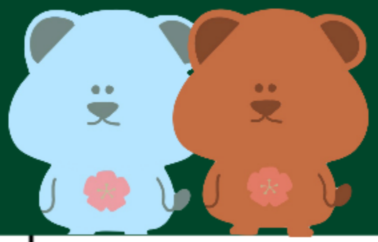
컴퓨터공학과 이주원(2276242)



15주차 발표

컴퓨터비전 ~ 입체형 이미지에서의 합성곱

EWHA EURON

목차

#01 Computer Visions

#02 CNN

#03 Convolutions over Volumes

#04 퀴즈 리뷰

1.1 Computer Vision Problems

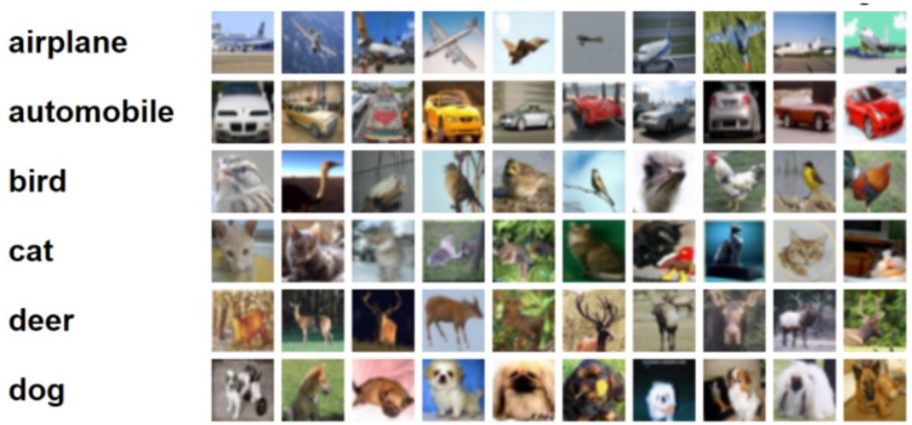
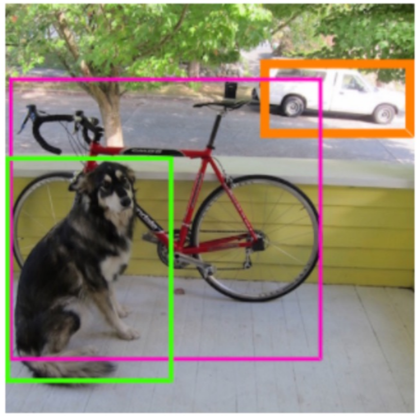


Image Classification



Object detection

content image

style image

generated image

Ancient city of Persepolis

The Starry Night (Van Gogh)

Persepolis
in Van Gogh style

Neural Style Transfer

1.1 Computer Vision Problems

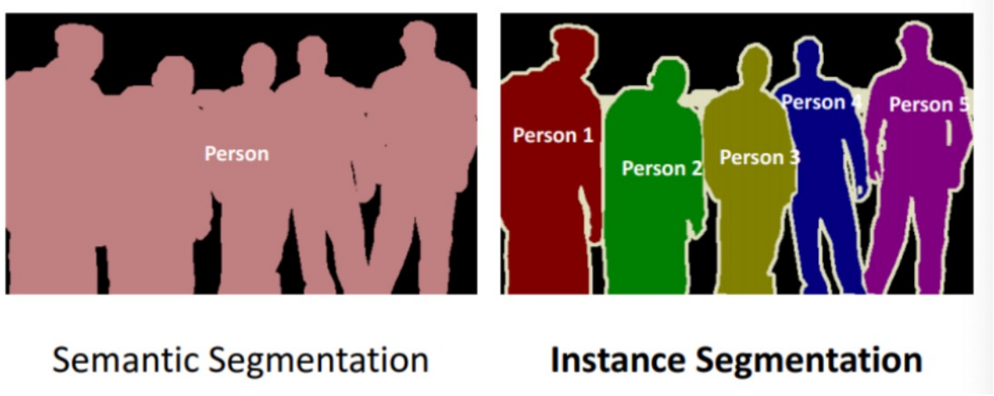


Image Segmentation

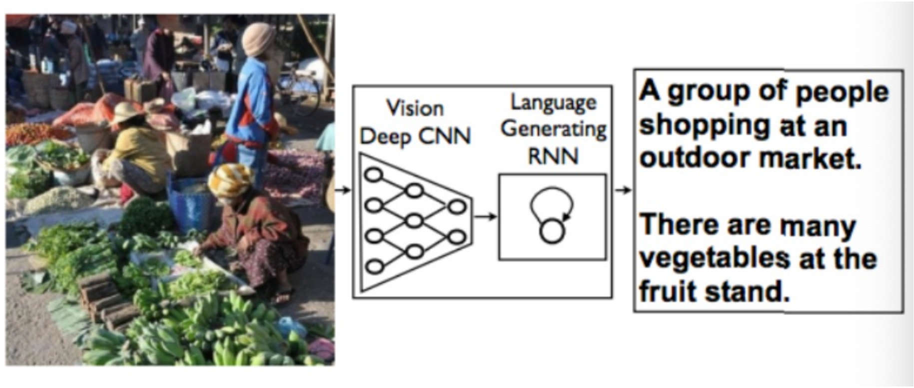
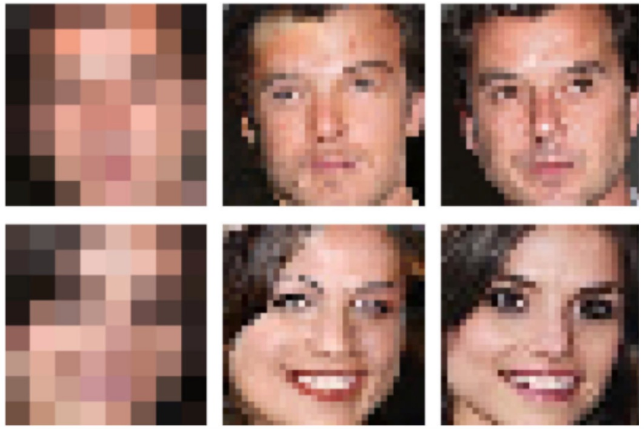
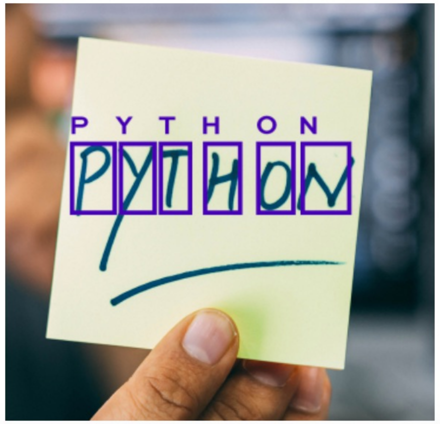


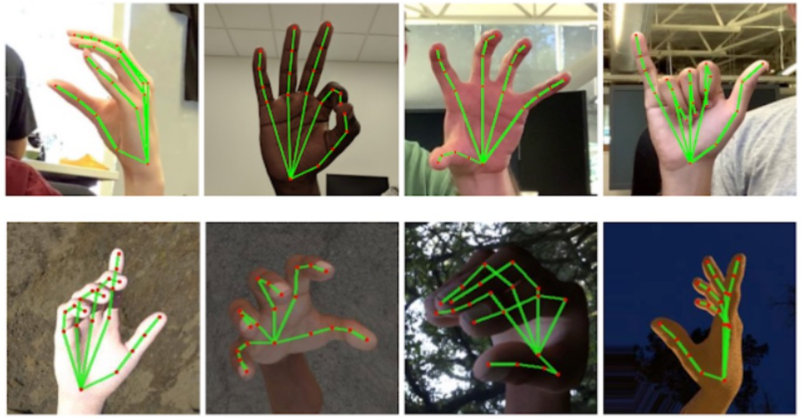
Image Captioning



Super Resolution

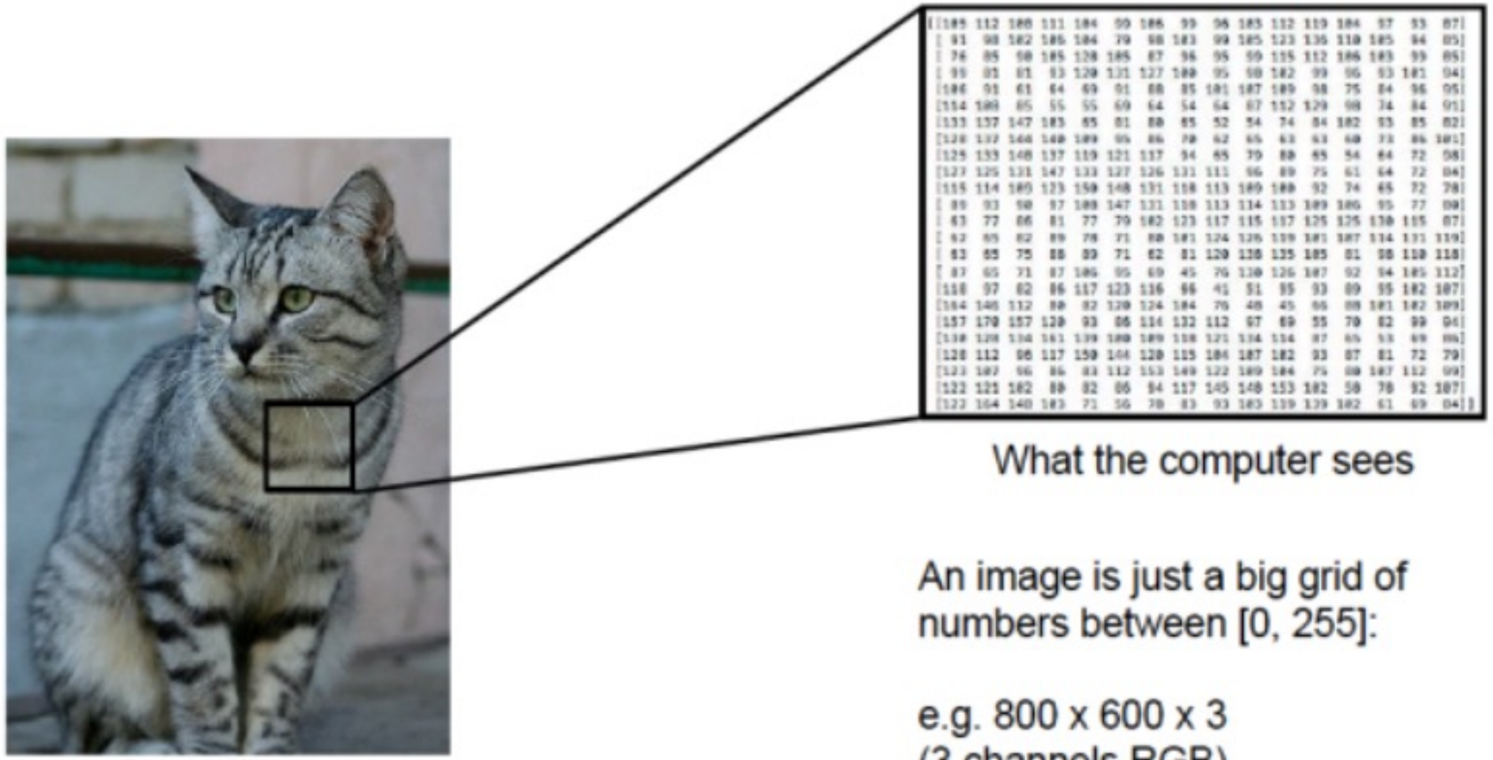


Text Detection & OCR



Keypoint Detection

1.2 Deep Learning on Large images

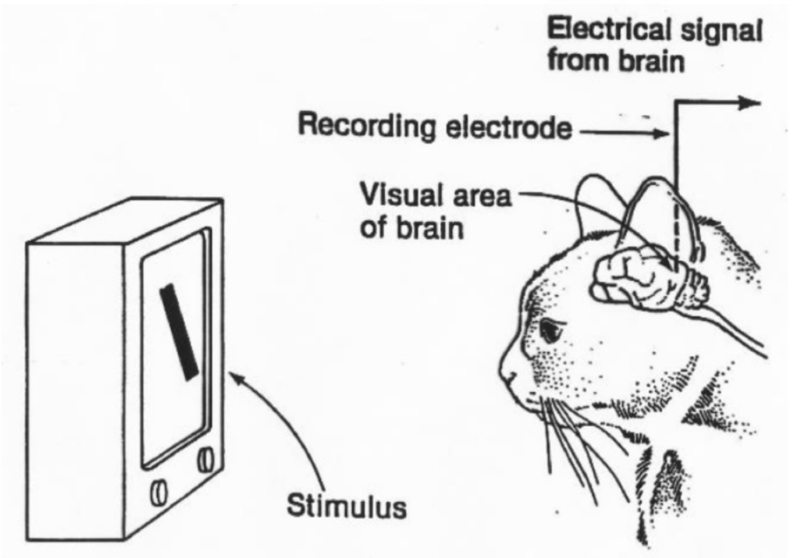


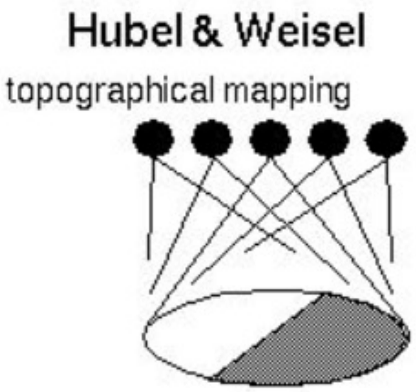
What the computer sees

An image is just a big grid of numbers between [0, 255]:

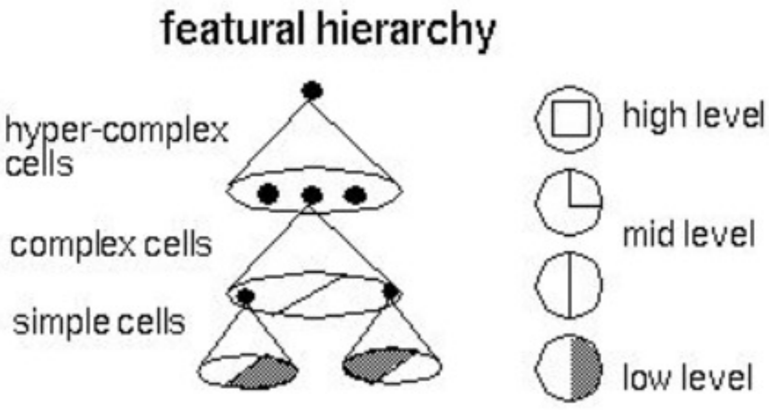
e.g. 800 x 600 x 3
(3 channels RGB)

2.1 CNN Introduction





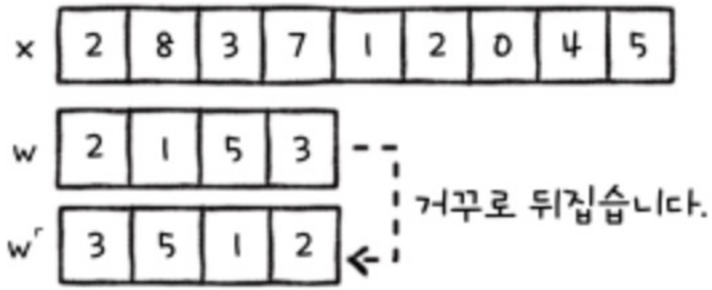
Topographical Mapping

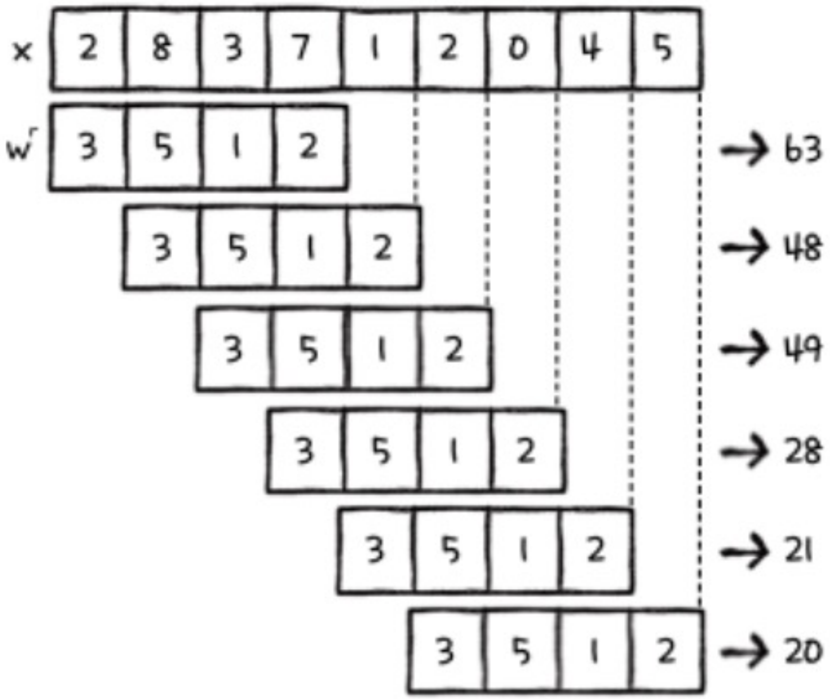


Hierarchical Organization

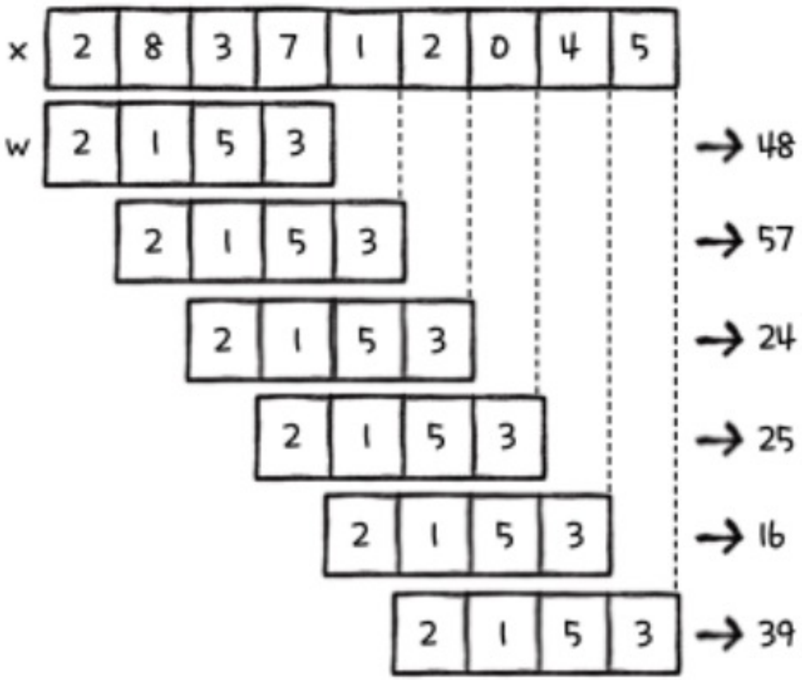
CNN

2.2 CNN - Convolution





합성곱



교차상관

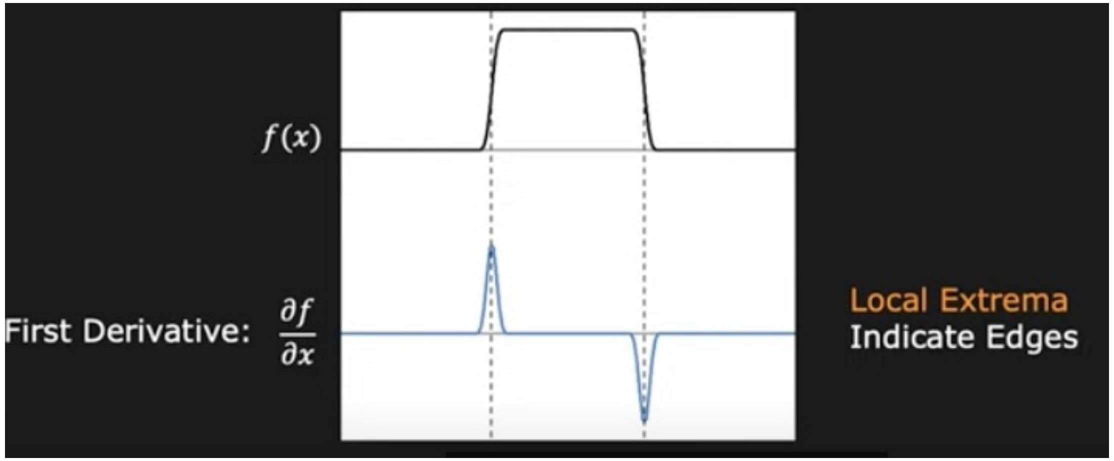
2.2 CNN - Convolution

$$\overset{6 \times 6}{\begin{bmatrix} 3 & 0^{1} & 1^{0} & 2^{-1} & 7 & 4 \\ 1 & 5^{1} & 8^{0} & 9^{-1} & 3 & 1 \\ 2 & 7^{1} & 2^{0} & 5^{-1} & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 4 & 2 & 5 & 2 & 3 & 9 \end{bmatrix}} * \overset{3 \times 3}{\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}} = \overset{4 \times 4}{\begin{bmatrix} -5 & -4 & & \\ & & & \\ & & & \\ & & & \end{bmatrix}}$$

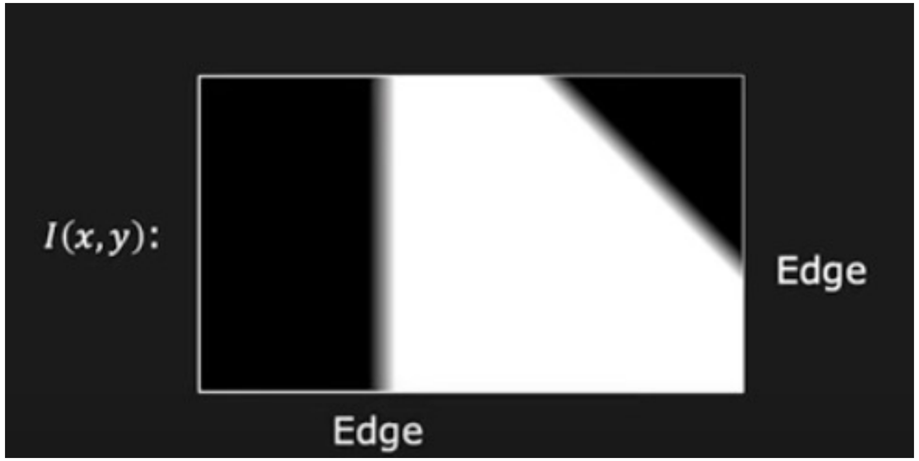
$$0 \times 1 + 1 \times 0 + 2 \times -1 + 5 \times 1 + 8 \times 0 + 9 \times -1 + 7 \times 1 + 2 \times 0 + 5 \times -1 = -4$$

2.3 CNN Filter - Edge Detection

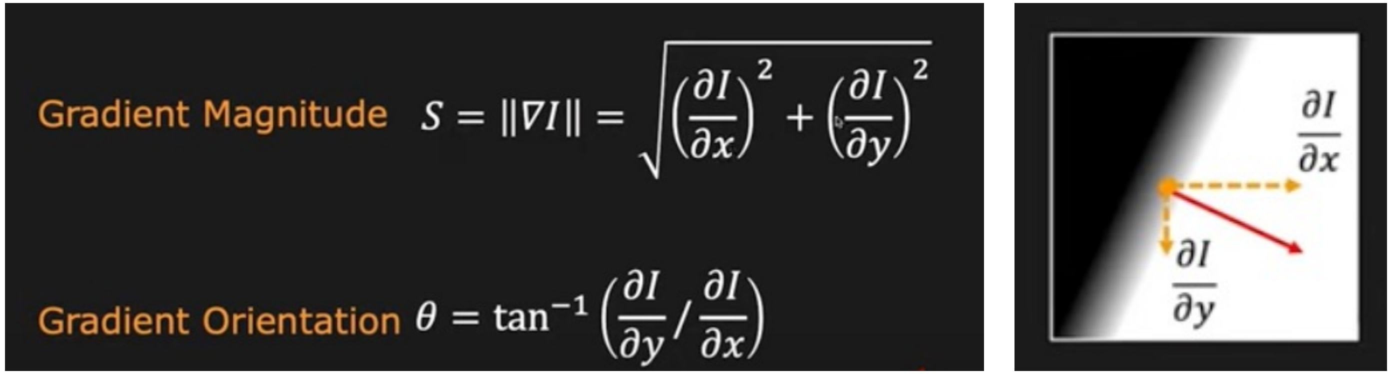
수학적 이해



1D Edge Detection



2D Edge Detection



Gradient as Edge Detector

2.3 CNN Filter - Edge Detection

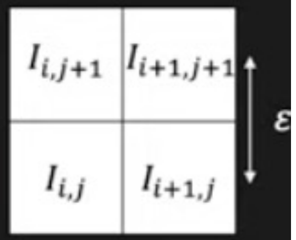
$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

유한 차분 근사 기법

Finite difference approximations:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon}\Big(\big(I_{i+1,j+1} - I_{i,j+1}\big) + \big(I_{i+1,j} - I_{i,j}\big)\Big)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon}\Big(\big(I_{i+1,j+1} - I_{i+1,j}\big) + \big(I_{i,j+1} - I_{i,j}\big)\Big)$$



Can be implemented as Convolution!

$$\frac{\partial}{\partial x} \approx \frac{1}{2\varepsilon}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \frac{\partial}{\partial y} \approx \frac{1}{2\varepsilon}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

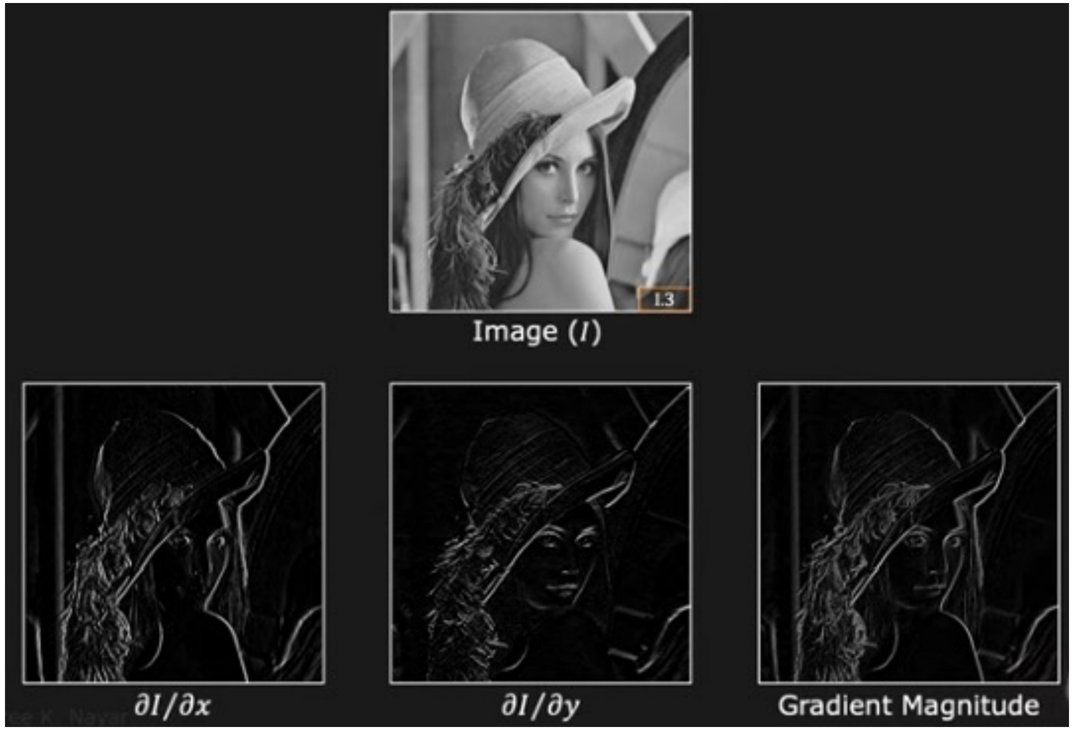
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

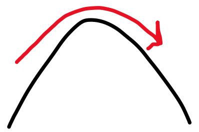
Prewitt Filter

2.3 CNN Filter - Edge Detection

Gradient	Roberts	Prewitt	Sobel (3x3)
$\frac{\partial I}{\partial x}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
$\frac{\partial I}{\partial y}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$



2.3 CNN Filter - Edge Detection



Vertical edge detection

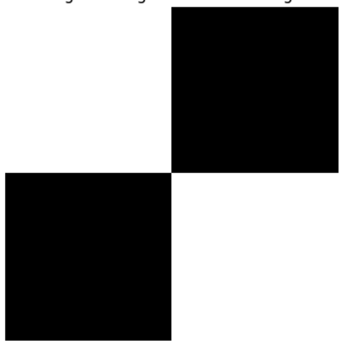
$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \end{bmatrix}$$

2.3 CNN Filter – Edge Detection

Vertical and Horizontal edge detection

Original Image with Vertical Edge



10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

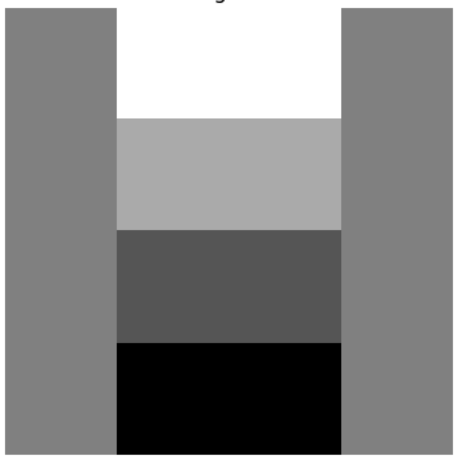
$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Vertical

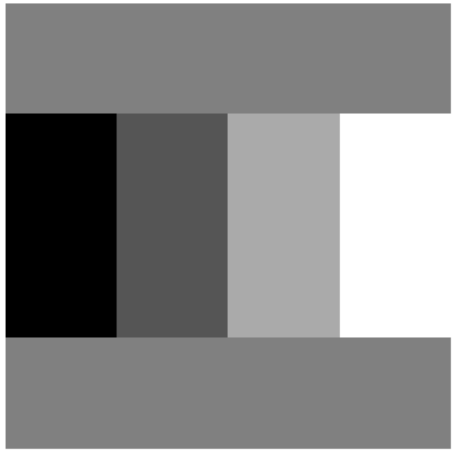
$$* \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Horizontal

Vertical Edge Detection



Horizontal Edge Detection



2.3 CNN Filter - Edge Detection

다양한 필터들

1	0	-1
1	0	-1
1	0	-1

vertical

1	0	-1
2	0	-2
1	0	-1

sobel

3	0	-3
10	0	-10
3	0	-3

scharr

1	1	1
0	0	0
-1	-1	-1

horizontal

1	2	1
0	0	0
-1	-2	-1

sobel

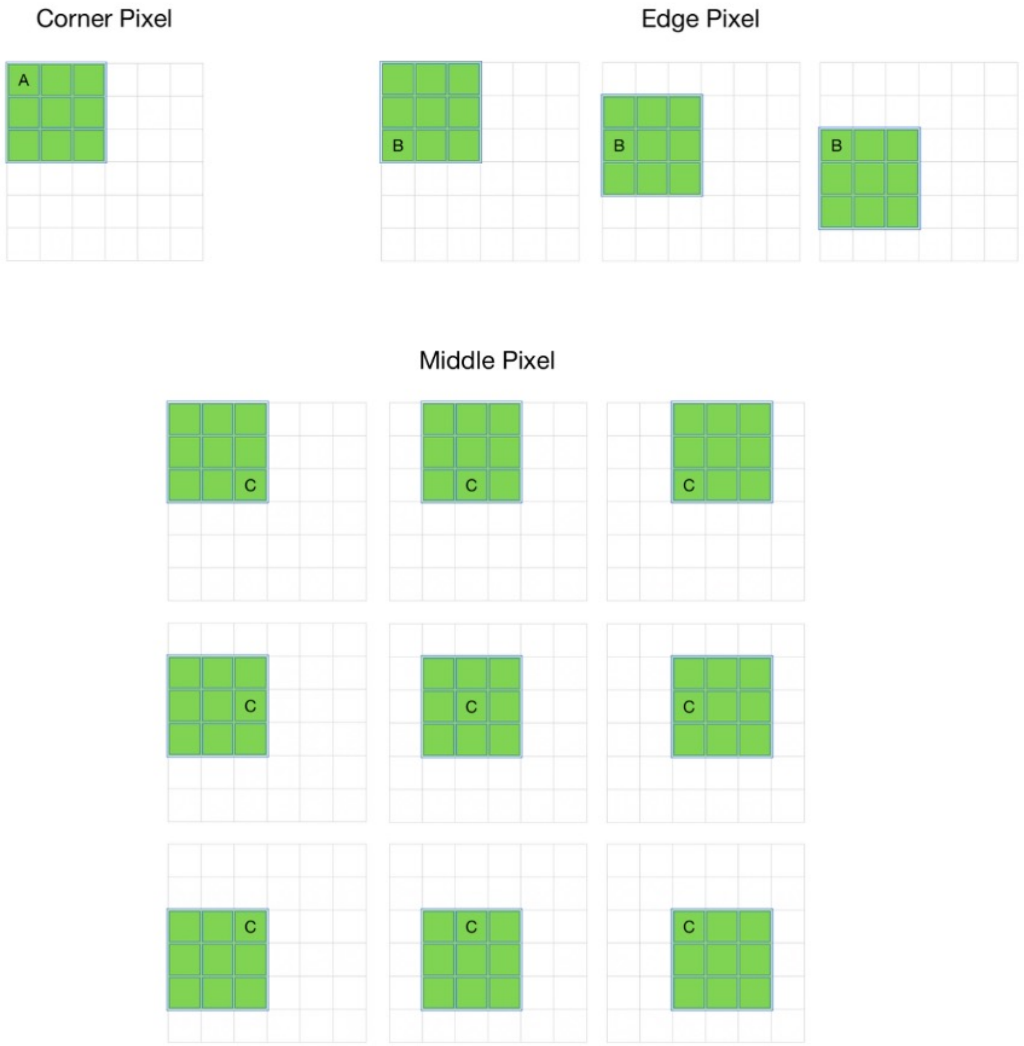
3	10	3
0	0	0
-3	-10	-3

scharr

적합한 필터 탐색

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

2.4 CNN – Padding



< 기존 방식 단점 >

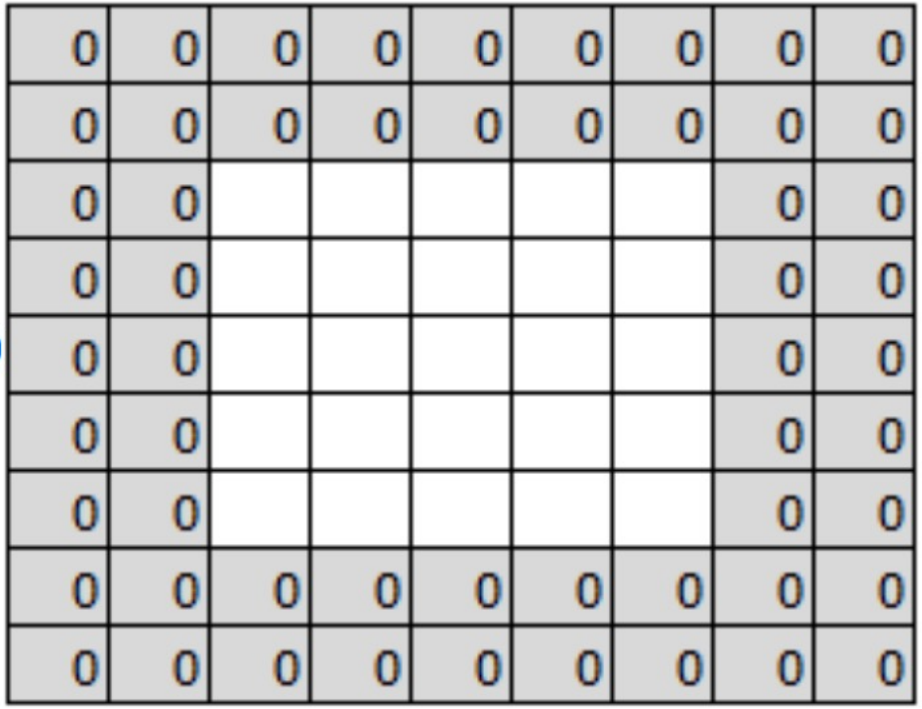
1) 합성곱 연산을 반복하면 이미지가 축소

2) 가장 자리 픽셀 -> 단 한번 사용

Padding

2.4 CNN – Padding

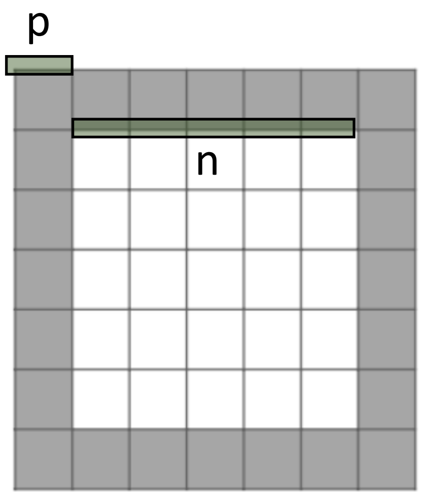
zero padding of size 2



Padding의 종류

1. Valid Padding : padding하지 않는 것

2. Same Padding : output image 크기 = input image의 크기



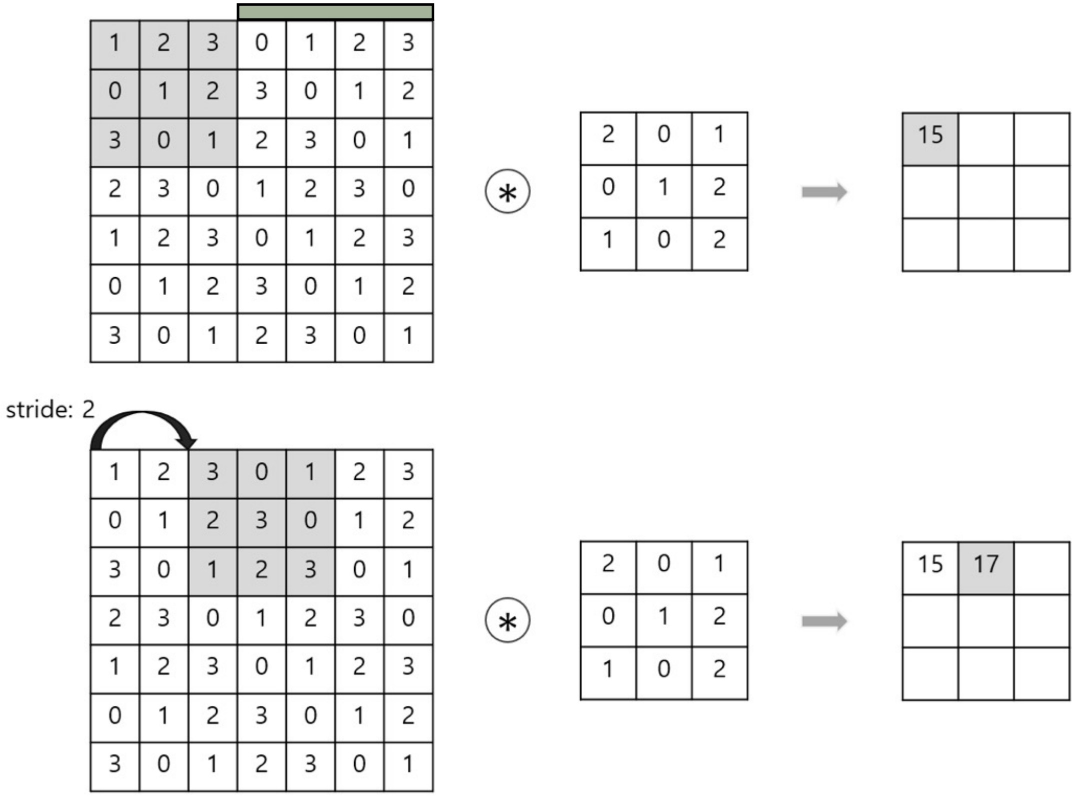
$$n = n + 2p - f + 1$$

$$p = \frac{f-1}{2}$$

n: 입력 이미지 한변 길이
f: 필터의 크기
p: 패딩의 크기

2.5 CNN - Stride

Stride = 2인 합성곱 연산

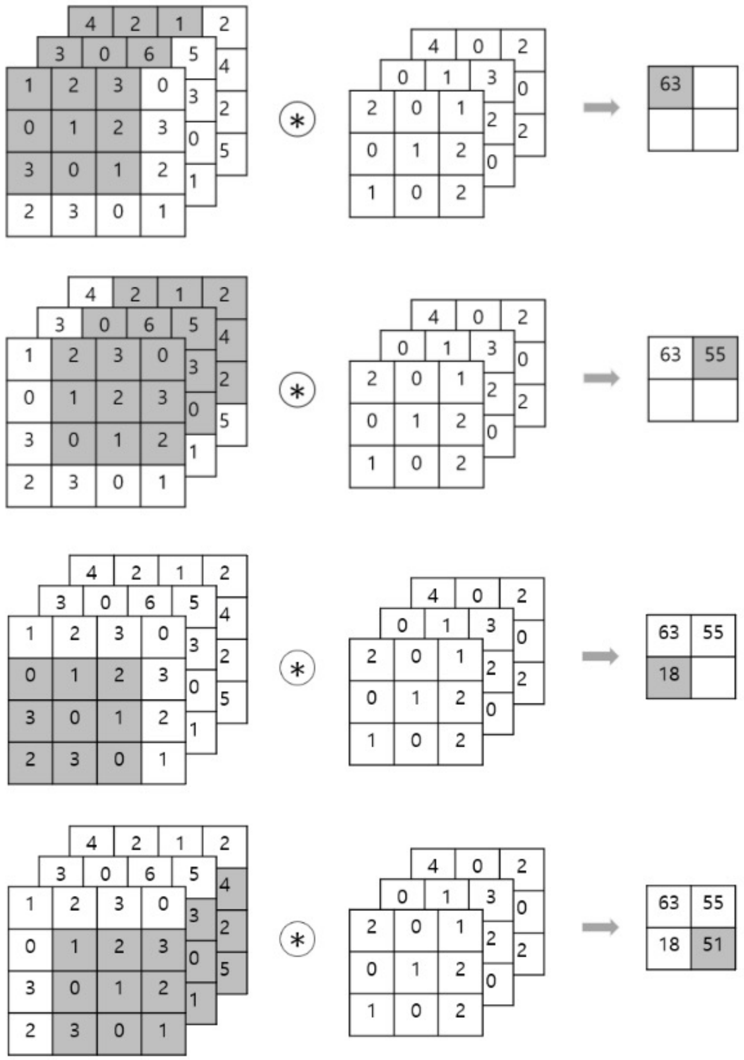


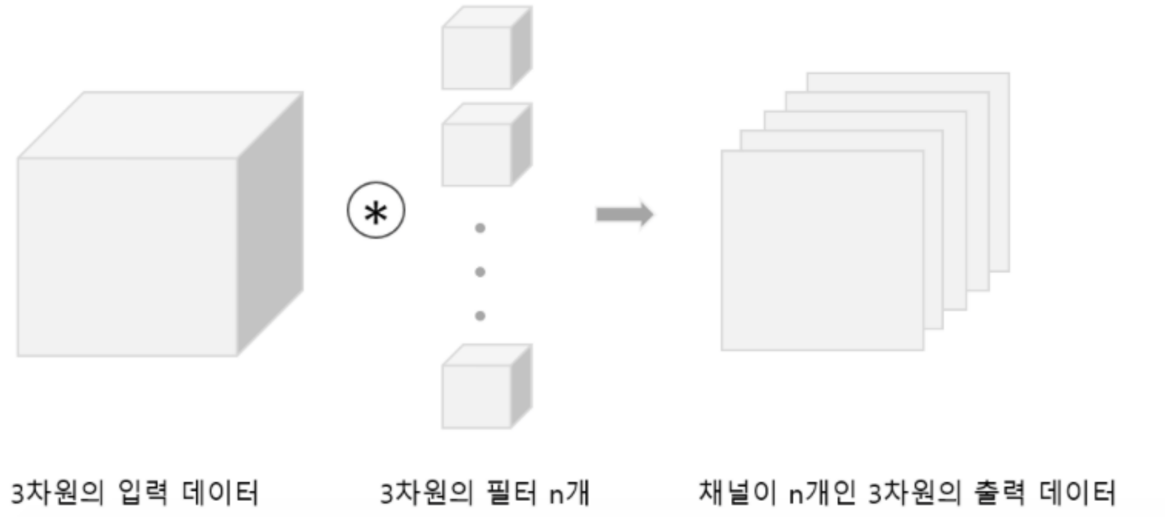
$n \times n$ image $f \times f$ filter

padding p stride s

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

3. CNN - Convolutions Over Volumes





$(n \times n \times n_c) * (f \times f \times n_c) = (n - f + 1) \times (n - f + 1) \times n_{c'}$ 형태가 됩니다.

- n : 이미지의 크기
- n_c : 채널의 개수
- f : 필터의 크기
- $n_{c'}$: 사용된 필터의 개수

4. 퀴즈 리뷰

1. 합성곱 연산에 대한 설명 중 틀린 것은 무엇인가요? *

- ○ 필터와 이미지의 각 픽셀에 대해 곱하고 모두 더해 결과를 얻습니다.
- ○ 합성곱 연산은 주로 이미지의 특징을 추출하는 데 사용됩니다.
- ◉ 작은 필터는 큰 특징을, 큰 필터는 작은 특징을 찾는 데 도움이 됩니다.

2. 패딩을 사용하는 이유는 무엇인가요? *

- ○ 이미지를 더 작게 만들기 위해서
- ◉ 이미지의 가장자리 정보를 유지하기 위해서
- ○ 연산 속도를 높이기 위해서

THANK YOU