

Why CNN

Q. 컴퓨터 비전은 어디에 사용 되고 있는가?

Q. 컴퓨터 비전에서 인공지능에 관심이 있는가?

(1) 기술 발전으로 더 발전 된 APP 개발

(2) 컴퓨터 없이도 실생활에 많이 응용

Computer Vision Problems

Image Classification

Image Classification



64x64

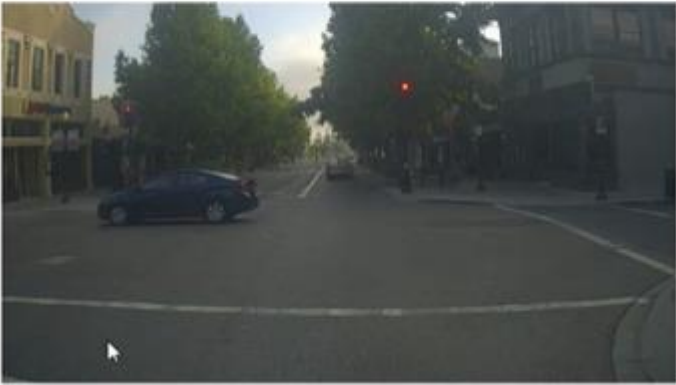
how to use?



Cat?
(1/0)

Object detection

Object detection

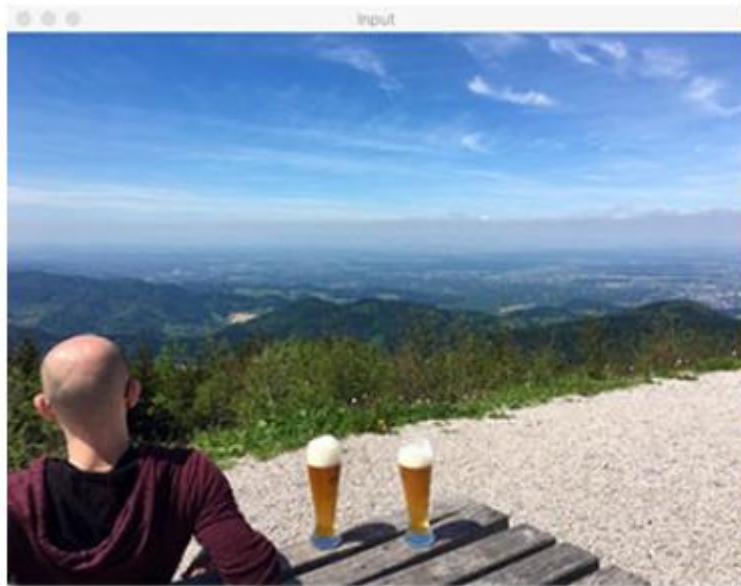


how to use?



Bouding Box

Neural style transfer



Input



Input



output

Large images problem



64x64

$64 \times 64 \times 3$

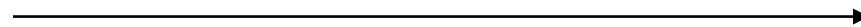


12288



1000 x 1000

$1000 \times 1000 \times 3$



3,000,000

Large images problem



1000 x 1000

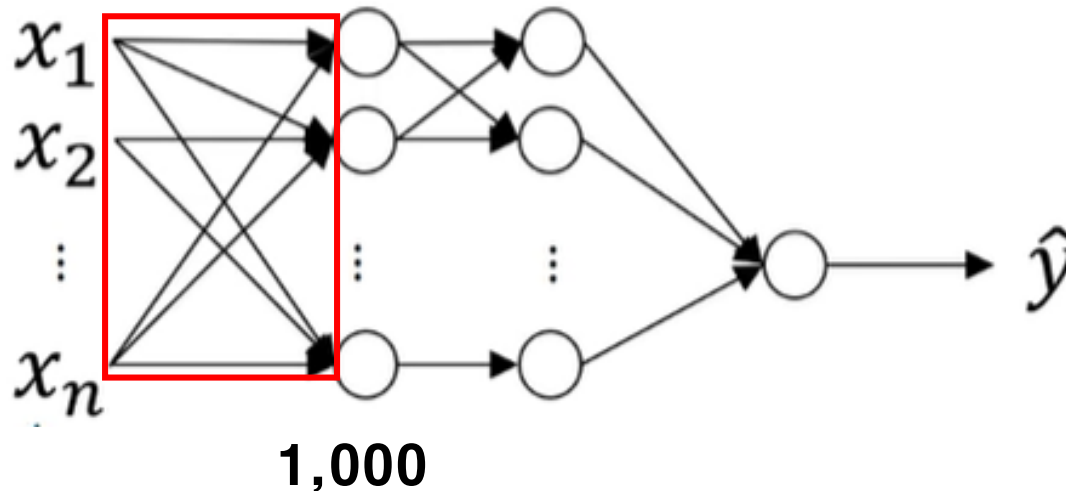
1000 x 1000 x 3



3,000,000

W : 1,000 x 3,000,000

if fully connected layer

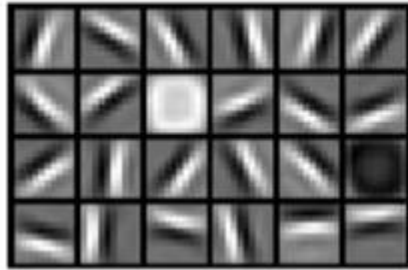


3,000,000

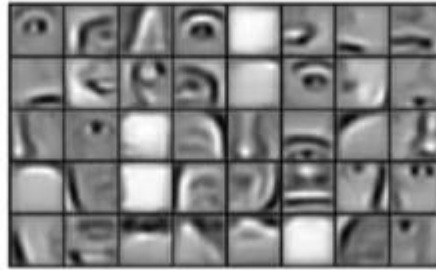
CNN

(Edge detection)

Computer Vision Problem



edge



possible object



complete object



vertical edges



horizontal edges

Vertical edge detection

6 x 6

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

convolution

*

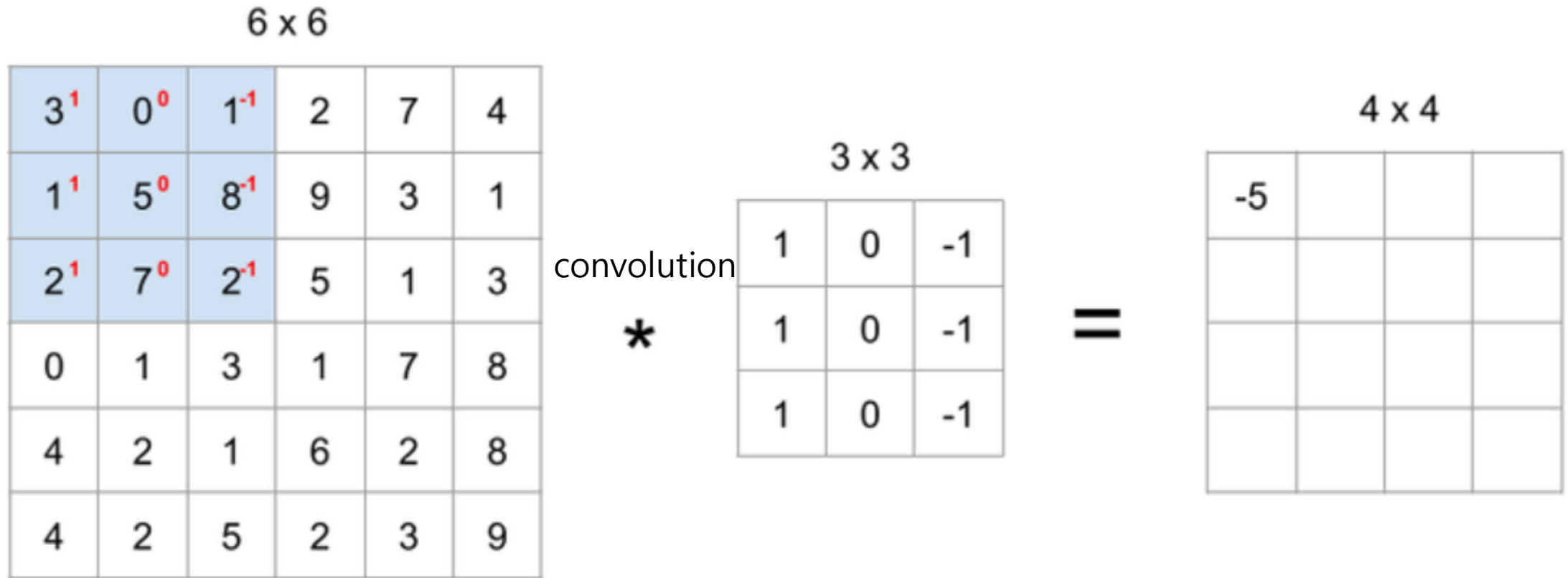
3 x 3

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

=

4 x 4

Vertical edge detection

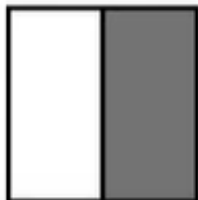


$$3 \times 1 + 0 \times 0 + 1 \times -1 + 1 \times 1 + 5 \times 0 + 8 \times -1 + 2 \times 1 + 7 \times 0 + 2 \times -1 = -5$$

Simple Vertical edge detection

6 x 6

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



*

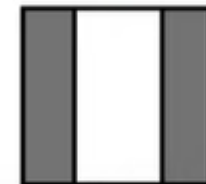
1	0	-1
1	0	-1
1	0	-1

=

4 x 4

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

*



Another Filters

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

Learning to detect edges

- ▶ CNN에서는 W의 값이 자동으로 정해진다.

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

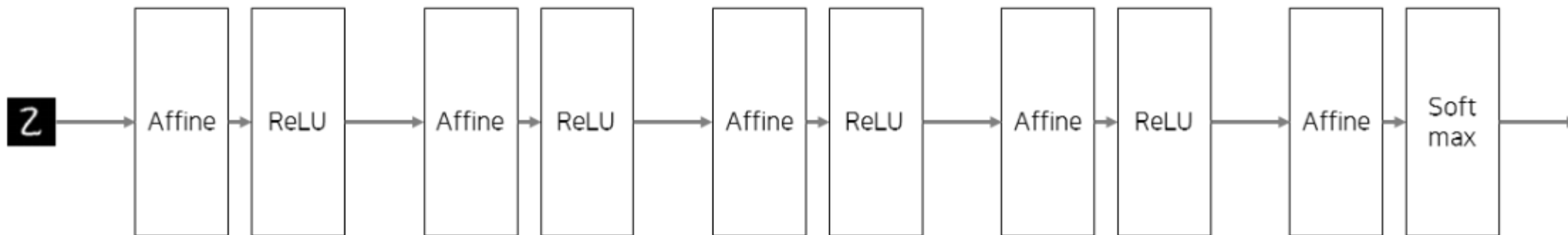
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Affine

1. 지금까지의 신경망 구조

완전연결(fully-connected) : 인접하는 계층의 모든 뉴런과 결합

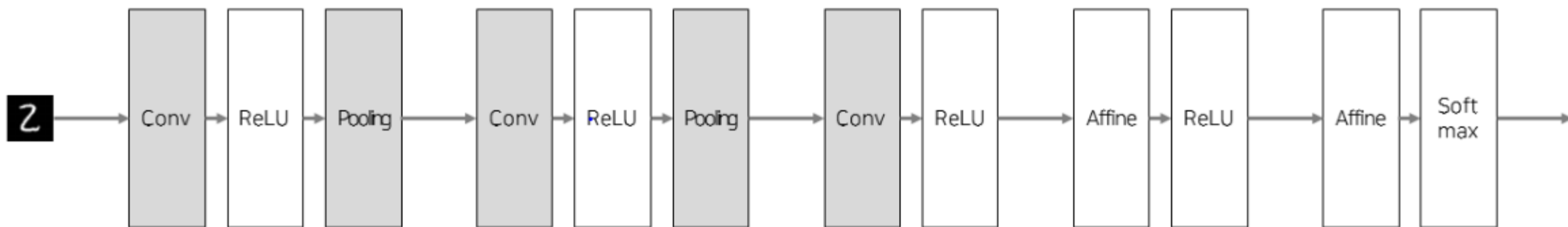
완전연결된 계층 : Affine 계층



Affine – ReLU 연결

CNN

2. CNN : 합성곱 계층(Conv)와 풀링 계층(Pooling)이 추가



- Affine – ReLU 연결 -> Conv – ReLU – (Pooling)
- 출력에 가까운 계층에서는 Affine – Softmax 조합을 그대로 사용

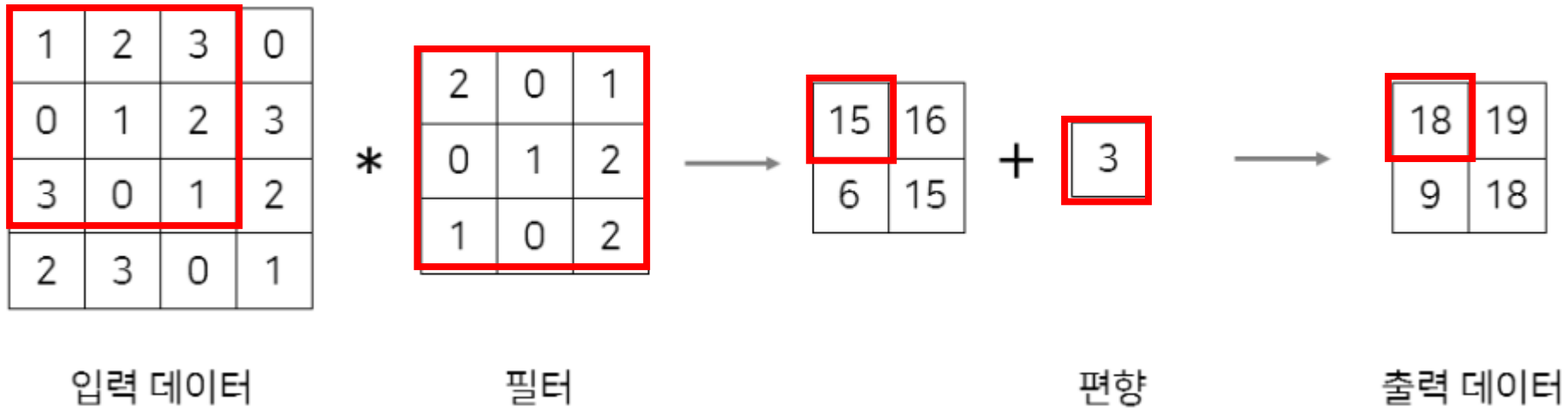
CNN

3. 완전연결계층의 문제점

- 완전연결 계층에 입력하기 위해서 3차원이 이미지 데이터를 1차원 데이터로 평탄화
=> 데이터의 형상이 무시되어 담긴 정보가 사라짐
- 합성곱 계층은 형상을 유지
- CNN에서는 합성곱 계층의 입출력 데이터를 특징 맵(feature map)이라고도 합니다.

합성곱연산

4. 합성곱 연산

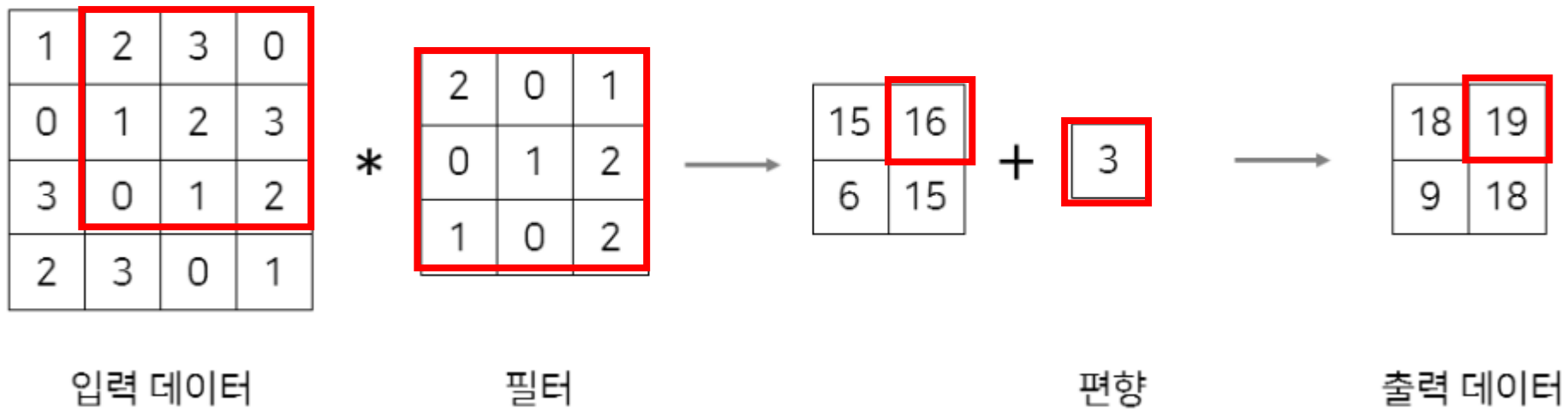


$$(1 \times 2) + (2 \times 0) + (3 \times 1) + (0 \times 0) + (1 \times 1) + (2 \times 2) + (3 \times 1) + (0 \times 0) + (1 \times 2) = 15$$

$$15 + 3 = 18$$

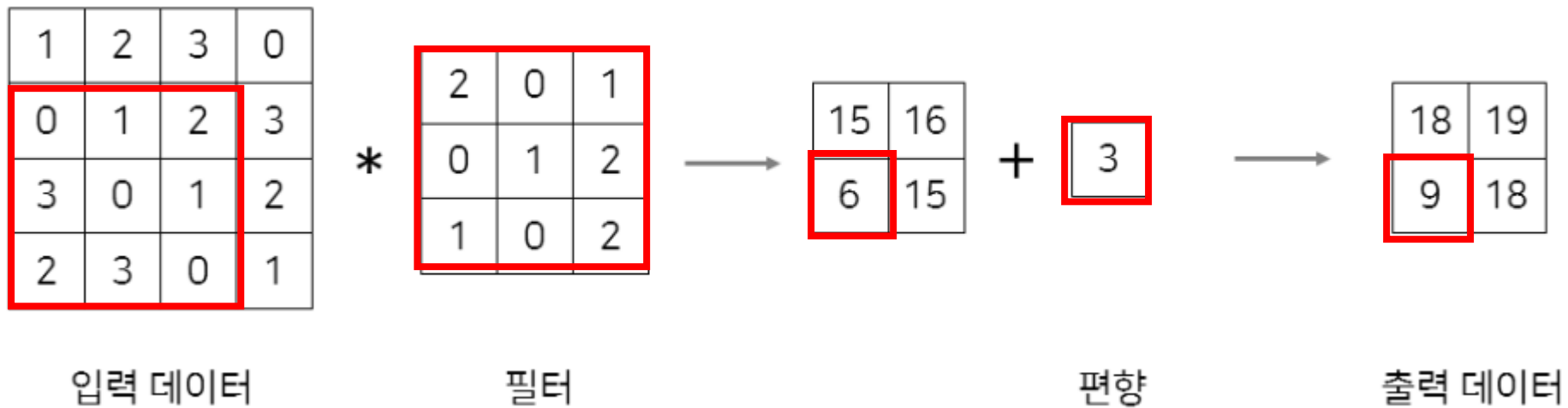
합성곱연산

4. 합성곱 연산



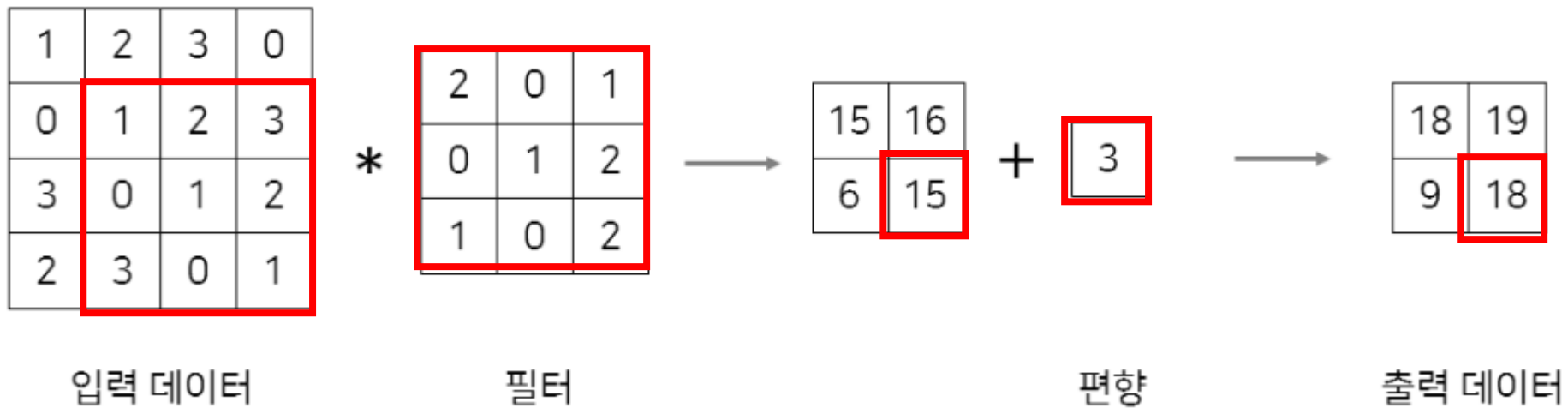
합성곱연산

4. 합성곱 연산



합성곱연산

4. 합성곱 연산



합성곱연산

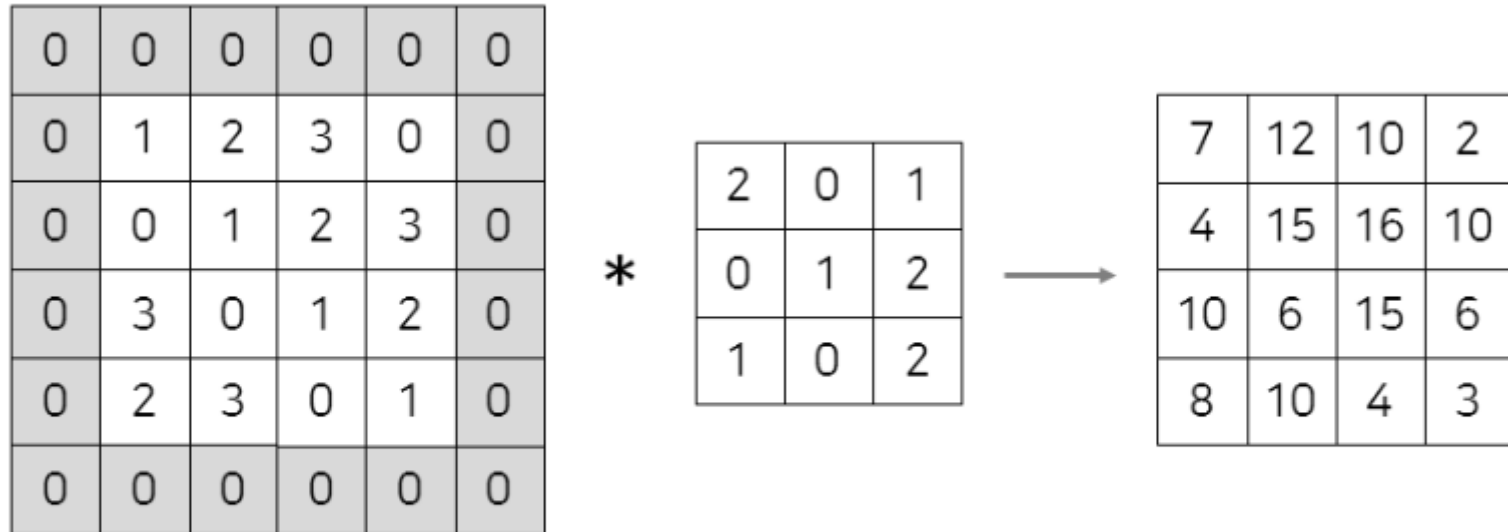
**Q1. 10×10 의 입력 데이터가 3×3 의 필터와
합성곱 연산은 했을 때 출력데이터의 크기는?**

Q. Why Padding?

패딩(Padding)

- ▶ 합성곱 연산 시 한번만 사용되는 가장 가장자리의 연산의 정보를 덜 버리게 한다.

5. 패딩(Padding)



입력 데이터

필터

출력 데이터

패딩 : 1 적용

패딩 적용으로 2x2 -> 4x4

스트라이드(Stride)

5. 패딩(Padding) : Stride = 2

1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1
2	3	0	1	2	3	0
1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

입력 데이터

*

2	0	1
0	1	2
1	0	2

필터



15	17	

출력 데이터

스트라이드(Stride)

5. 패딩(Padding) : Stride = 2

1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1
2	3	0	1	2	3	0
1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

입력 데이터

*

2	0	1
0	1	2
1	0	2

필터



15	17	

출력 데이터

스트라이드(Stride)

5. 패딩(Padding) : Stride = 2

1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1
2	3	0	1	2	3	0
1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

입력 데이터

*

2	0	1
0	1	2
1	0	2

필터



15	17	

출력 데이터

출력층 계산

6. 출력층 계산

입력 크기: (H, W)

필터 크기: (FH, FW)

출력 크기: (OH, OW)

패딩: P

스트라이드: S

$$OH = \frac{H + 2P - FH}{S} + 1$$

$$OW = \frac{W + 2P - FW}{S} + 1$$

Q. 입력 (6, 8) , 패딩 2 , 스트라이드 : 2 , 필터 : (2, 2) 일 때 출력 사이즈는?
(가급적 그려 볼 것!)

출력층 계산

6. 출력층 계산

입력 크기: (H, W)

필터 크기: (FH, FW)

출력 크기: (OH, OW)

패딩: P

스트라이드: S

$$OH = \frac{H + 2P - FH}{S} + 1$$

$$OW = \frac{W + 2P - FW}{S} + 1$$

Ex) 입력: (6,8), 패딩: 2, 스트라이드: 2, 필터: (2,2)

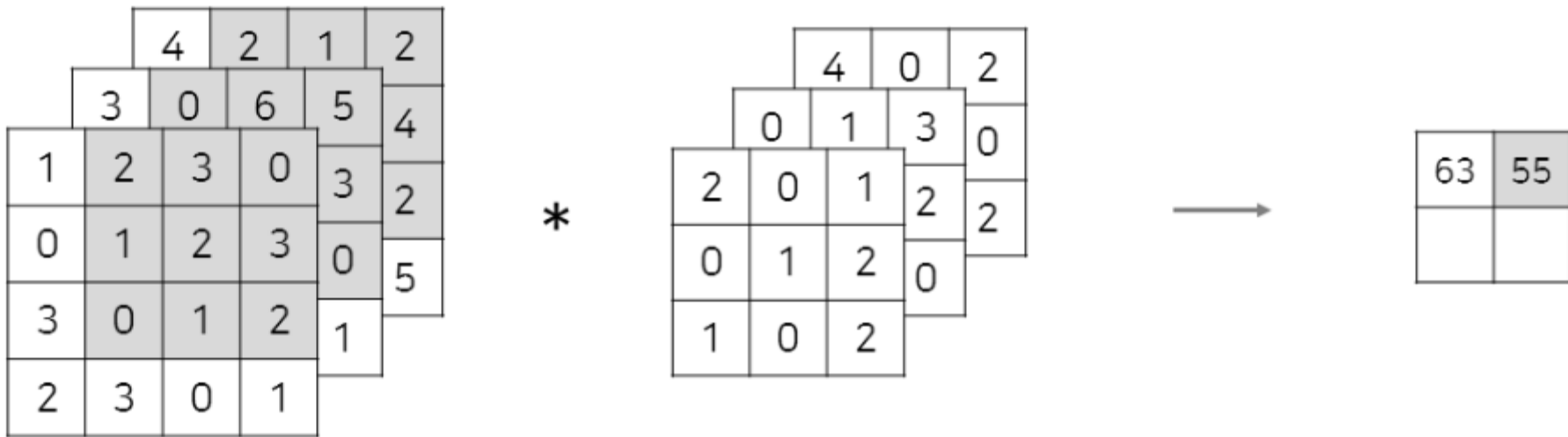
$$OH = \frac{6 + 2 * 2 - 2}{2} + 1 = 5$$

$$OW = \frac{8 + 2 * 2 - 2}{2} + 1 = 6$$

출력 크기: (5, 6)

3차원 데이터의 합성곱 연산

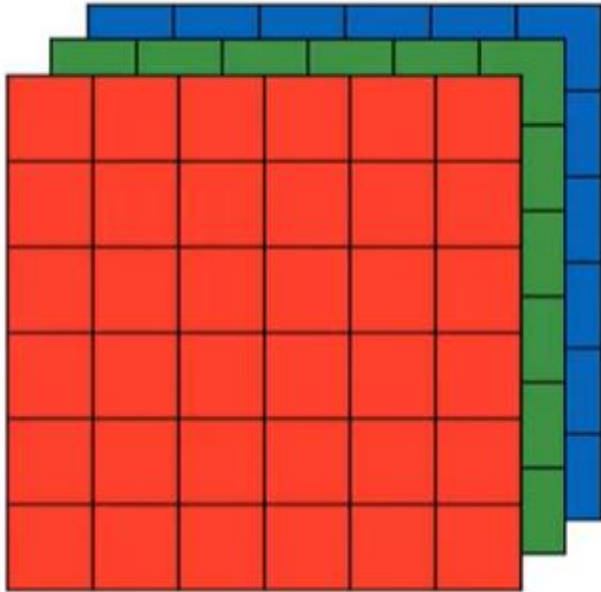
▶ 3차원 데이터의 합성곱 연산 : 입력데이터의 채널 수와 필터의 채널 수가 같아야 한다.



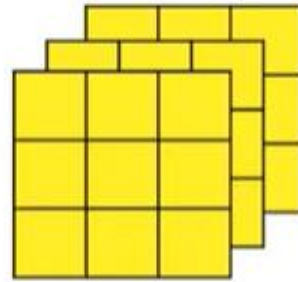
3차원의 데이터로 하나의 블록으로 생각하면 된다.

3차원 데이터의 합성곱 연산

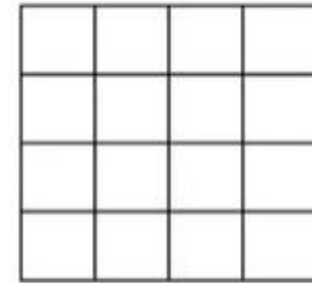
▶ Q. RGB 이미지에서 빨간색 (수직)엣지를 검출 하려고 할 때 필터는 어떻게 구성되는가?



*



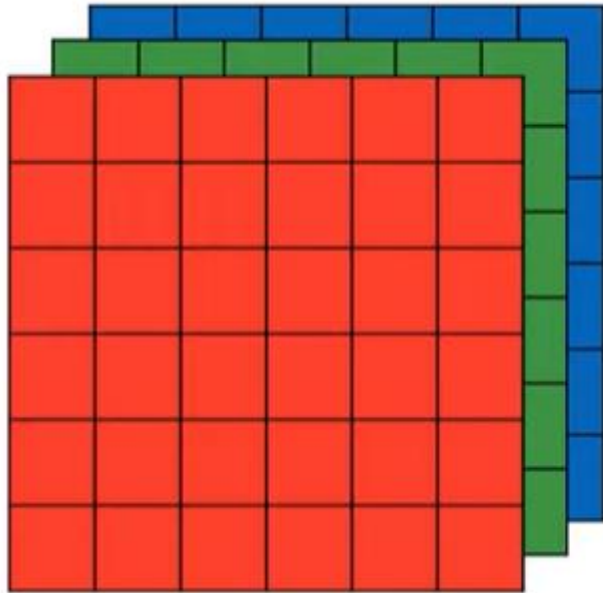
=



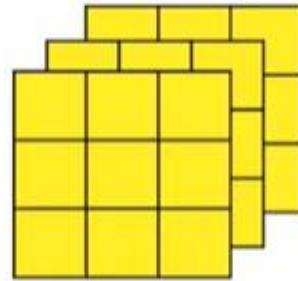
4 x 4

3차원 데이터의 합성곱 연산

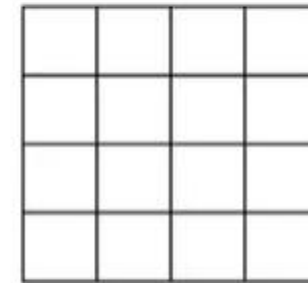
▶ Q. RGB 이미지에서 빨간색 엣지를 검출 하려고 할 때 필터는 어떻게 구성되는가?



*



=



4 x 4

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

R

0	0	0
0	0	0
0	0	0

G

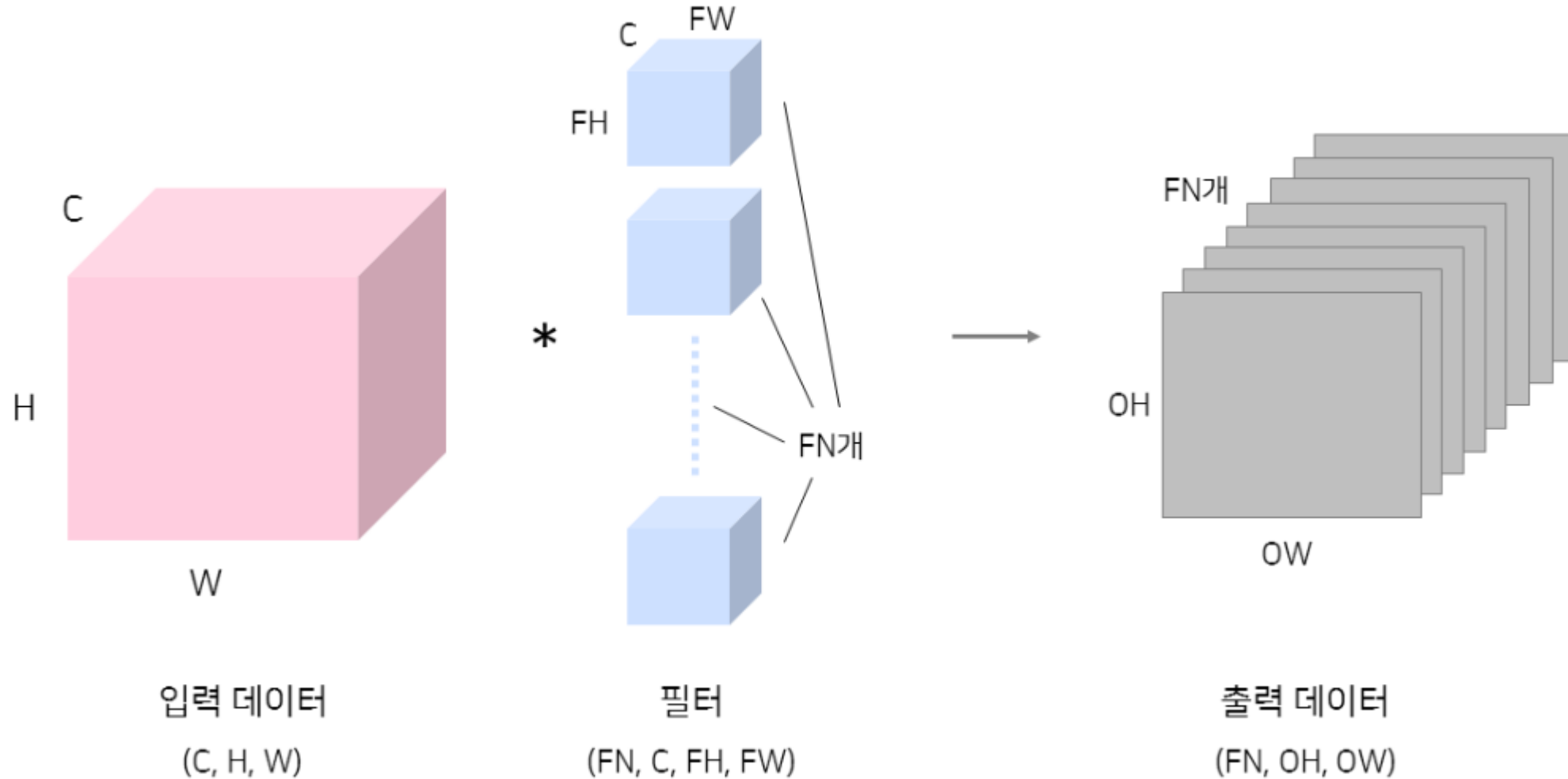
0	0	0
0	0	0
0	0	0

B

**Q. 다양한 특징(feature mpa)을
검출 하고 자 할 때
많아야 것은?**

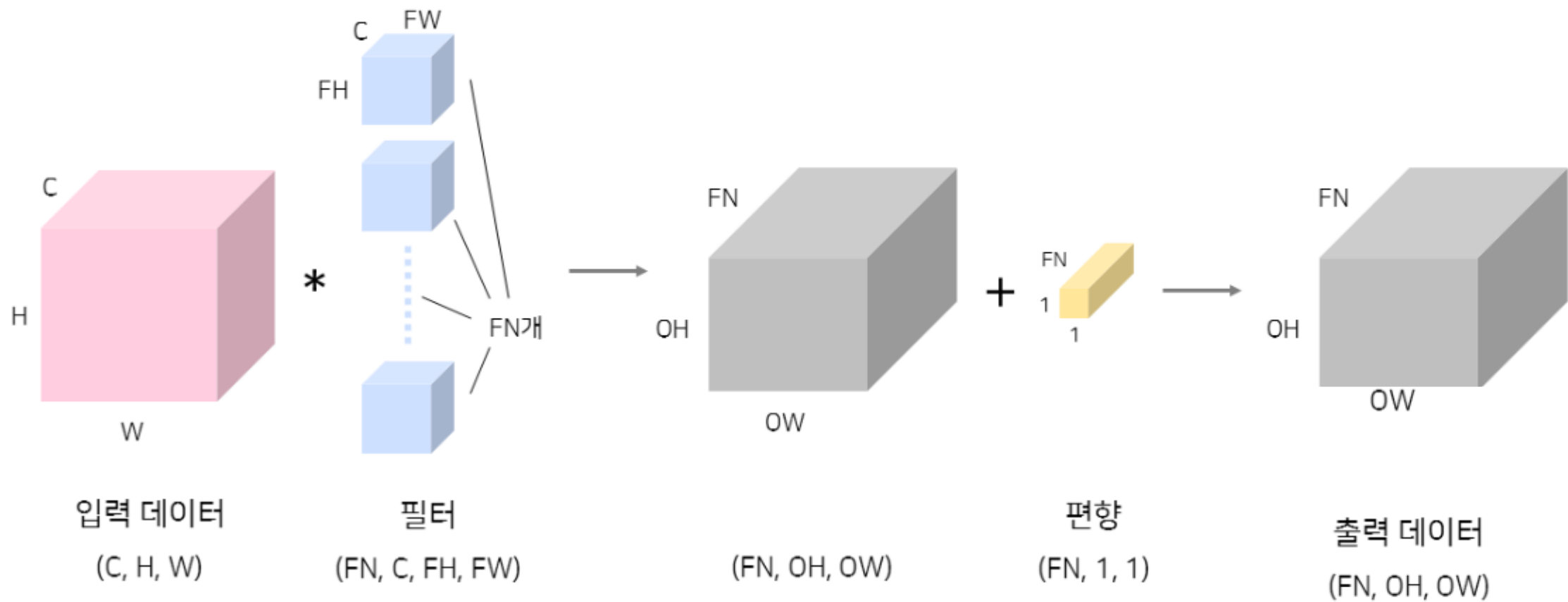
3차원 데이터의 합성곱 연산

- ▶ FN개의 필터를 적용하여 FN개의 출력데이터(Feature map) 생성



3차원 데이터의 합성곱 연산

▶ 동일하게 편향 적용 가능

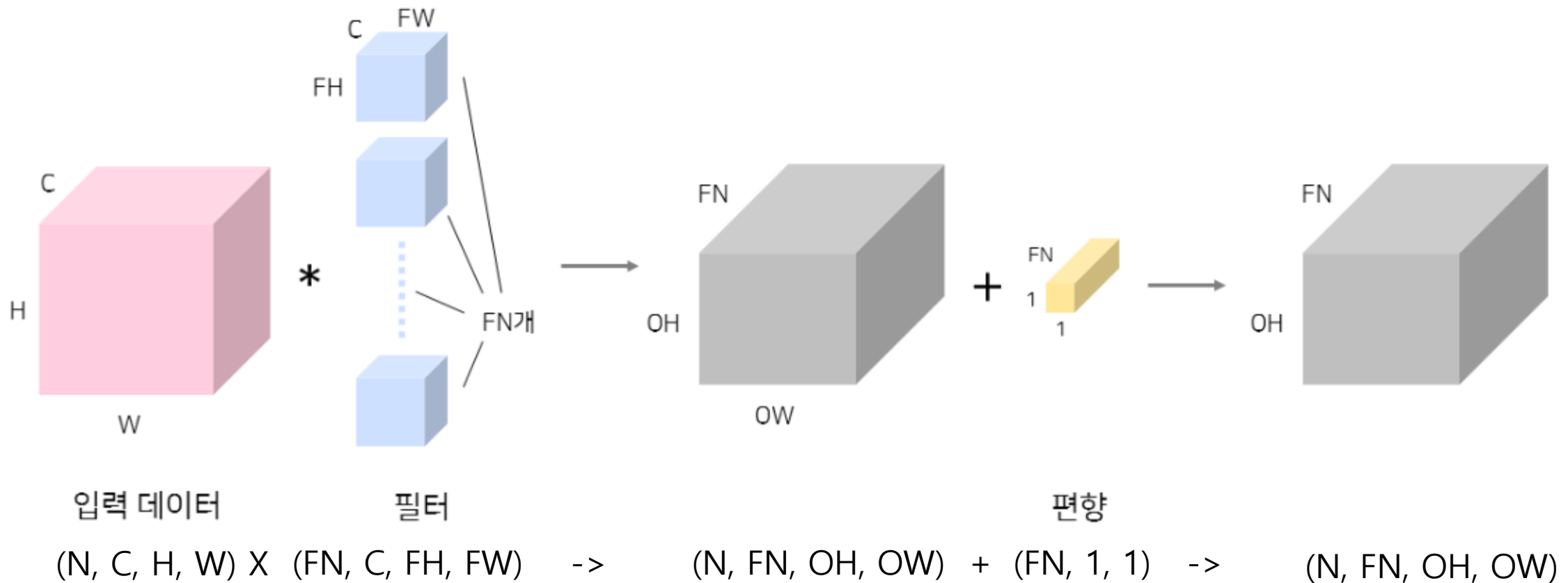


3차원 데이터의 합성곱 연산

▶ 합성곱 연산에서는 데이터의 수도 고려해야 한다.

-> 3차원 데이터를 데이터 수를 추가한 4차원 데이터로 저장합니다.

(데이터 수, 채널 수, 높이, 너비) 순으로 저장



**Q1. $28 \times 28 \times 3$ 이미지가 5×5 의 필터 20
개를 사용하여 계산 한다면 결과 값은?**

Q3. “?” 채우기.

입력 크기: (H, W)

필터 크기: (FH, FW)

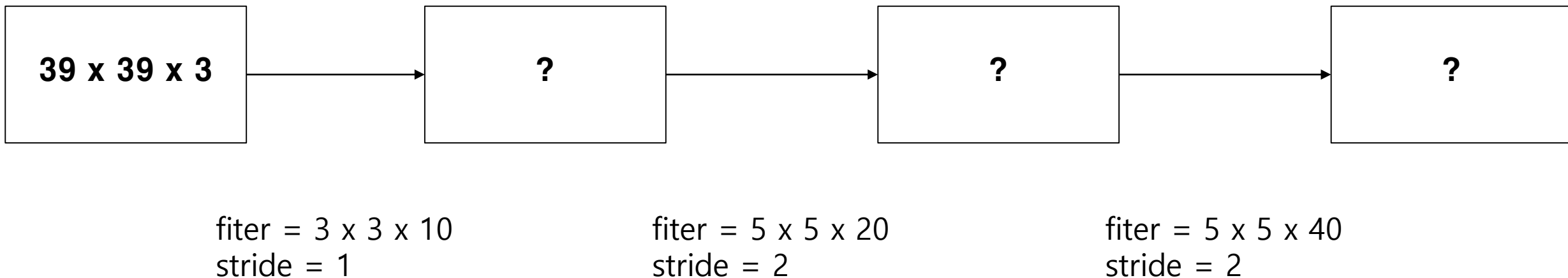
출력 크기: (OH, OW)

패딩: P

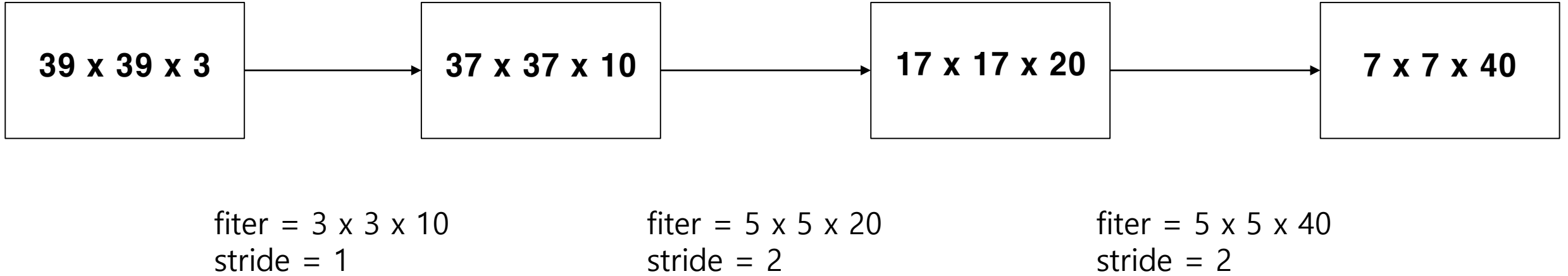
스트라이드: S

$$OH = \frac{H + 2P - FH}{S} + 1$$

$$OW = \frac{W + 2P - FW}{S} + 1$$



Q3.



풀링(pooling)

- ▶ 풀링은 세로, 가로 방향의 공간을 줄이는 연산입니다. (overfitting) 방지가 목적)

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4



10	

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4



10	21

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4



10	21
6	

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4



10	21
6	8

풀링(pooling)

▶ Q. 최대 풀링으로 빈칸을 채워보세요.

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

filter : 3 x 3
stride : 1

평균풀링(Average Pooling)

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2



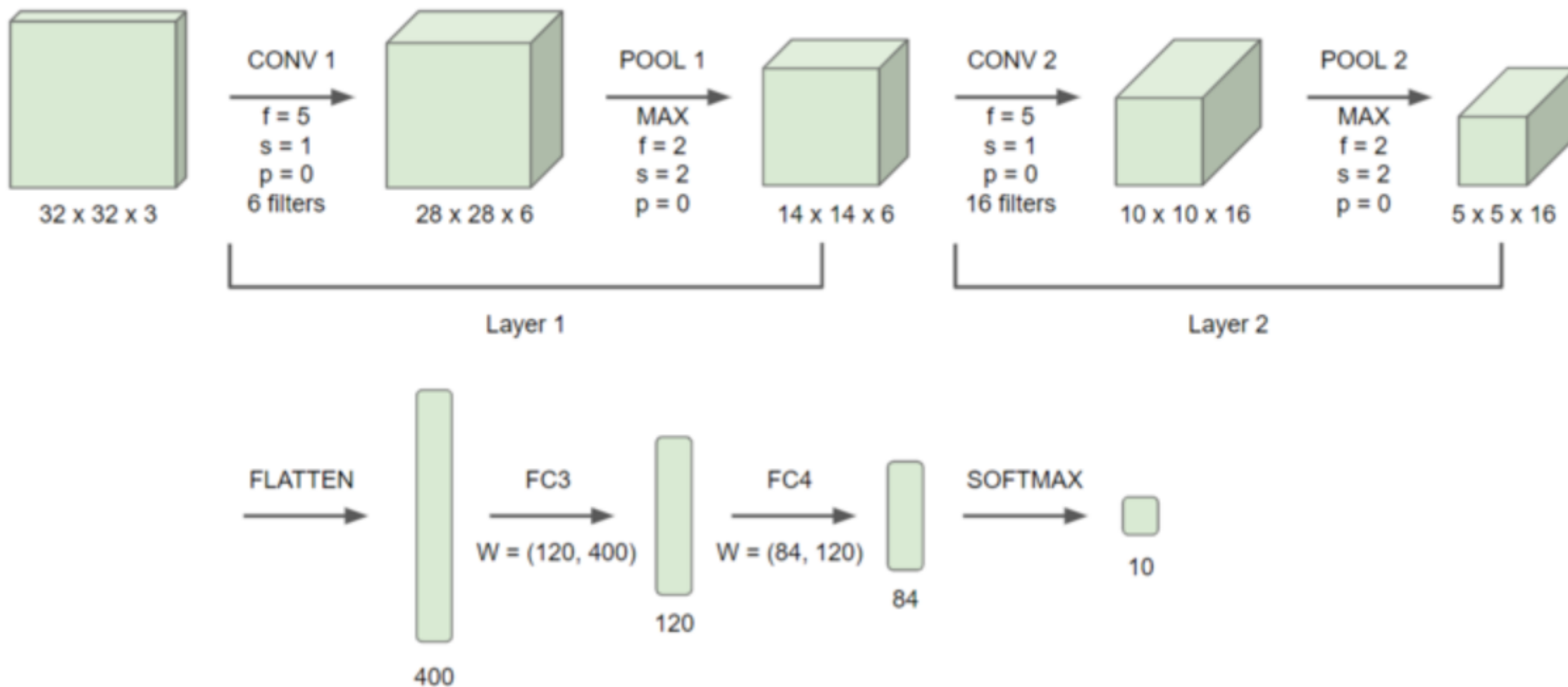
Average Pooling

3.75	1.25
3.75	2.0

- 풀링의 윈도우 크기와 스트라이드는 보통 같은 값으로 설정
- 최대풀링외에 평균풀링이 있으나, 이미지분석에서는 **주로 최대 풀링을 사용**

Simple CNN

LeNet - 5



Simple CNN

	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072	0
CONV1 (f=5, s=1)	(28,28,8)	6,272	208
POOL1	(14,14,8)	1,568	0
CONV2 (f=5, s=1)	(10,10,16)	1,600	416
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48,001
FC4	(84,1)	84	10,081
Softmax	(10,1)	10	841