

Deep Learning

CNN (Convolutional Neural Network)



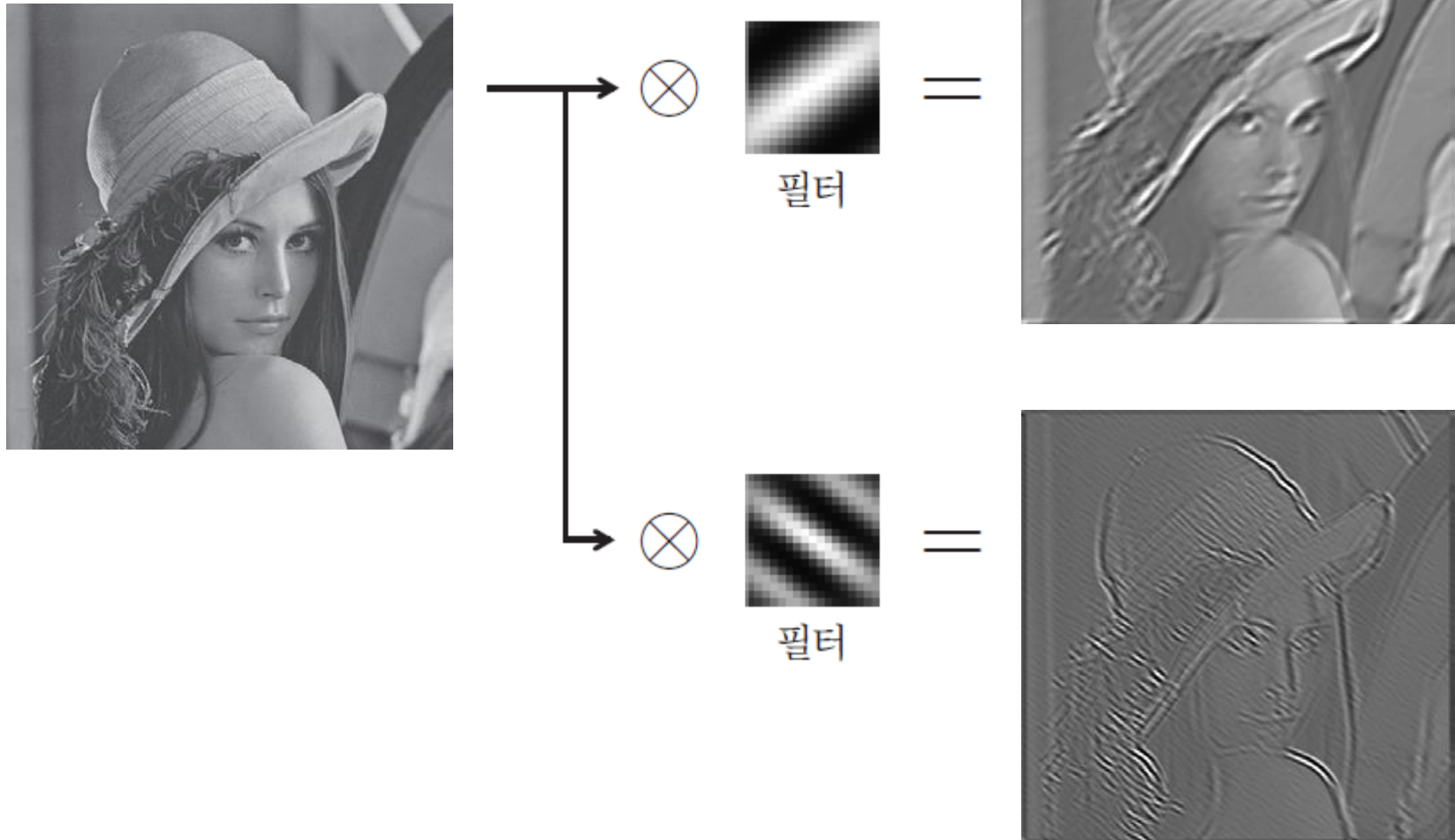
Ha-Jin Yu, Dept. of Computer Science, University of Seoul
서울시립대학교 컴퓨터과학부 유하진
2019

HJYU@UOS.AC.KR

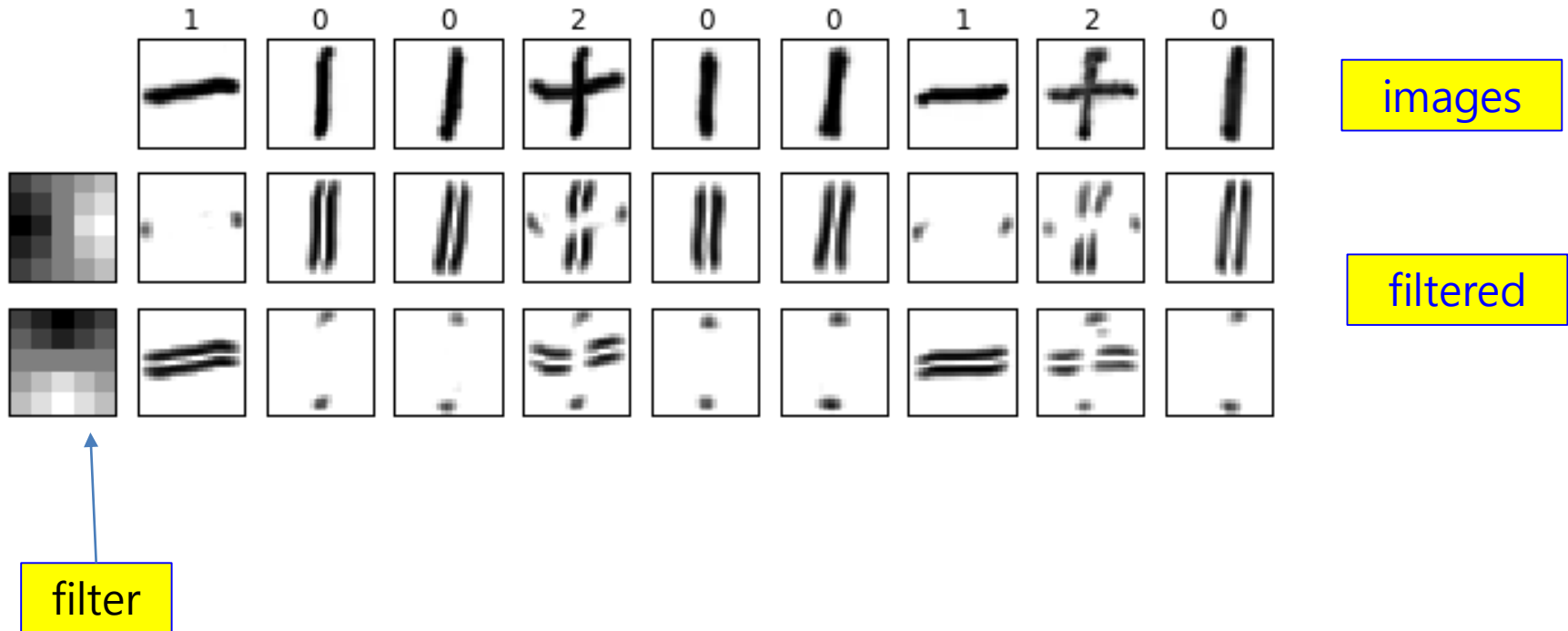


서울시립대학교
UNIVERSITY OF SEOUL

Example of filtering



Example of filtering 2



Convolutional Neural Networks

1	0	1
0	1	0
1	0	1

Kernel (3x3)

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

Convolution

Input

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34

Kernel or filter
(2x2)

w11	w12
w21	w22

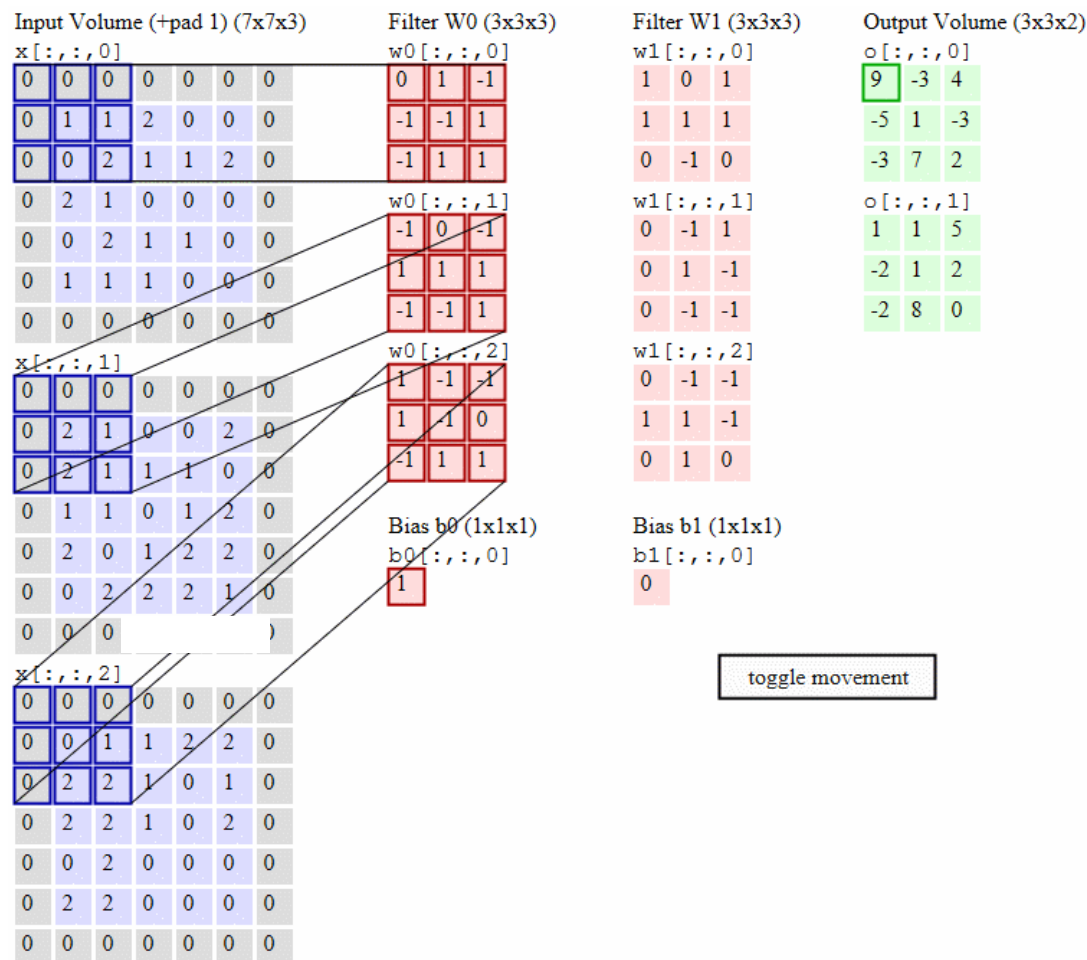
Output

$x_{11}w_{11} + x_{12}w_{12} + x_{21}w_{21} + x_{22}w_{22}$	$x_{12}w_{11} + x_{13}w_{12} + x_{22}w_{21} + x_{23}w_{22}$	$x_{13}w_{11} + x_{14}w_{12} + x_{23}w_{21} + x_{24}w_{22}$
$x_{21}w_{11} + x_{22}w_{12} + x_{31}w_{21} + x_{32}w_{22}$	$x_{22}w_{11} + x_{23}w_{12} + x_{32}w_{21} + x_{33}w_{22}$	$x_{23}w_{11} + x_{24}w_{12} + x_{33}w_{21} + x_{34}w_{22}$

Convolutional layer

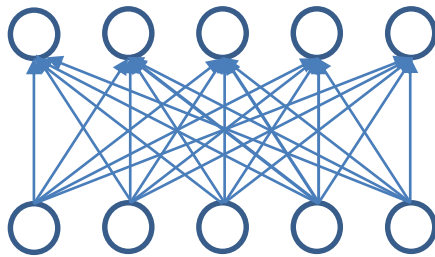
- Filter(또는 kernel) 단위로 연산

정해진 크기의 filter
(예: 3x3)가
일정 간격(예: 2x2)만큼
이동하면서
convolution 합 계산

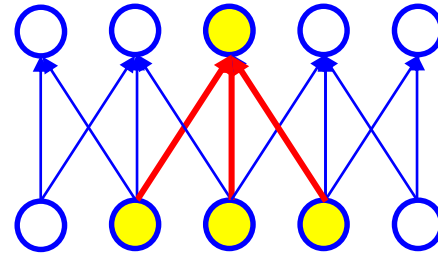


Sparse interactions

- 이미지에서 작고 의미 있는 특징 (예: 선분) 추출
→ Store fewer parameters
- *Sparse connectivity* or *sparse weights*
- Simple building blocks을 사용하여 복잡한 관계를 효과적으로 표현



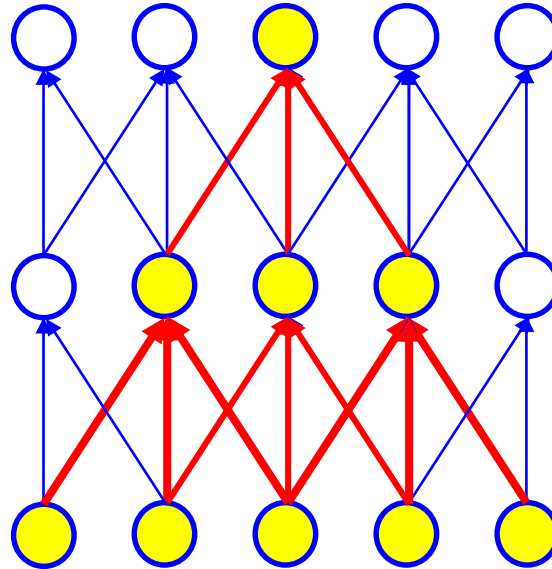
Fully Connected
Layer,
= Dense Layer
(전결합층)



Convolution
layer

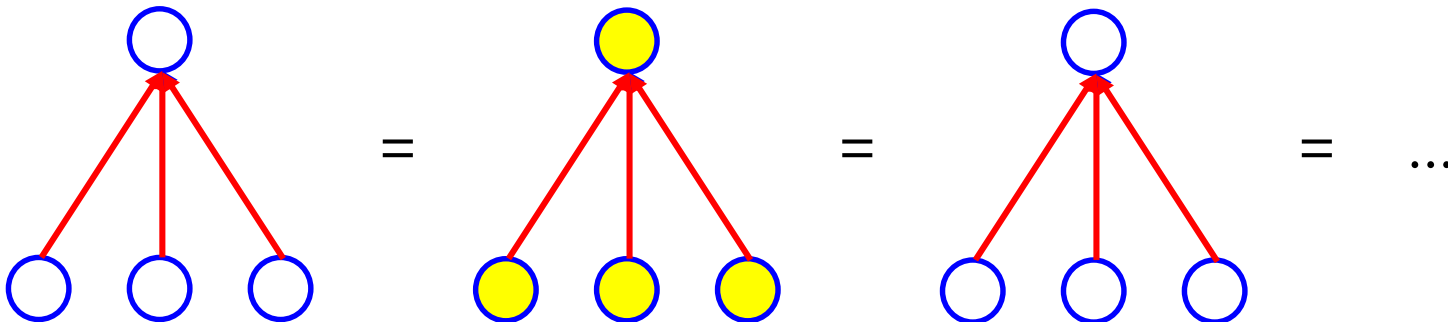
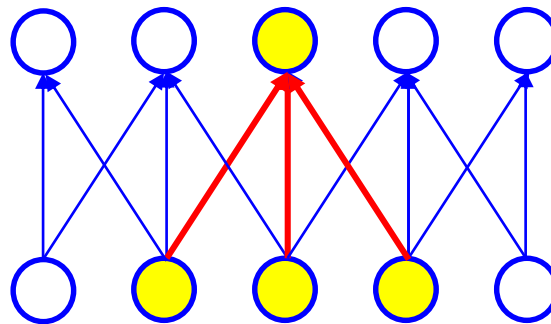
Deep convolutional network

- 더 넓은 입력 범위와 간접적으로 연결



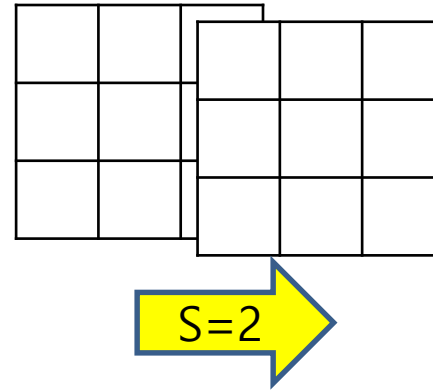
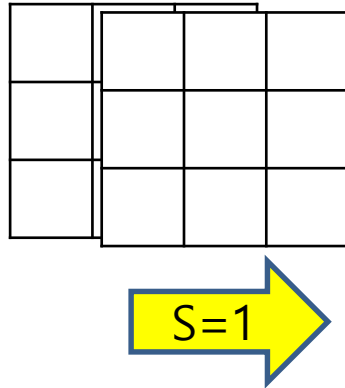
Parameter sharing

- Tied weights
- Memory requirements and statistical efficiency



Stride

- Stride : filter 의 적용 위치 간격 s



- 출력 이미지의 크기

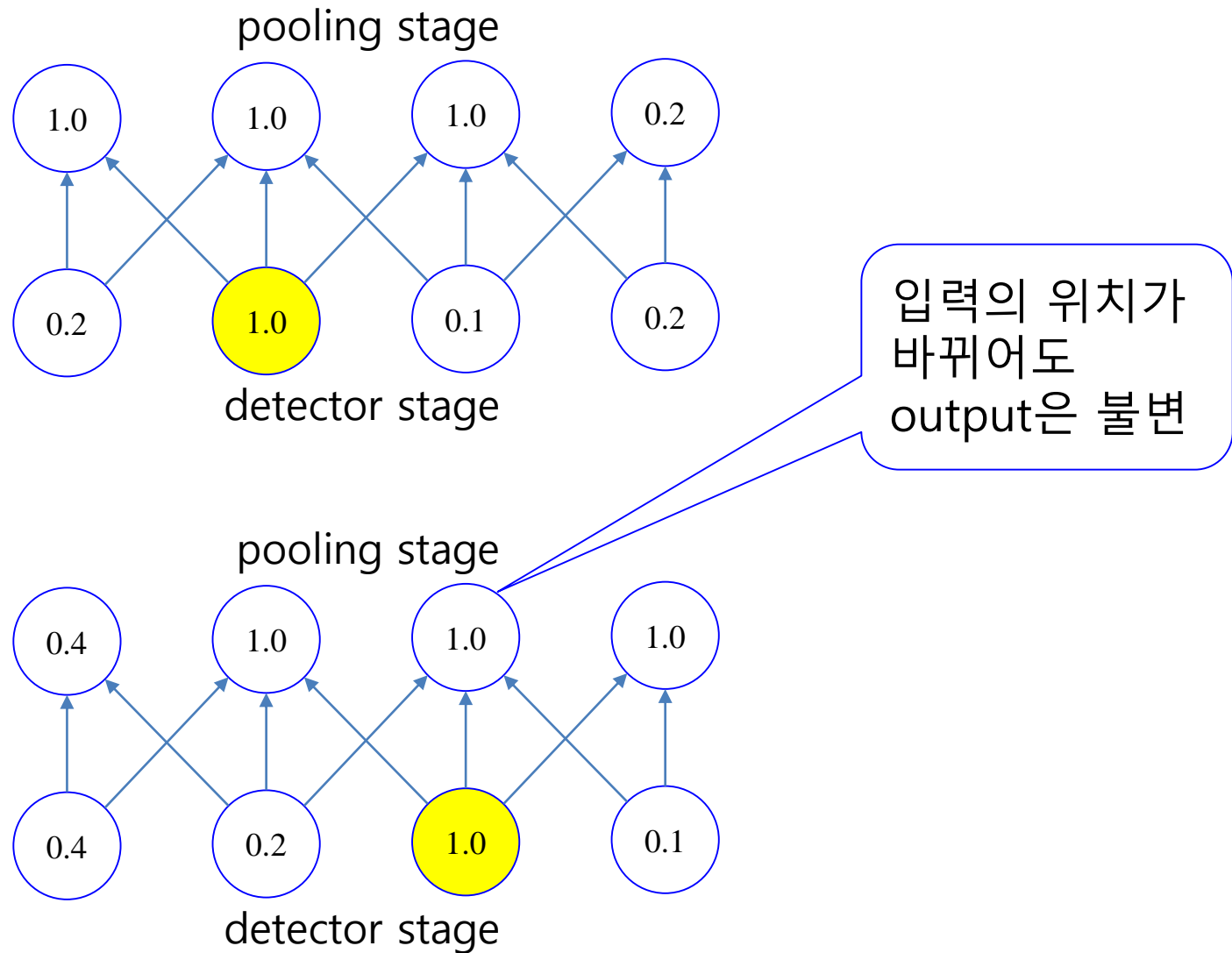
$$(\lfloor (W - 1)/s \rfloor + 1) \times (\lfloor (W - 1)/s \rfloor + 1)$$

W : image size

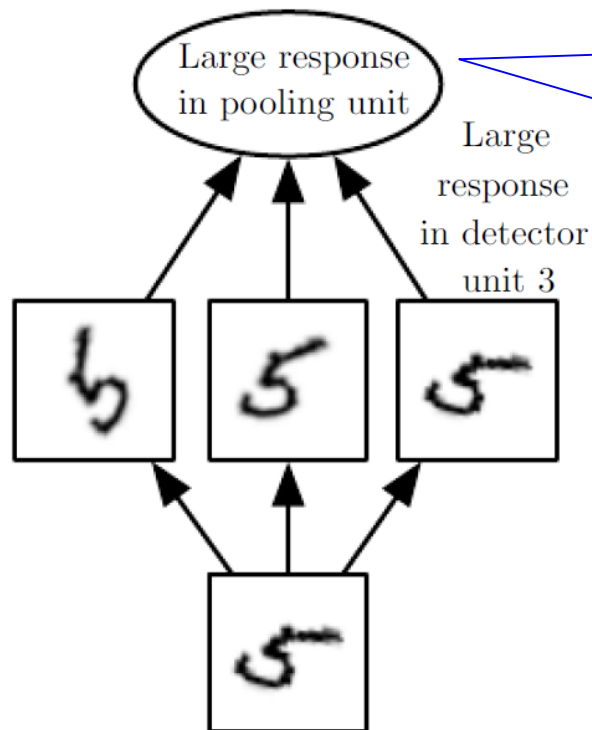
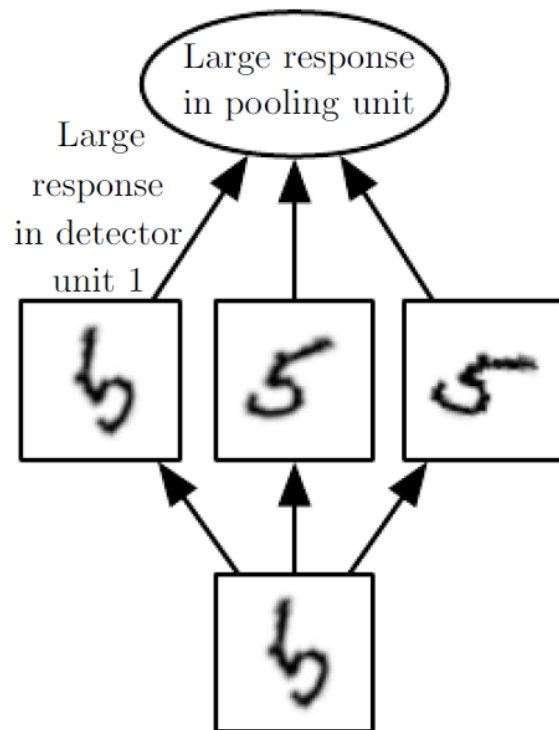
Pooling layer

- Filter를 이동시켜가며 filter가 가리키는 영역에서의 대표값을 취함
 - 대표값 선정에 따른 구분
 - Max pooling(최대값), average pooling(평균) 등
- Input의 위치 변화에 *invariant*
 - 어떤 특징이 “어디“ 있는가 보다 존재 여부가 중요
 - 예) 얼굴을 찾을 때 눈,코,입의 존재 여부

Max pooling 예



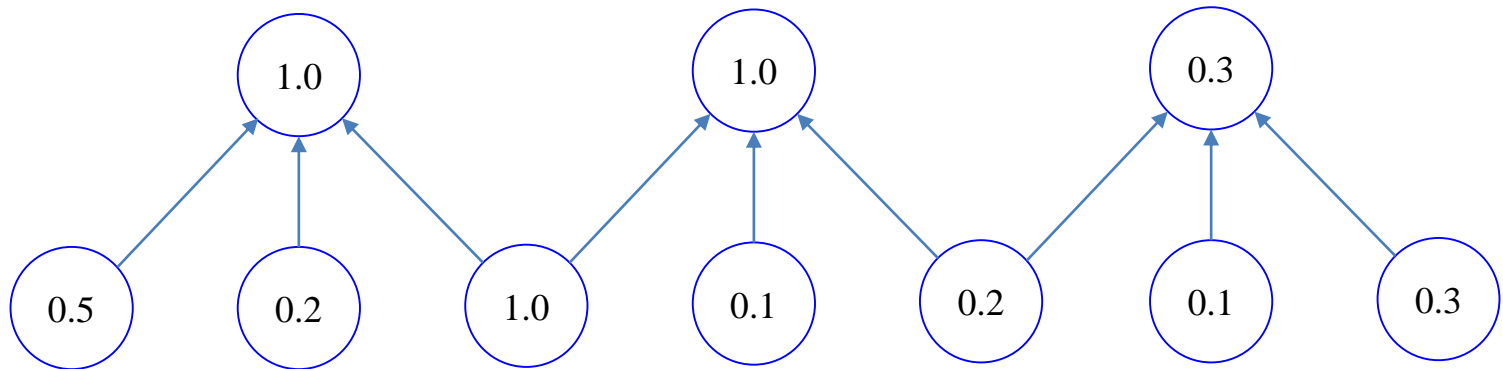
Max pooling 예



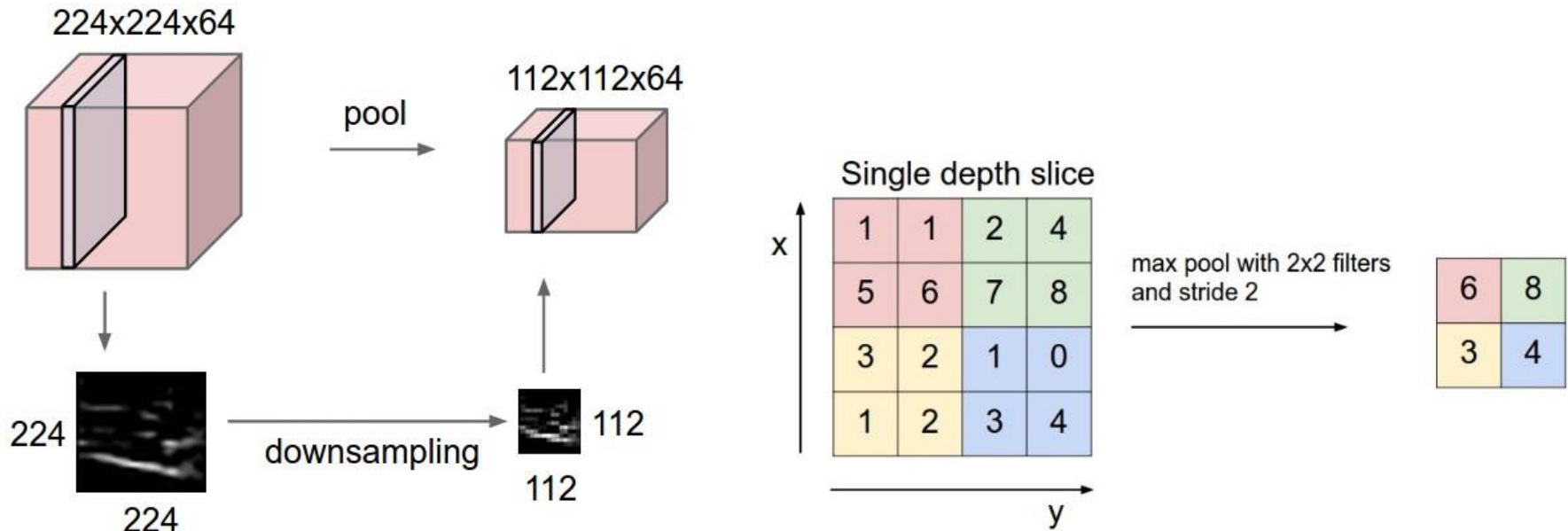
입력의 방향이 바뀌어도 output은 불변

Pooling with downsampling

- Improves the computational efficiency
- Essential for handling inputs of varying size → fixed output size.



Max pooling의 예



2*2 filter, 2*2 stride(subsampling)
max pooling의 예

Padding

- 합성곱 계산의 결과 이미지 크기 → 원래 크기보다 작아짐

$$(W - 2 \lfloor H/2 \rfloor) \times (W - 2 \lfloor H/2 \rfloor)$$

W : image size

H : filter size

- zero-padding : 바깥 부분을 0으로 추가하여 크기 유지

Zero-padding : “VALID”

- 크기 축소

77	80	82	78	70	82	82	140
83	78	80	83	82	77	94	151
87	82	81	80	74	75	112	152
87	87	85	77	66	99	151	167
84	79	77	78	76	107	162	160
86	72	70	72	81	151	166	151
78	72	73	73	107	166	170	148
76	76	77	84	147	180	168	142



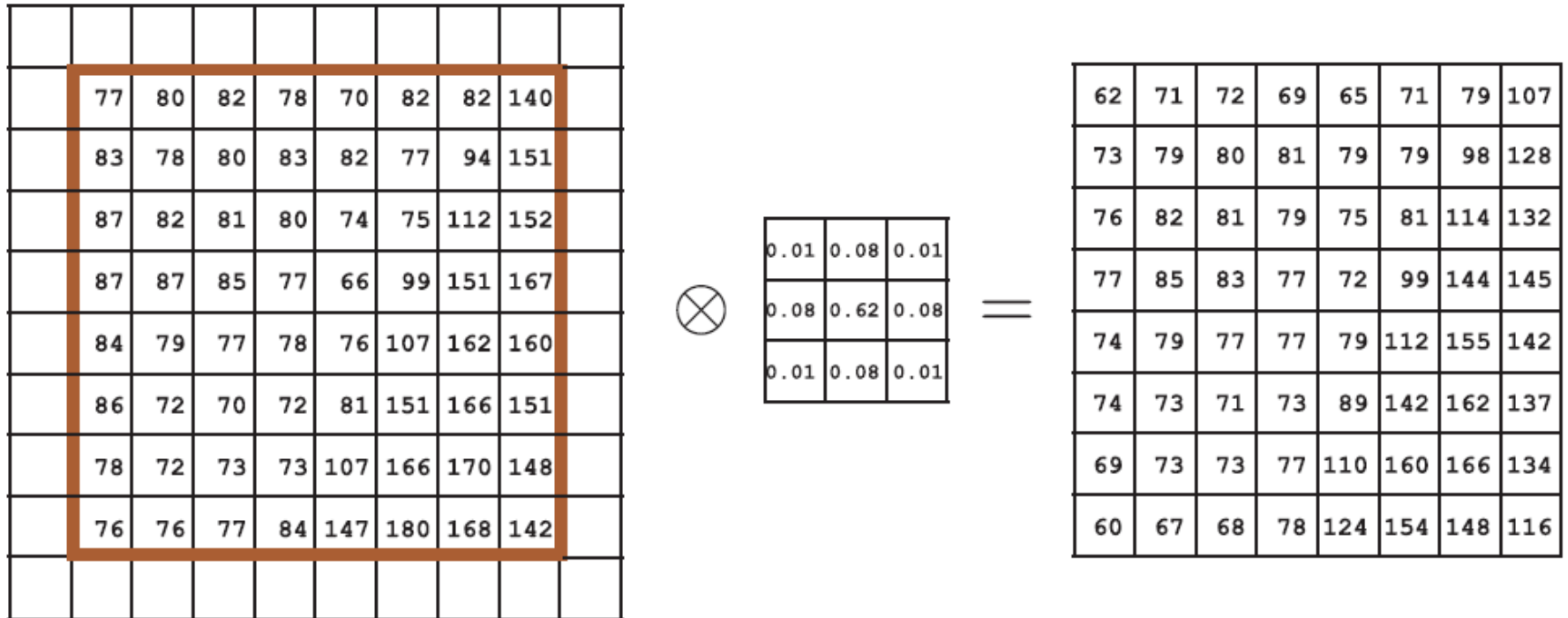
0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

=

79	80	81	79	79	98
82	81	79	75	81	114
85	83	77	72	99	144
79	77	77	79	112	155
73	71	73	89	142	162
73	73	77	110	160	166

Zero-padding : “SAME”

- 크기 8×8 인 이미지 주위로 zero padding을 적용하여 합성곱을 한 후에도 입력과 같이 8×8 이 되도록 한 경우.



CNN 구성의 예

