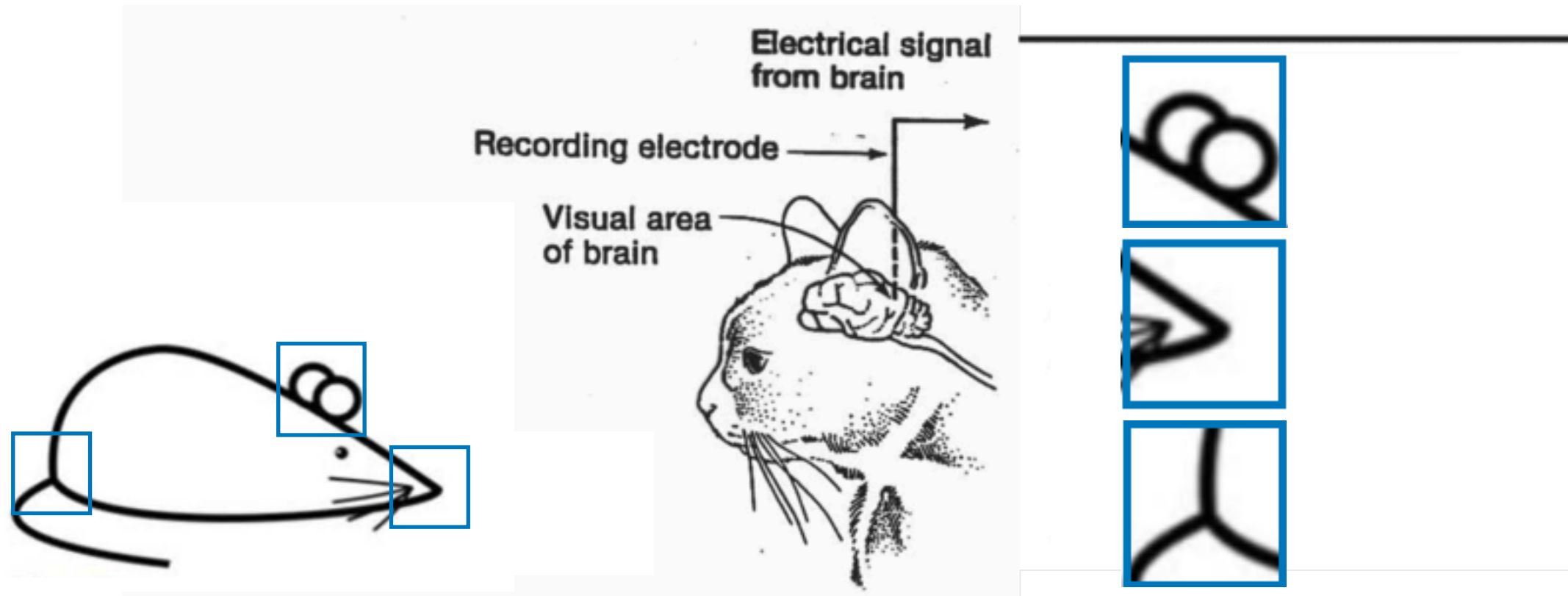


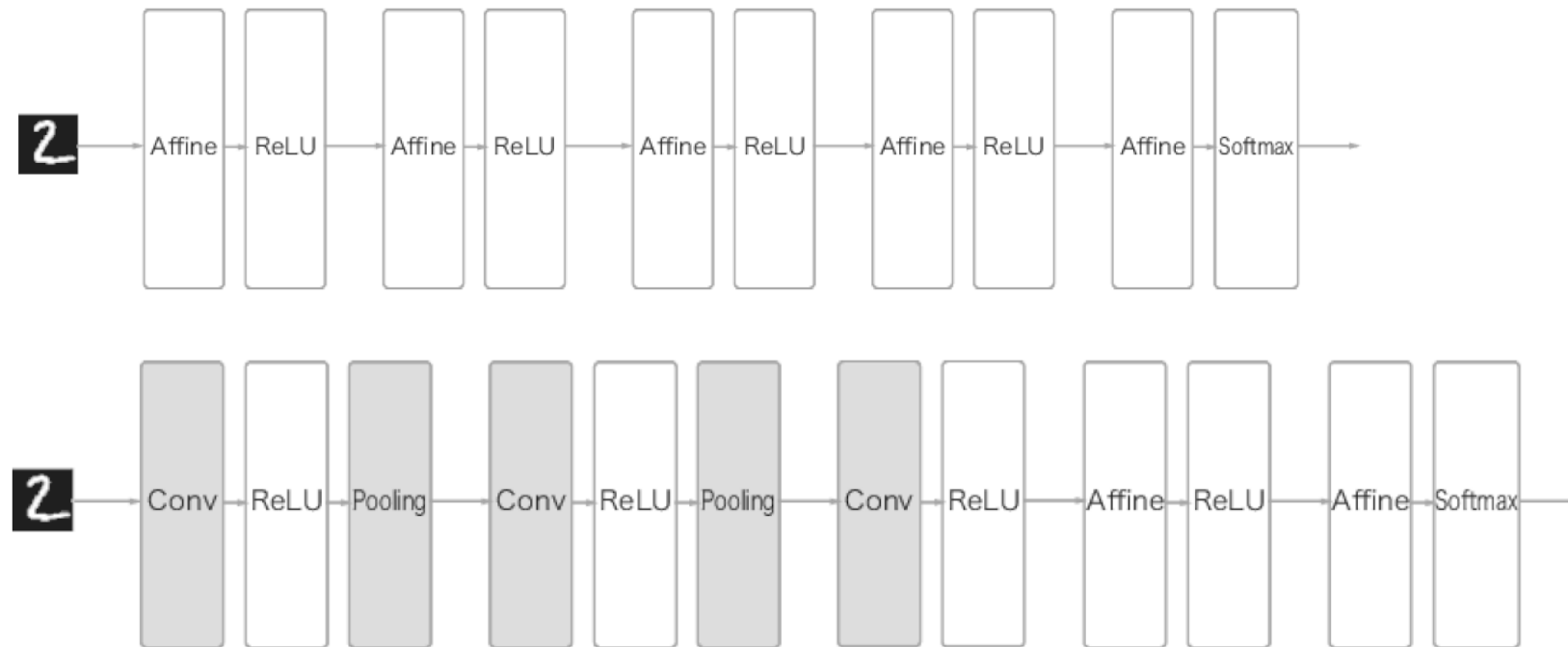
# Convolutional Neural Network

# Convolutional NN



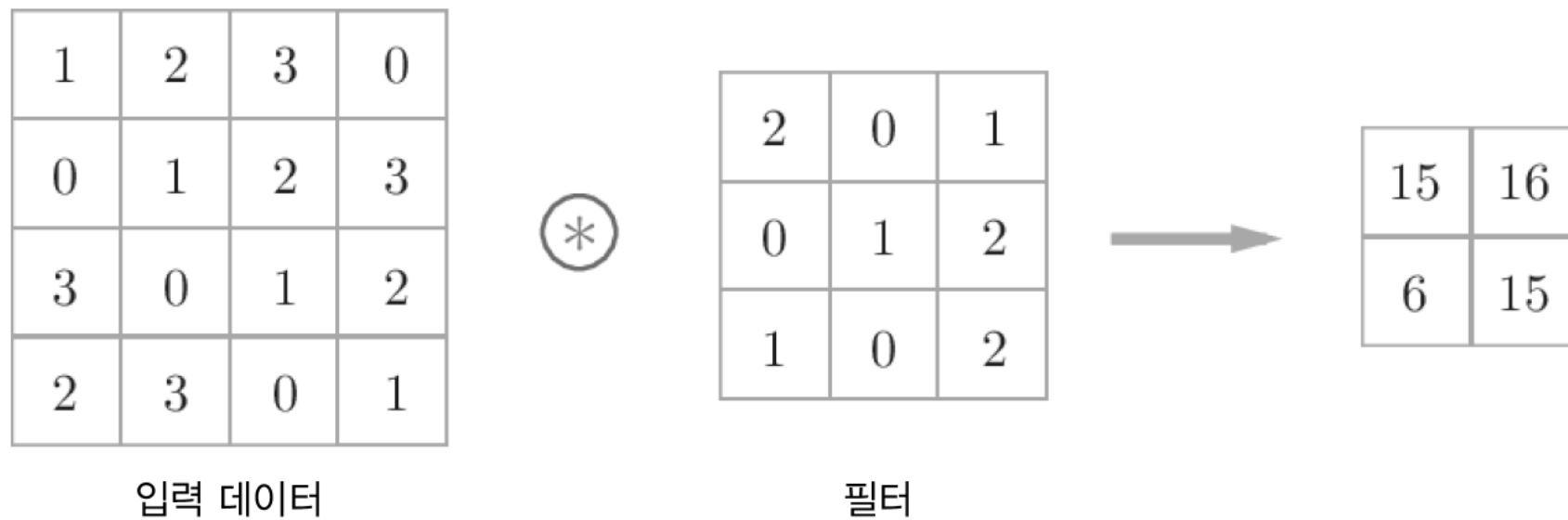
- 고양이에게 그림을 보여주니 각각의 뉴런들이 각기 다른 부분에서 활성화 되는것을 관찰
- 뉴런들이 입력을 조금씩 나누어서 처리
- 이미지 정보를 조금씩 나누어서 각기 학습하는 방법이 제시

# Convolution NN



- 일반적으로 신경망은 Affine 층과 ReLU층이 반복
- 신경망 앞단에 Convolution-ReLU-(Pooling)층을 반복

# Convolution(합성곱)



- 이미지의 크기가 4 x 4이고 컬러가 1인 이미지
- 3 x 3 크기의 필터를 이용한 Convolution
- 4 x 4 이미지를 3 x 3 필터로 합성곱을 하면 2 x 2

# Convolution(합성곱)

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

 $\otimes$ 

2	0	1
0	1	2
1	0	2

 $\rightarrow$ 

15	

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

 $\otimes$ 

2	0	1
0	1	2
1	0	2

 $\rightarrow$ 

15	16

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

 $\otimes$ 

2	0	1
0	1	2
1	0	2

 $\rightarrow$ 

15	16
6	

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

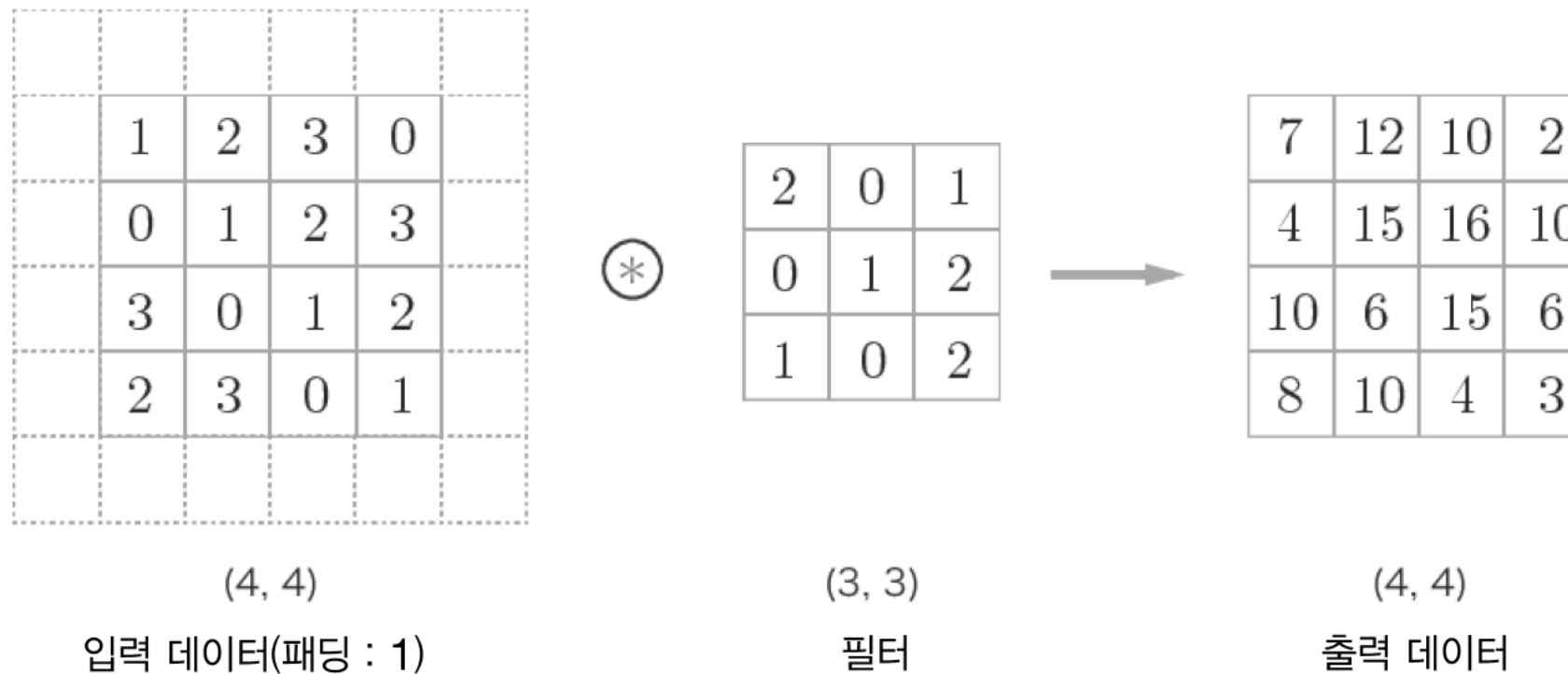
 $\otimes$ 

2	0	1
0	1	2
1	0	2

 $\rightarrow$ 

15	16
6	15

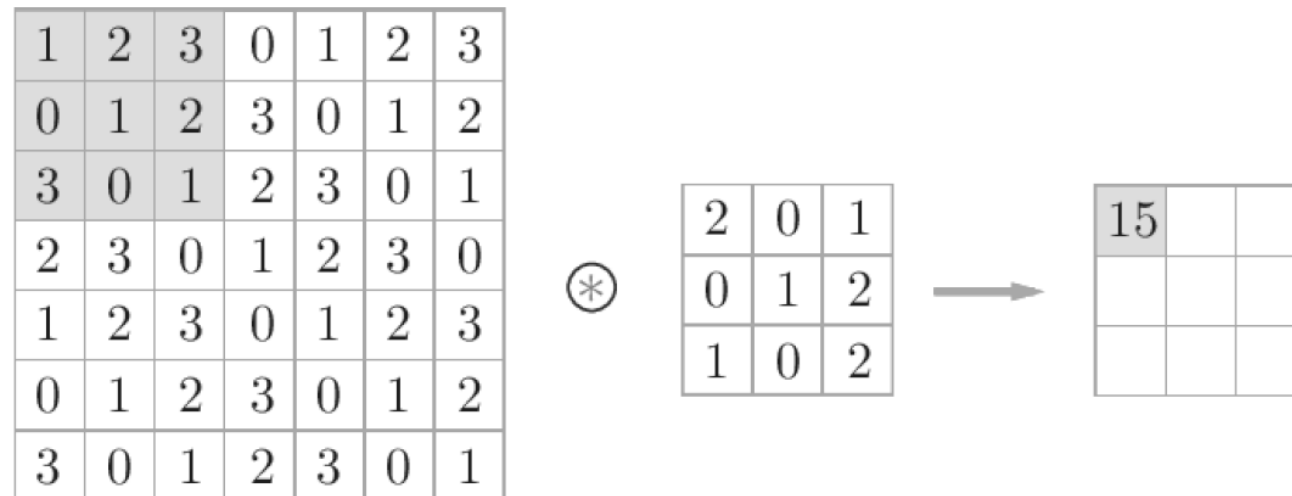
# Padding



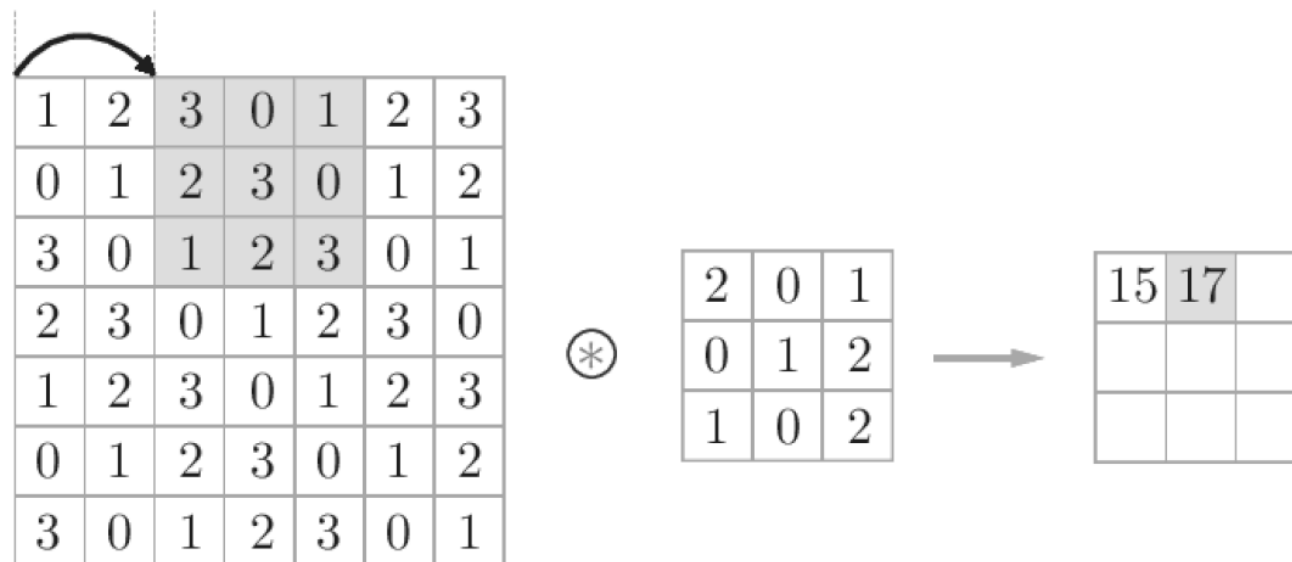
- 필터 적용후에 출력의 크기는 줄어듬
- 반복적인 필터 적용을 하면 출력의 크기는 1 x 1로 줄어듬
- 이를 방지하기 위해서 값이 0인 테두리를 추가함

# Strides

- 필터를 적용할때 움직이는 보폭



스트라이드 : 2



# 필터 적용 후 출력의 사이즈

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

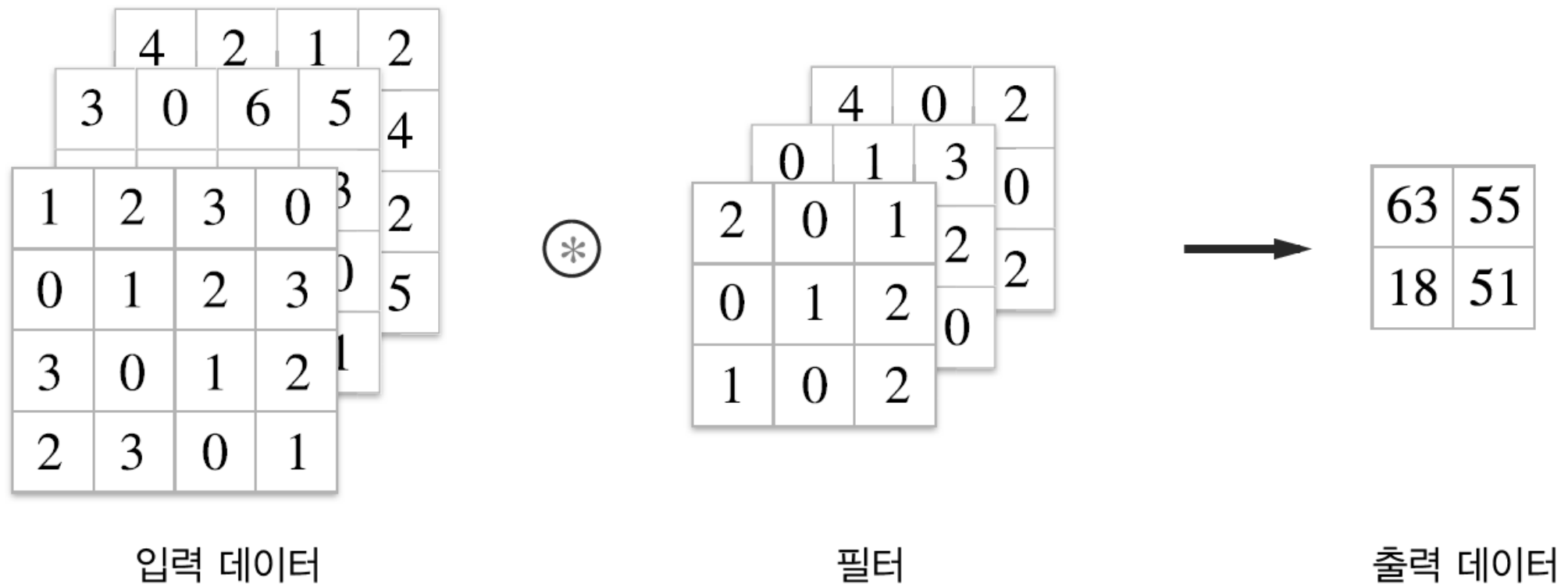
$$OW = \frac{W + 2P - FW}{S} + 1$$

H : 입력의 높이  
W : 입력의 폭  
P : Padding  
FH : 필터의 높이  
S : Strides  
OH: 출력의 높이  
OW: 출력의 폭

- input: (4,4) padding : 1, stride : 1, filter: (3,3)

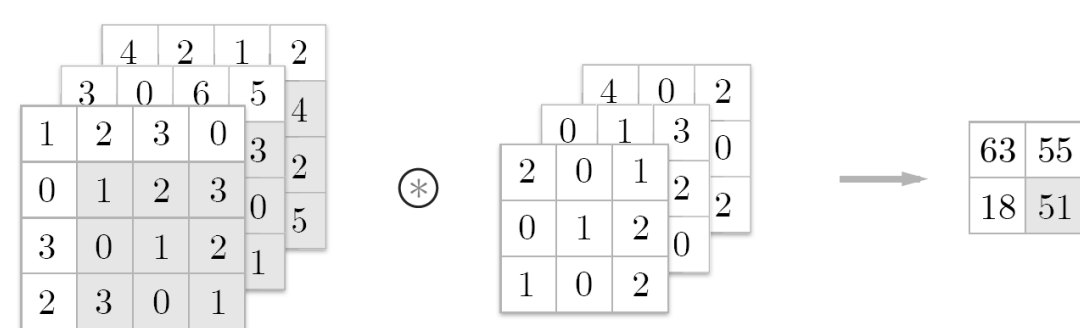
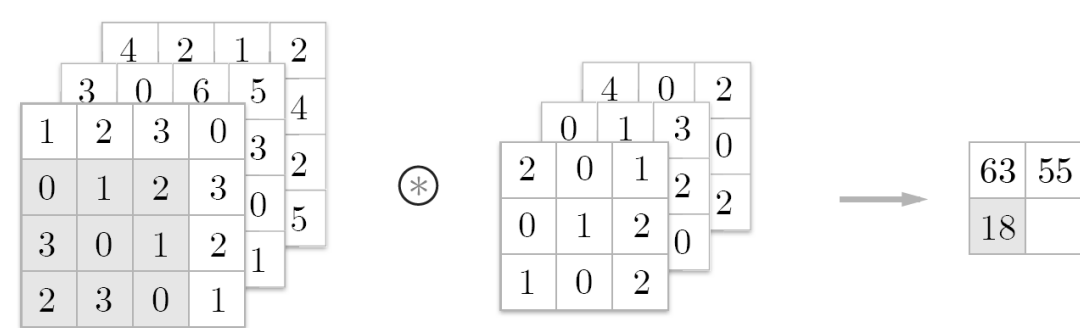
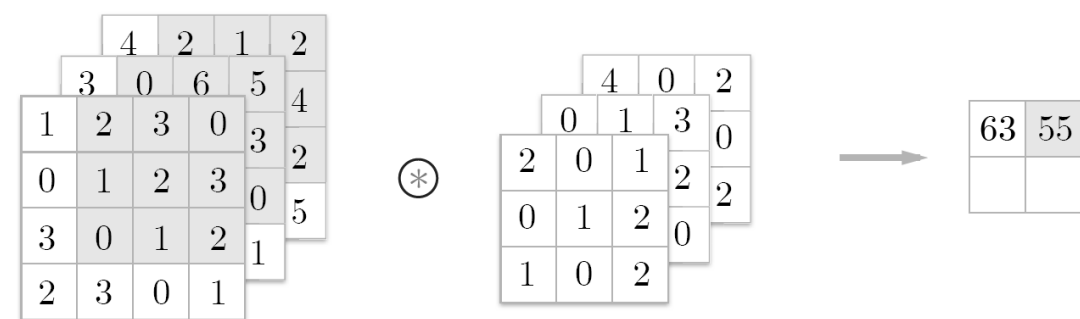
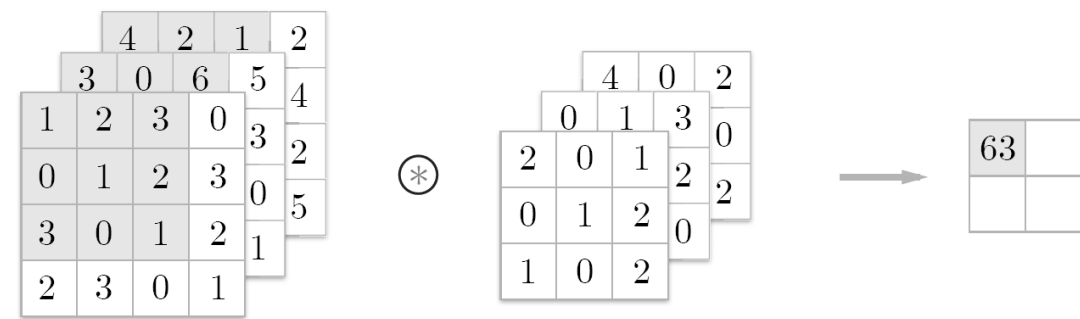


# 멀티 채널

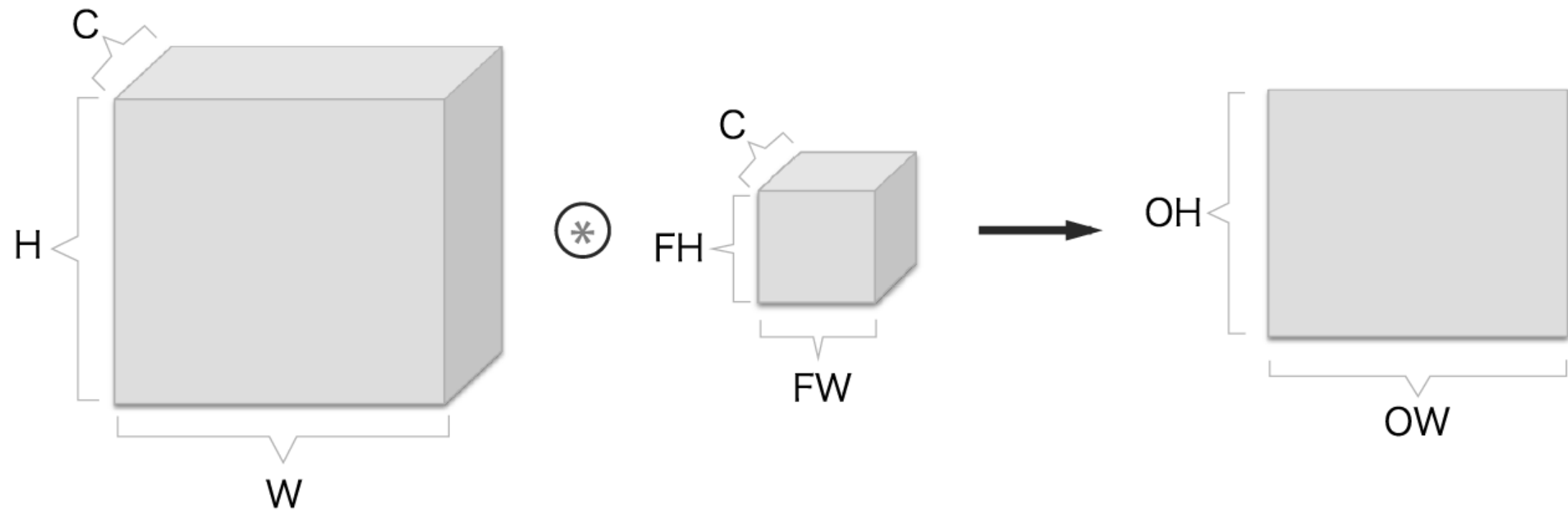


- 채널까지 고려한 3차원 데이터는 채널마다 합성곱을 수행하고, 그 결과를 합해서 하나의 출력을 얻음

# 멀티 채널



# 멀티 채널



$(W, H, C)$   
입력 데이터

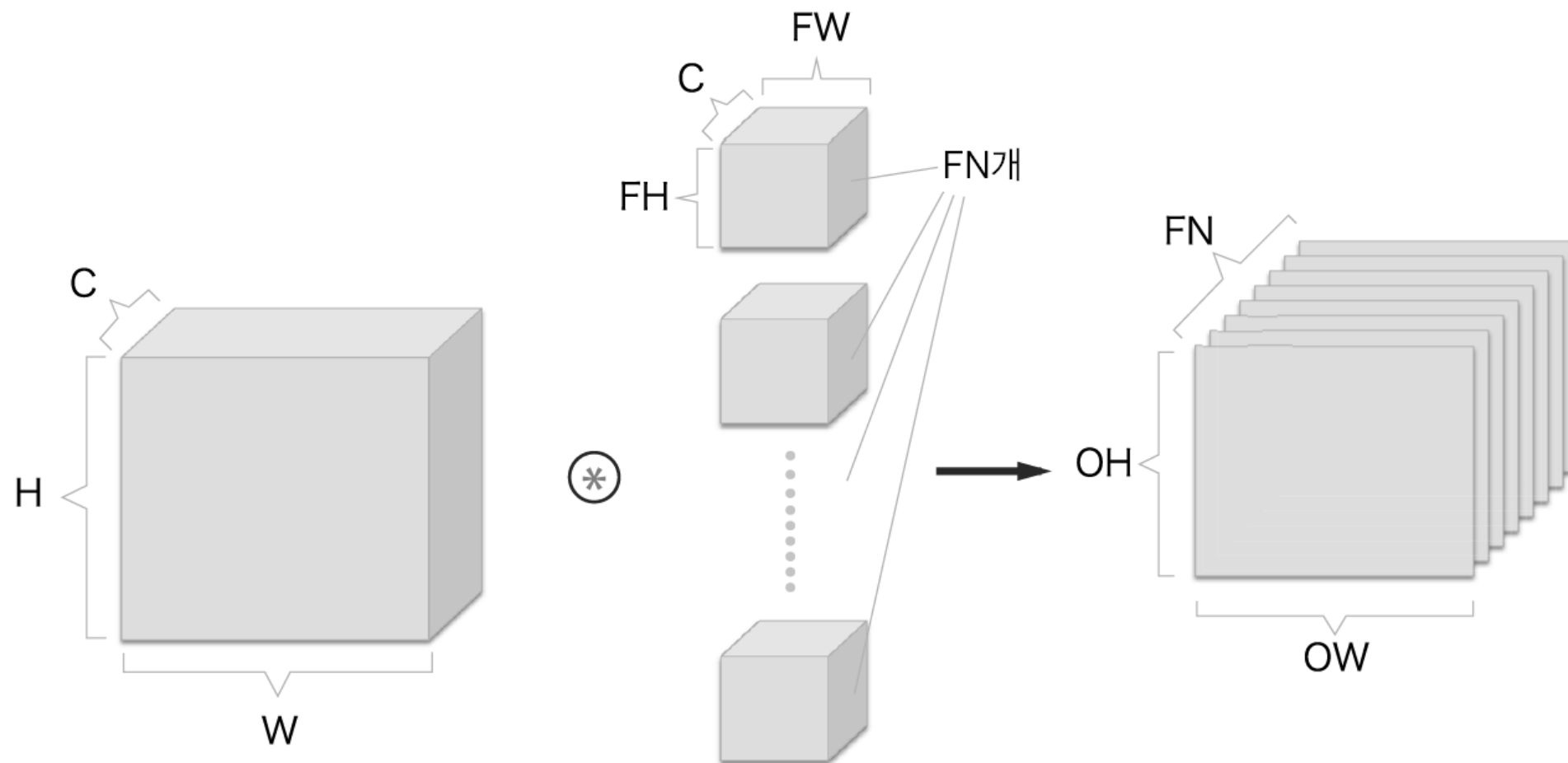
$\otimes$

$(FW, FH, C)$   
필터

$\longrightarrow$

$(1, OW, OH)$   
출력 데이터

# 멀티필터



$(W, H, C)$   
입력 데이터

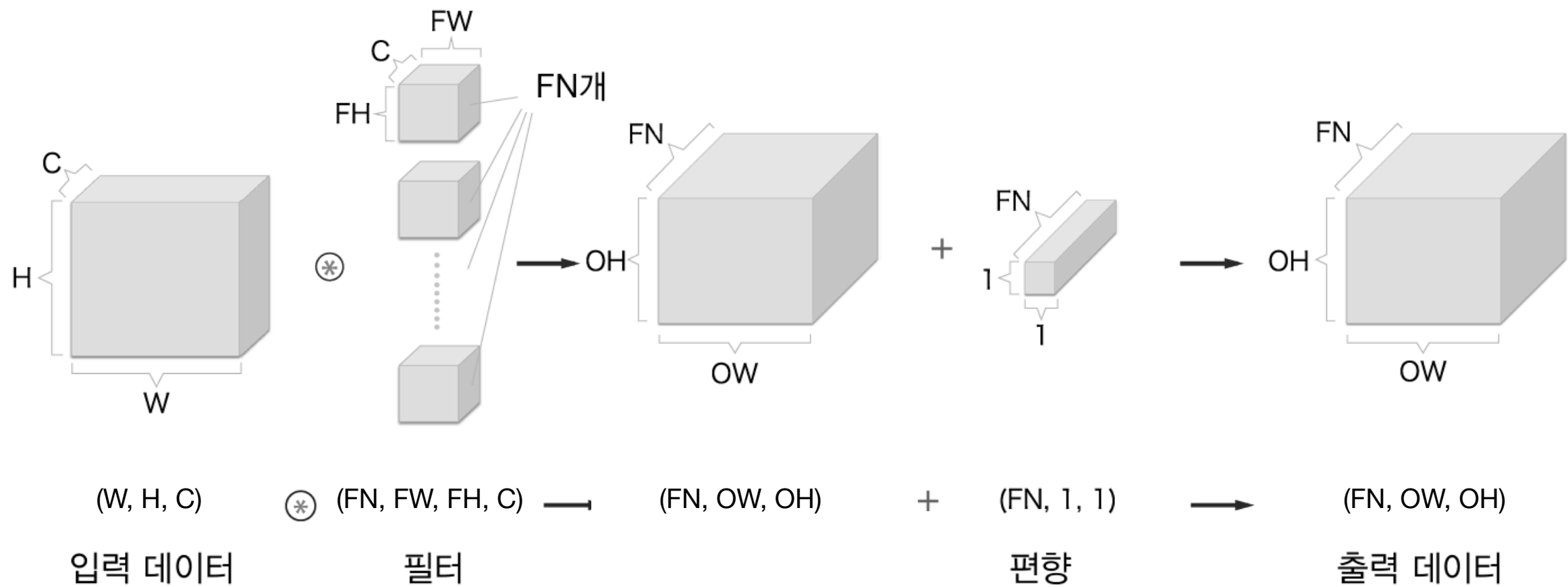
$\circledast$

$(FN, FW, FH, C)$   
필터

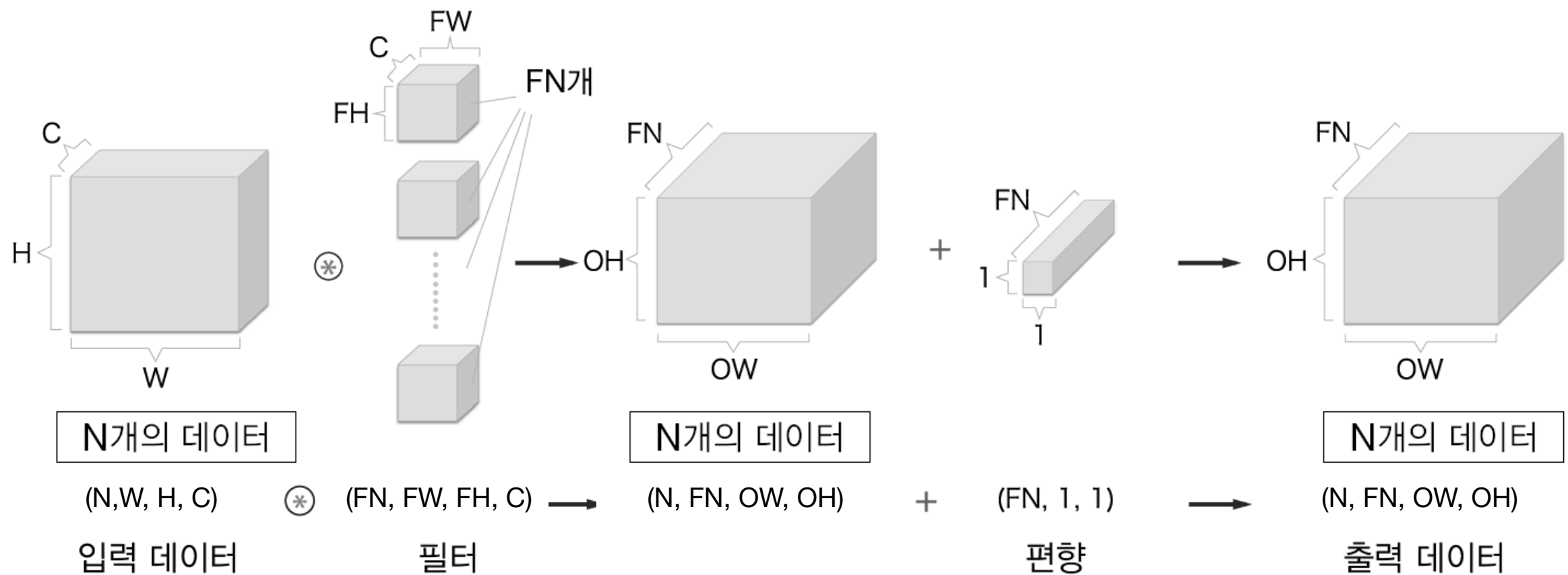
$\longrightarrow$

$(FN, OW, OH)$   
출력 데이터

# 멀티필터 + 편향



# N개의 데이터



# Pooling

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	

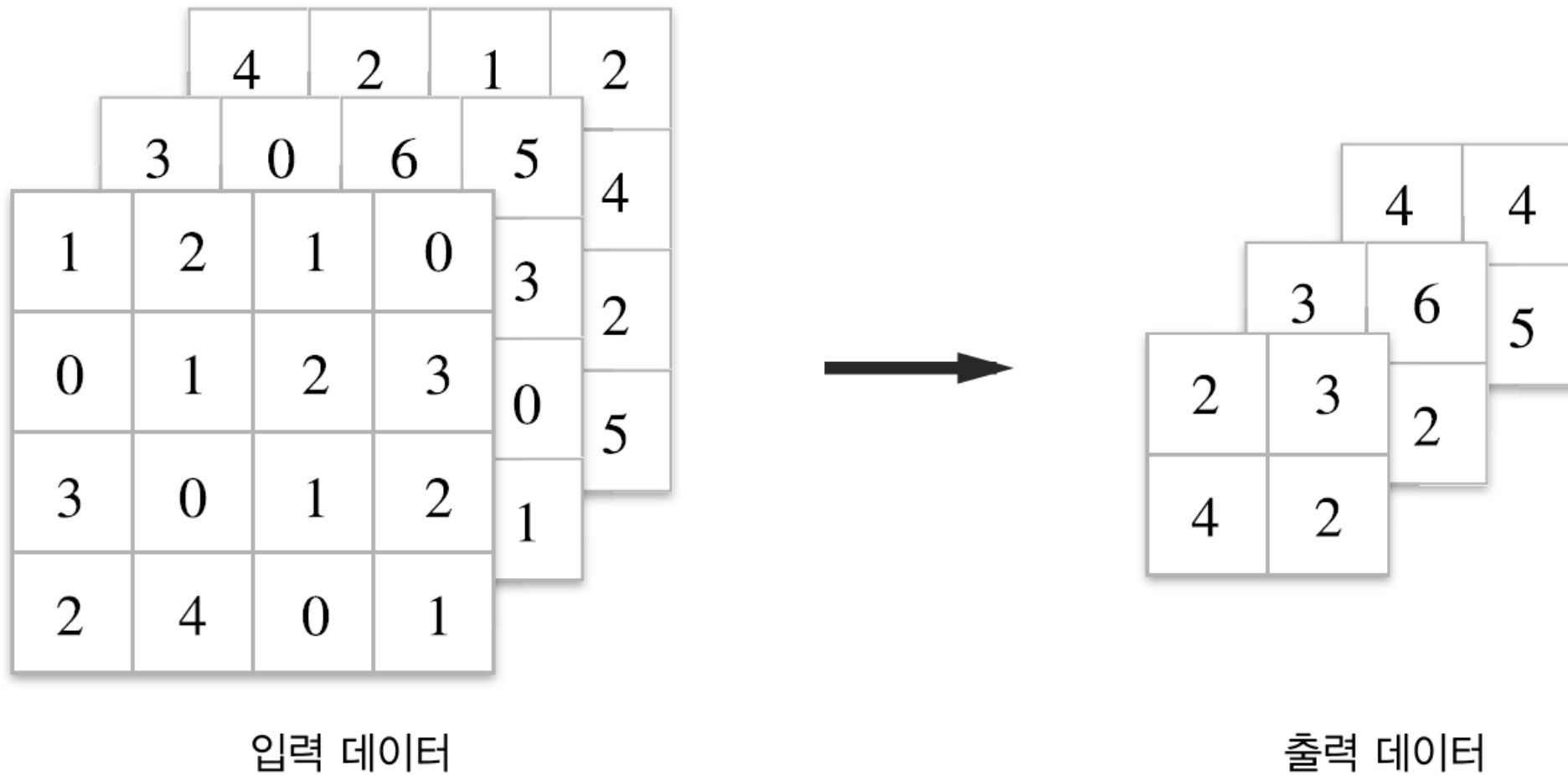
1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	2

- 풀링은 데이터의 크기를 줄이는 연산
- 풀링은 일반적으로 스트라이드 크기와 윈도우 크기를 같게 처리

# Pooling



- 풀링은 데이터의 크기를 줄이는 연산
- 풀링은 일반적으로 스트라이드 크기와 윈도우 크기를 같게 처리



# Convolution NN

