영상

게시판

강의 개요

공지사항

강좌 Q&A

1주차 학습자료

이번주 강의

3주차 [3월16일 - 3월22일]

[3-1] Image filtering (1) 2023-03-16 00:00:00 ~ 2023-03-22 23:59:59 (지각 : 2023-03-22 23:59:59), 32:42

[3-2] Image filtering (2) 2023-03-16 00:00:00 ~ 2023-03-22 23:59:59 (지각 : 2023-03-22 23:59:59), 35:51



## 강의실 홈

### 강의정보 ▲

· 강의계획서

### 성적/출석관리 ▲

· 온라인출석부

수강생 알림 ▼

기타 관리 ▼

교수화면 보기

## 학습활동

📅 과제

📁 파일

▶ 동영상

🗨️ 게시판

🏠 > 영상처리(2023년도, ...

## 과제

주	과제	종료 일시	제출	성적
2주차 [3월9일 - 3월15일]	영상처리 2주차 실습 과제	2023-03-24 23:55	미제출	-



# 과제

강의실 홈

## • 제출

강의정보 ▾

· 강의계획서

성적/출석관리 ▴

· 온라인출석부

수강생 알림 ▾

기타 관리 ▾

교수화면 보기

학습활동

+

🏠 > 영상처리(2023년도, ... > 2주차 [3월9일 - 3월... > 영상처리 2주차 실습 ...

## 영상처리 2주차 실습 과제

보고서(PDF)와 Python 파일을 압축하여 제출하세요.

자세한 내용은 실습자료를 확인바랍니다.

### 제출 상황

제출 여부	제출 안 함
채점 상황	채점되지 않음
종료 일시	2023-03-24 23:55
마감까지 남은 기한	8 일 6 시간
최종 수정 일시	-
제출물 설명	▶ 댓글 (0)

과제 제출하기



# 과제

강의실 홈

## 제출

강의계획서

성적/출석관리 ▲

온라인출석부

수강생 알림 ▼

기타 관리 ▼

교수화면 보기

학습활동



🏠 > 영상처리(2023년도, ... > 2주차 [3월9일 - 3월... > 영상처리 2주차 실습 ...

## 영상처리 2주차 실습 과제

보고서(PDF)와 Python 파일을 압축하여 제출하세요.

자세한 내용은 실습자료를 확인바랍니다.

모든 업로드된 파일은 표절검사를 받게 됩니다.

첨부파일

파일의 최대 크기: 1GB, 최대 첨부 파일 갯수: 1

▶ 📁 파일

첨부파일을 마우스로 끌어 놓으세요.

저장

취소

[저장] 버튼을 클릭 시 최종 과제 제출일이 업데이트 되오니 주의하세요.



# 과제


## • 제출

### 영상처리 2주차 실습 과제

보고서(PDF)와 Python 파일을 압축하여 제출하세요.

자세한 내용은 실습자료를 확인바랍니다.

### 제출 상황

제출 여부	제출 완료
채점 상황	채점되지 않음
종료 일시	2023-03-24 23:55
마감까지 남은 기한	8 일 6 시간
최종 수정 일시	2023-03-16 17:26
첨부파일	 <a href="#">Lena.png</a>
제출물 설명	▶ 댓글 (0)

제출한 과제 편집



# 과제

- 제출

- 파일

- Python + 보고서 압축(zip) 파일
      - PDF포맷의 보고서 및 사용한 py파일 전부 제출
    - [IP]202201230\_홍길동\_3주차\_과제.zip으로 제출

- 보고서

- [IP]202201230\_홍길동\_3주차\_과제.pdf
    - PDF 파일 형식으로 제출 (파일 형식이 안 맞는 경우 감점)
    - 보고서 양식 사용



# 과제

## • 기한

- 실습 동영상에 올라온 날부터 2주간
  - 03월 17일 ~ 03월 31일 23시 55분까지
  - 추가 제출 기한 없음
  
- 채점(10점)
  - 미 제출, copy 시 0점
  - 제출 시 최소 점수: 1점
  - 감점요인
    - 보고서 미 제출 (-3점)
    - 보고서 PDF 포맷이 아닌 경우 (-1점)
    - 파일 명을 지키지 않은 경우 (-1점)
    - 빈 코드를 그대로 제출하는 경우 (-6점)
    - 구현을 못하거나, 잘못 구현한 경우
    - 보고서 내용이 부실한 경우

