

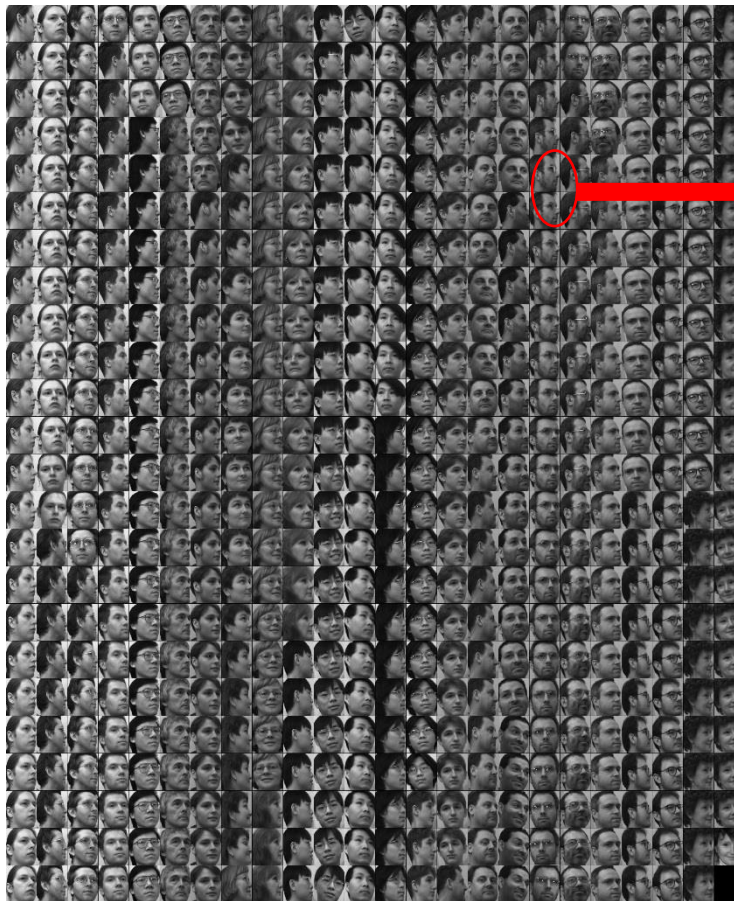
참고. 이미지 분류를 위한 CNN

1. 이미지 데이터: 디지털 이미지의 값을 2차원 수치로 표현
2. CNN: 이미지 분류를 위한 딥러닝 기법으로 합성곱 신경망이라고도 하며 이미지 분류에 잘 활용됨
3. CNN 과정: 필터를 스트라이드하는 Convolution과정과, 이 결과에서 Feature값을 줄이는 Pooling 과정을 거쳐 이미지를 인공신경망에 적용
4. CNN 단계별 절차: CNN은 입력 이미지의 공간정보를 유지한 상태에서 Convolution과 Pooling을 반복한 후 인공신경망을 통해 분류하는 과정을 거침
5. 주요 하이퍼 파라미터: Convolution의 필터의 수, 필터의 크기, Padding, Stride 등이 CNN의 주요 하이퍼 파라미터임.
6. Convolution과 Pooling의 과정: Convolution과정을 통해서는 필터의 수 만큼 Feature Map이 생성되고, Pooling을 통해서는 주요 Feature들이 강화 됨.

참고. 이미지 분류를 위한 CNN

딥러닝, *얼굴 인식? 영상 분석? 의료 이미지?*

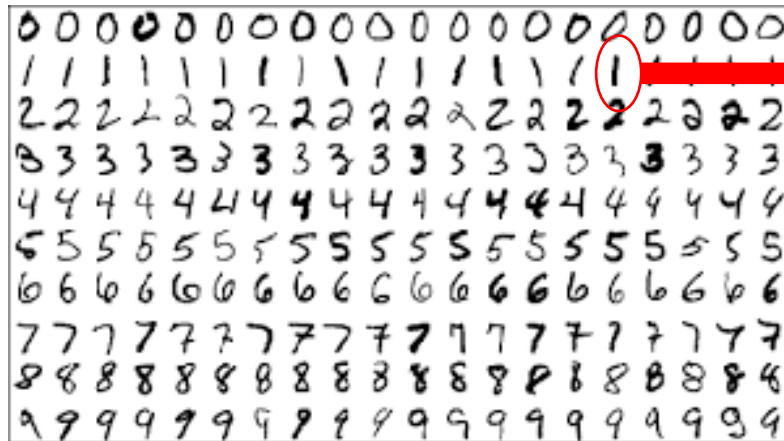
- Olivetti Faces Data set
- Face images from 1992~1994 at AT&T Lab



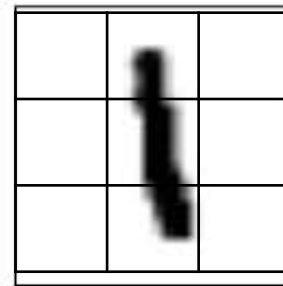
Label, v1,v2,v3, v4,v5,v6,... v782,v783,v784,...
"30", 0, 1, 0, 0, 1, 0,... 0, 1, 0,...

이미지 예시

9화소(3X3) 디지털카메라!

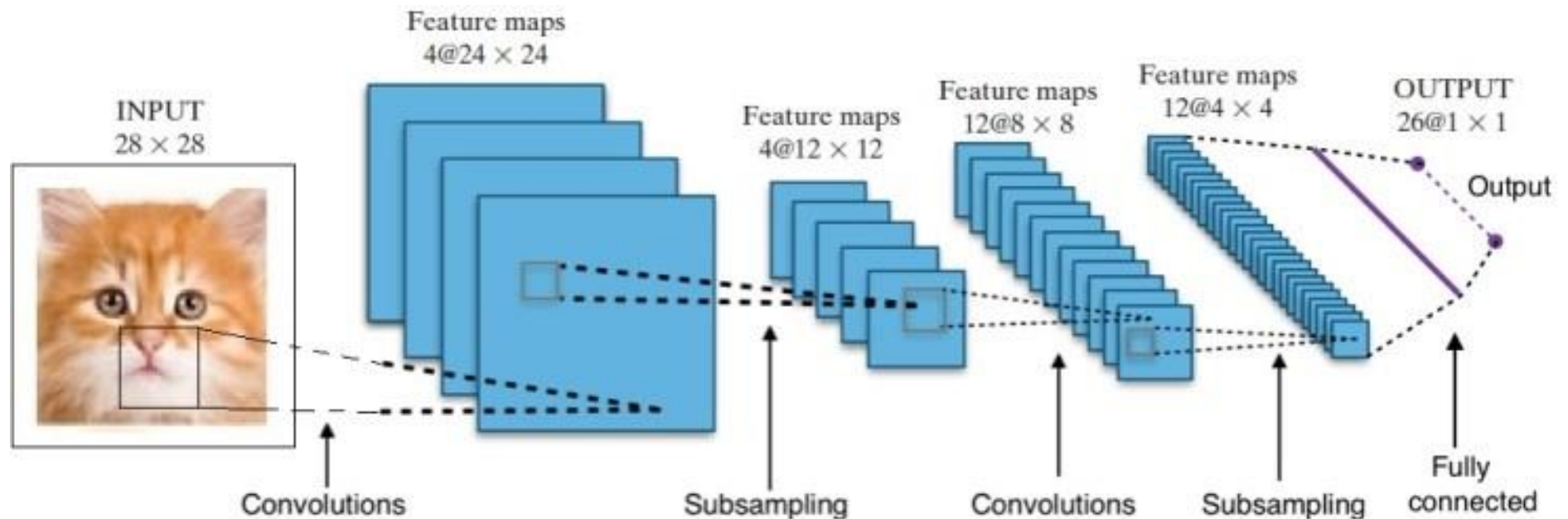


MNIST Hand written digits



0	254	0
0	254	0
0	254	0

Convolution Neural Network 합성곱 신경망 이미지 인식을 위한 인공신경망



CNN은 ...

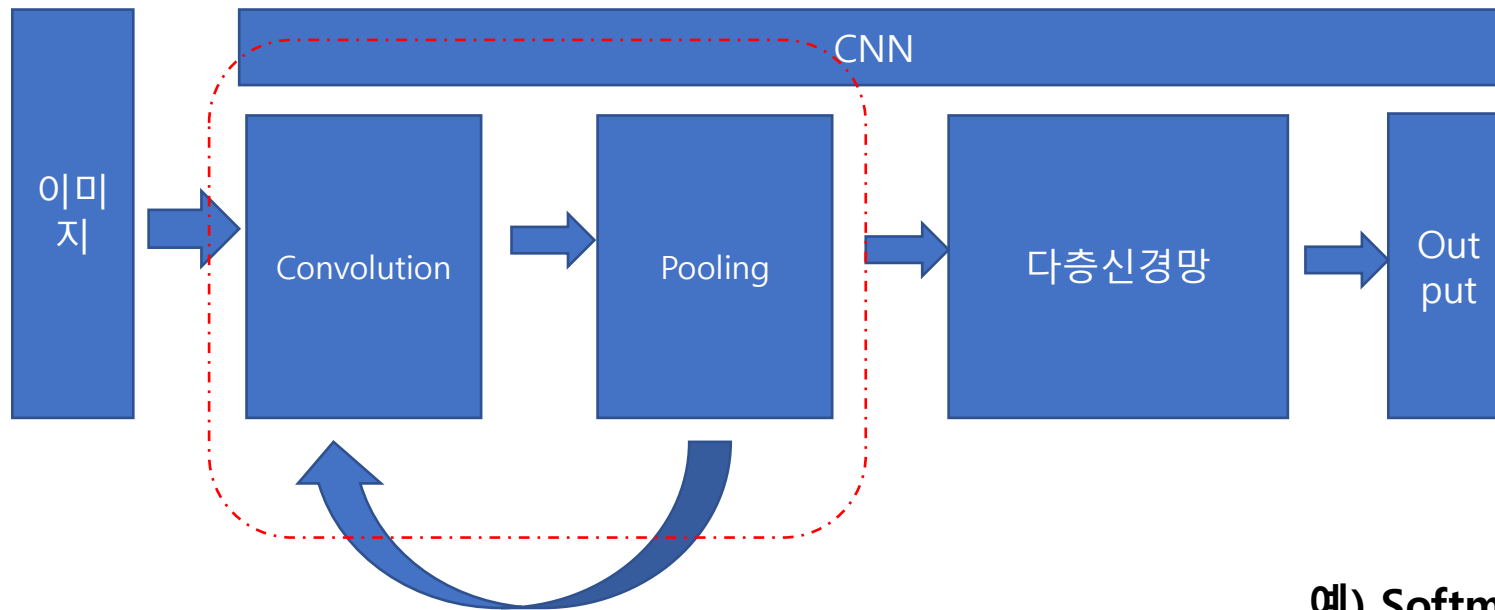
- Convolution과 Pooling을 반복하여 적용해서 Feature를 추출하고,
- 발견된 Feature로 인공신경망을 적용하는 기법
- 이미지의 공간 정보를 유지한(=1차원으로 평면화 하지 않은) 딥러닝!



이미지 처리에 많이 활용!

-객체 인식
-얼굴 인식
-자율 주행

CNN의 주요 단계



예) Softmax 활성화함수
:출력을 0~1사이 값
:출력의 합은 1

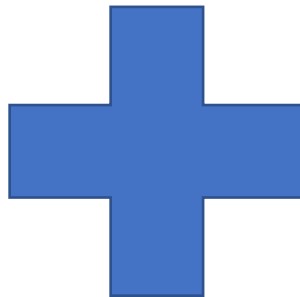
CNN 주요 하이퍼 파라미터

Convolution Filter의 수

Filter의 크기

Padding 여부

Stride



인공신경망의

기본적인

하이퍼 파라미터

CNN 주요 하이퍼 파라미터

- Filter: 원래 입력 이미지에서 주요 Feature를 탐지하는 역할, Feature Map이 나오게 됨
- Convolution Filter의 수: 원래 이미지로 부터 Filter의 수 만큼 다른 방식으로 Feature Map을 탐지
- Padding 여부: Padding은 입력 데이터보다 출력이 작아지는 것을 방지, 주로 원래 이미지 주변을 0으로 채워 Filter가 이미지 첫 시작점부터 작동할 수 있도록 함.
- Stride: Filter가 움직이는 간격

멀티채널과 단일채널

멀티채널(칼라): $R+G+B$

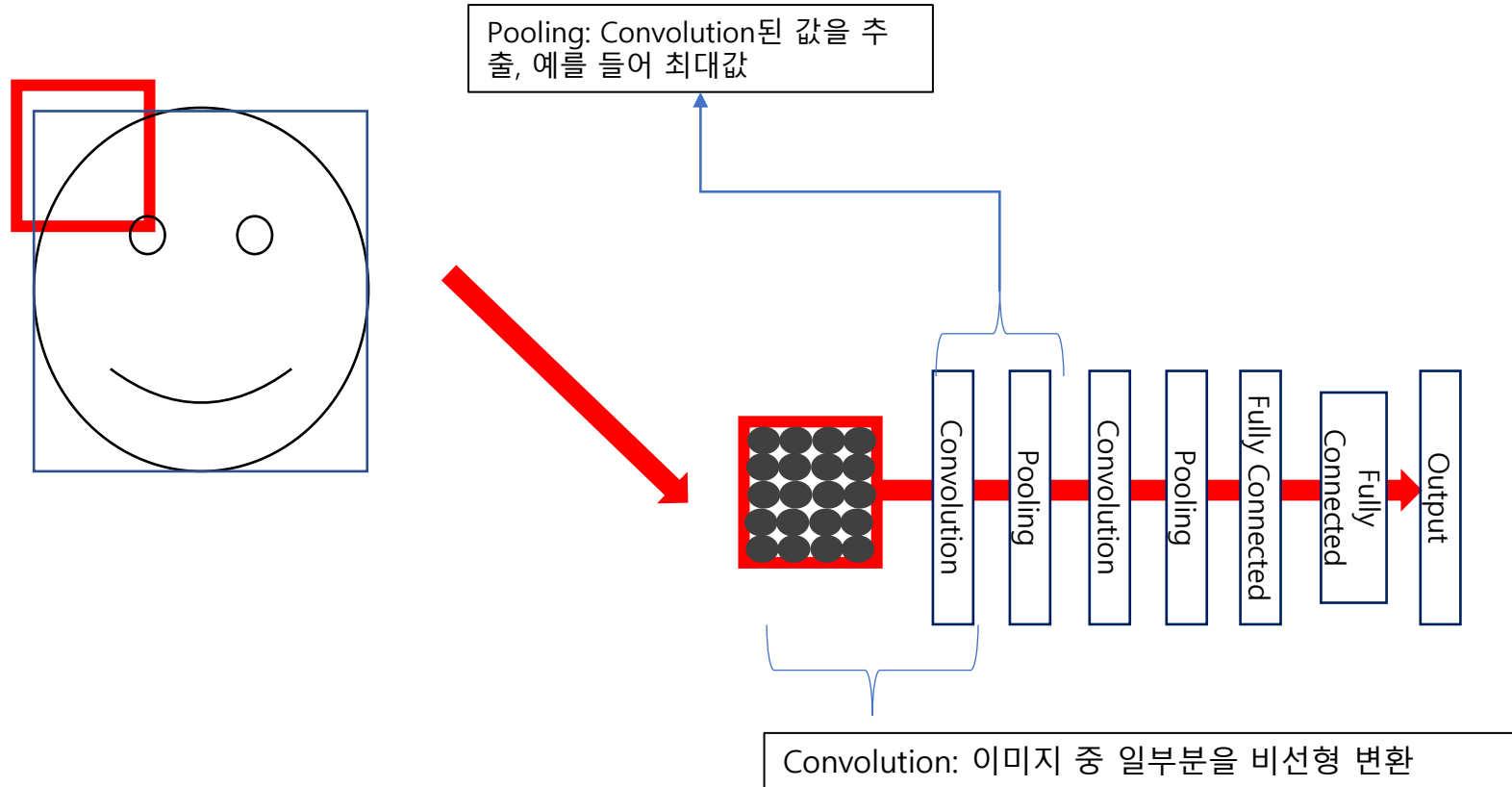
단일채널(흑백)

- 생성되는 Convolution결과의 수
- 흑백: Filter의 수와 같은 Feature Map
- 칼라(R,G,B인 경우): Filter의 수와 같은 Feature Map X 3

Convolution과 Pooling

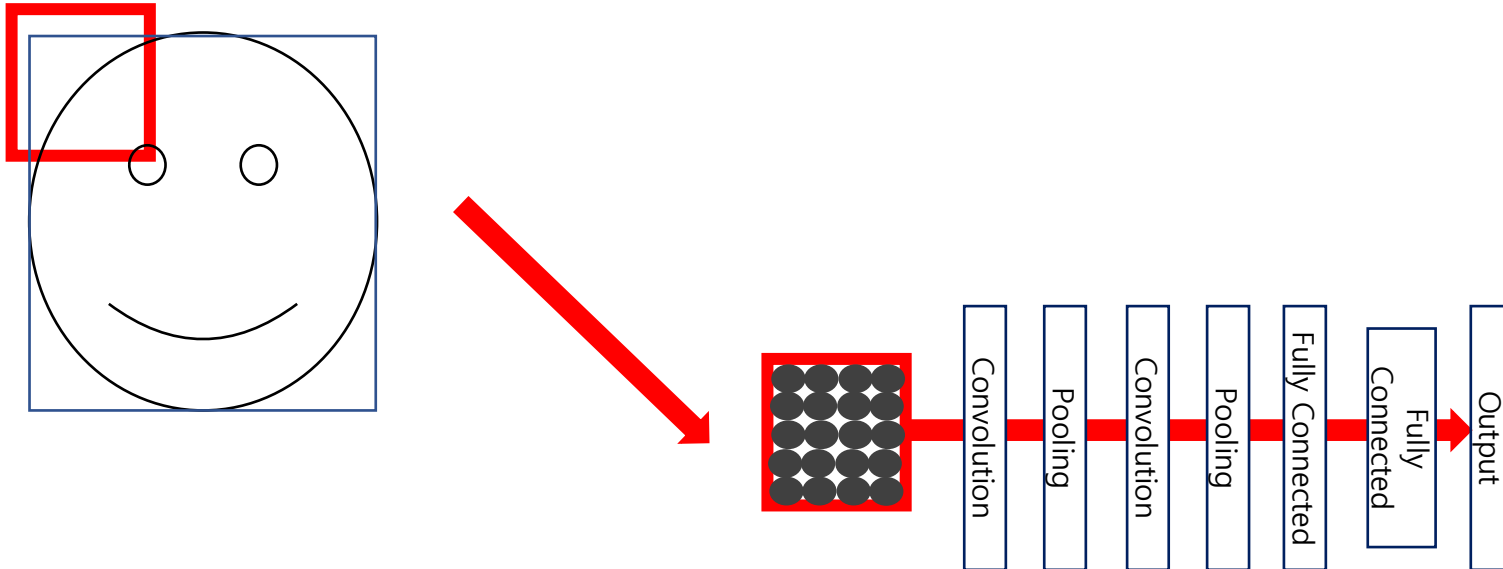
- Convolution:
 - 이미지의 주요 Feature를 다양한 관점의 Filter에 따라 추출 하이퍼파라미터를 지정
- Pooling:
 - Feature Map에서 특정 Feature를 강조
 - Pooling은 하이퍼파라미터 없음
 - Pooling을 해도 채널의 수는 동일하지만, 행렬 크기는 감소

Filter의 적용



Convolution은 필터를 통해 이미지 일부를 새로운 값으로 계산

Stride



Convolution 과정

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

Convolution!

4		



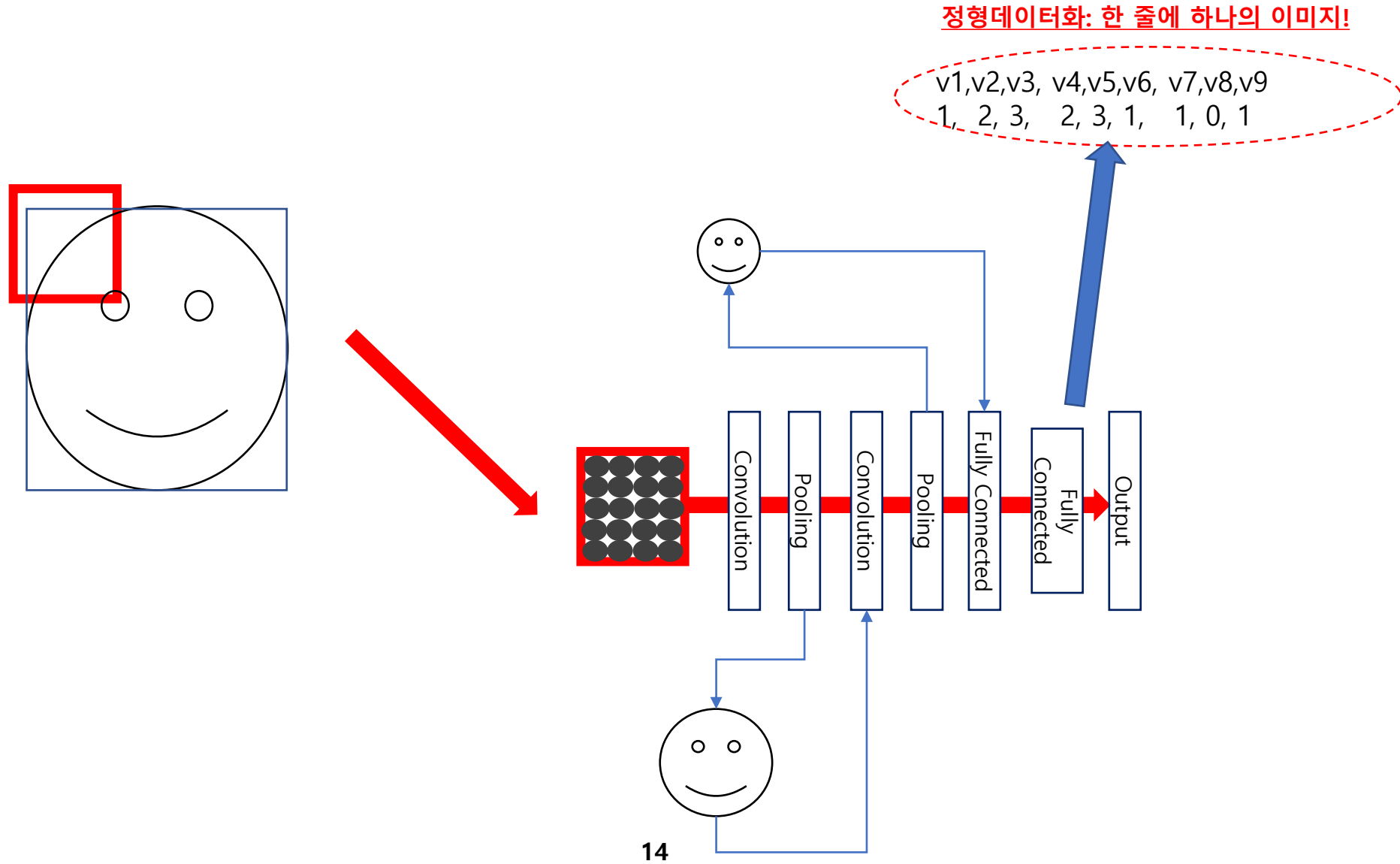
4	3	4
2	4	3
2	3	4



Max pooling

4

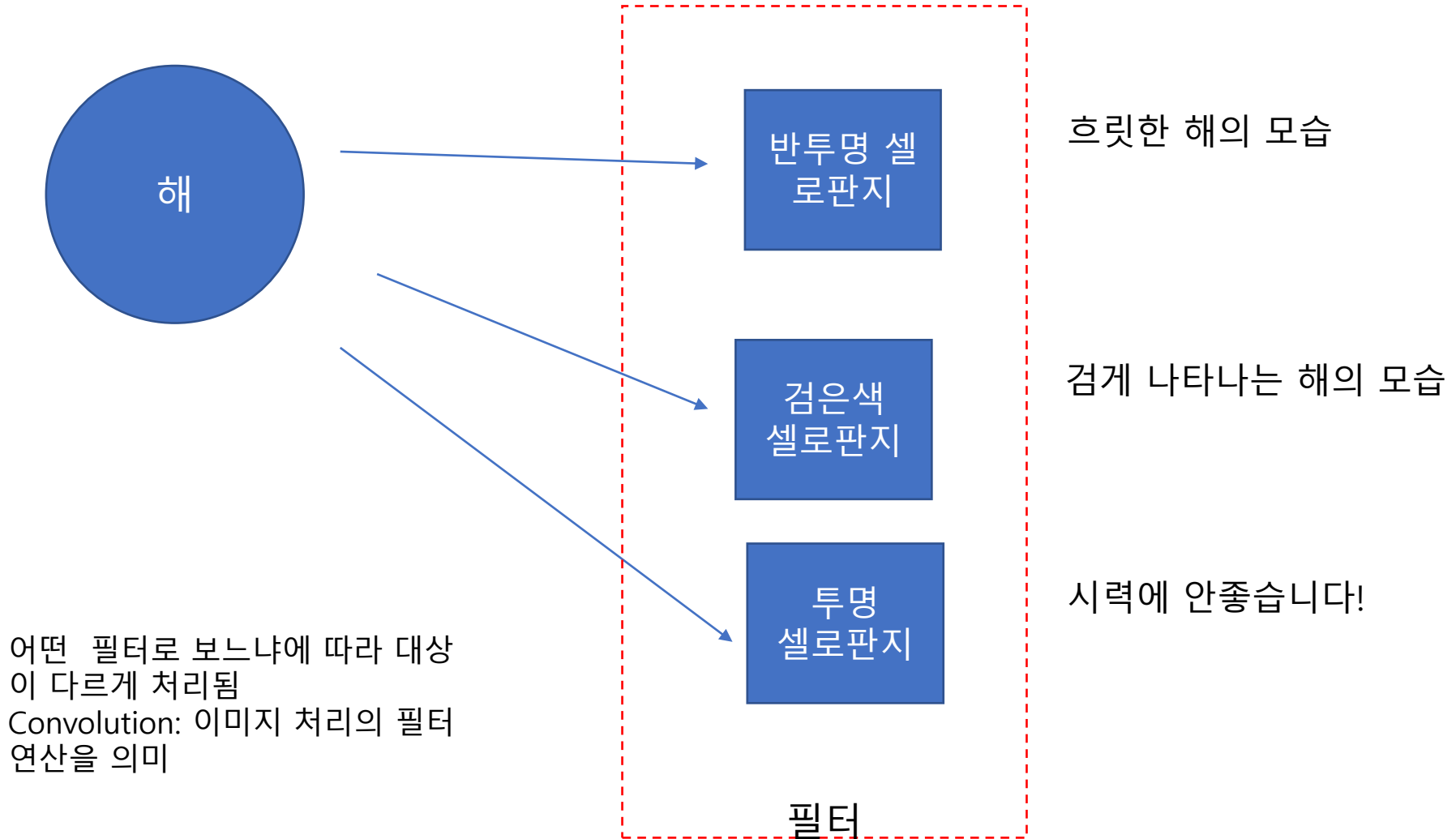
Convolution 과정



CNN의 특징은!

- 입력 데이터의 공간 정보를 유지 = 2차원(또는 3차원)입력데이터를 사용
- Feature Map을 사용 = Feature Map은 입력데이터, 은닉층으로 전달되고, 은닉층에서 계산되는 값을 의미
 - **합성곱(Convolution) 연산을 수행!**

Convolution



Convolution

어떤 필터로 보느냐에 따라 대상이 다르게 처리됨
Convolution: 이미지 처리의 필터(=Kernel) 연산을 의미

1	2	3
0	2	1
2	0	3

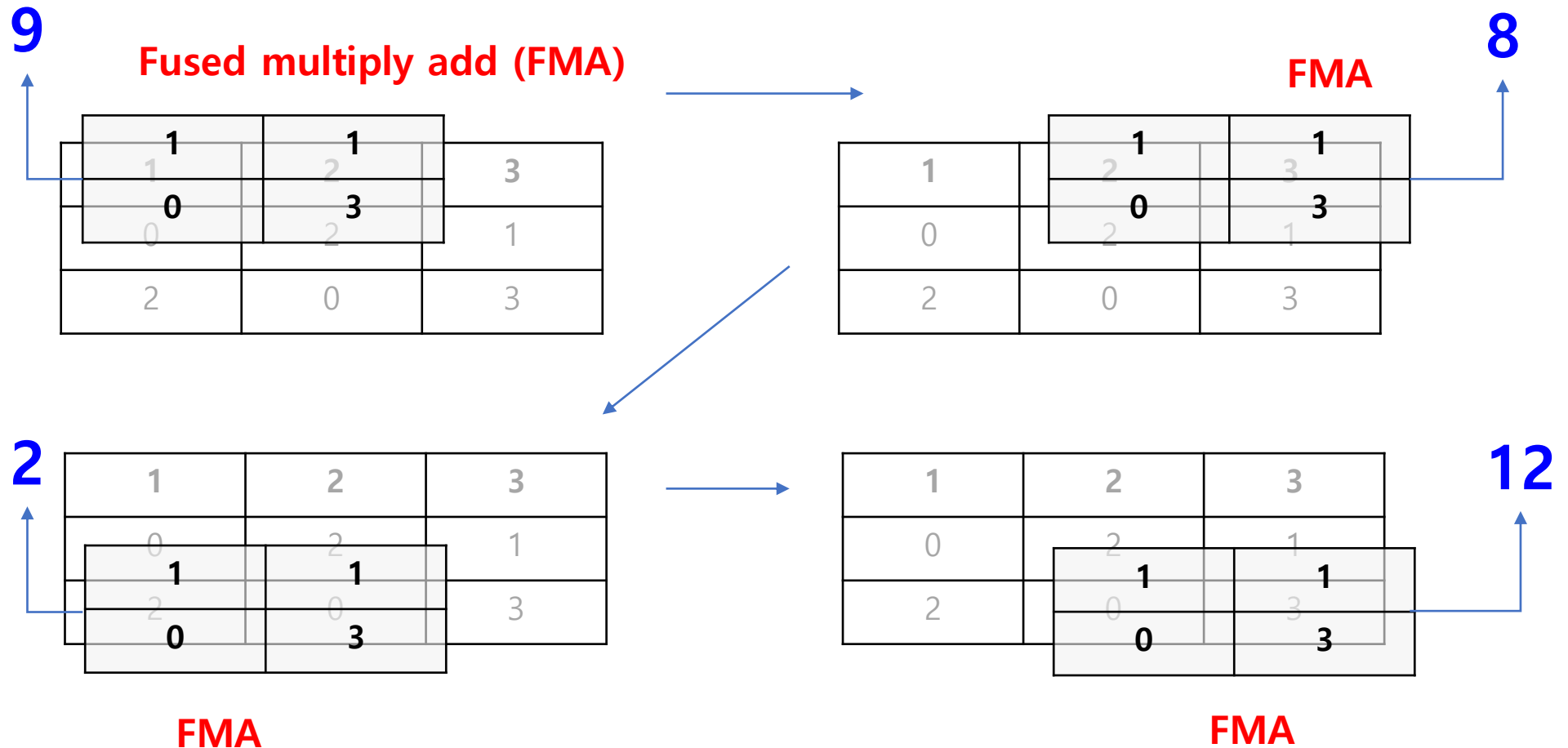
이미지

1	1
0	3

필터

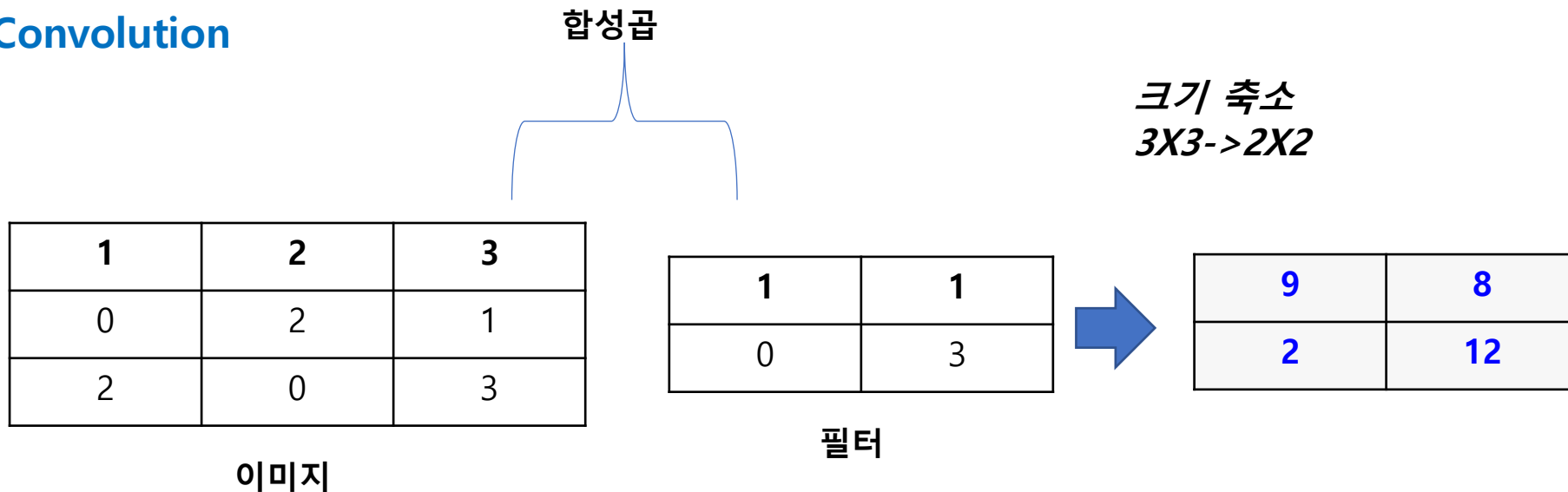
Convolution

한 칸씩 움직임=Stride의 크기

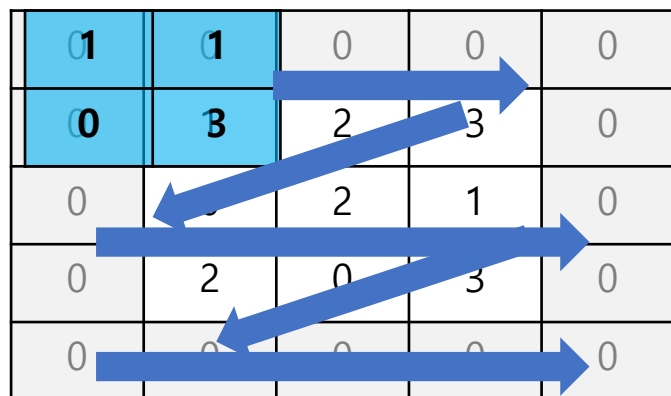


참고. 이미지 분류를 위한 CNN

Convolution



패딩: 이미지 주위를 특정값(예: 0)으로 둘러쌓기



이미지

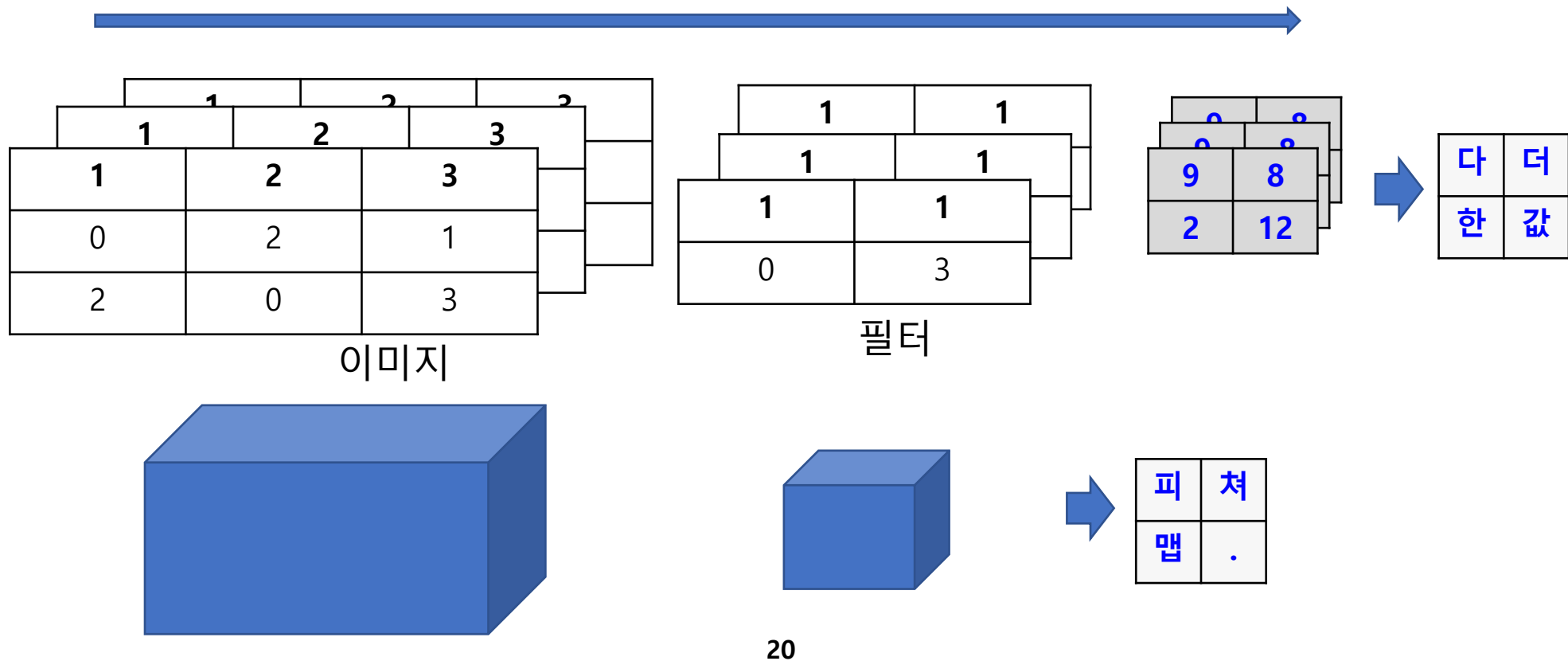
$$\text{결과 높이} = \frac{\text{이미지 높이} + 2 \times \text{패딩 크기} - \text{필터 높이}}{\text{스트라이드}} + 1$$

$$\text{결과 넓이} = \frac{\text{이미지 넓이} + 2 \times \text{패딩 크기} - \text{필터 넓이}}{\text{스트라이드}} + 1$$

3차원 데이터의 합성곱

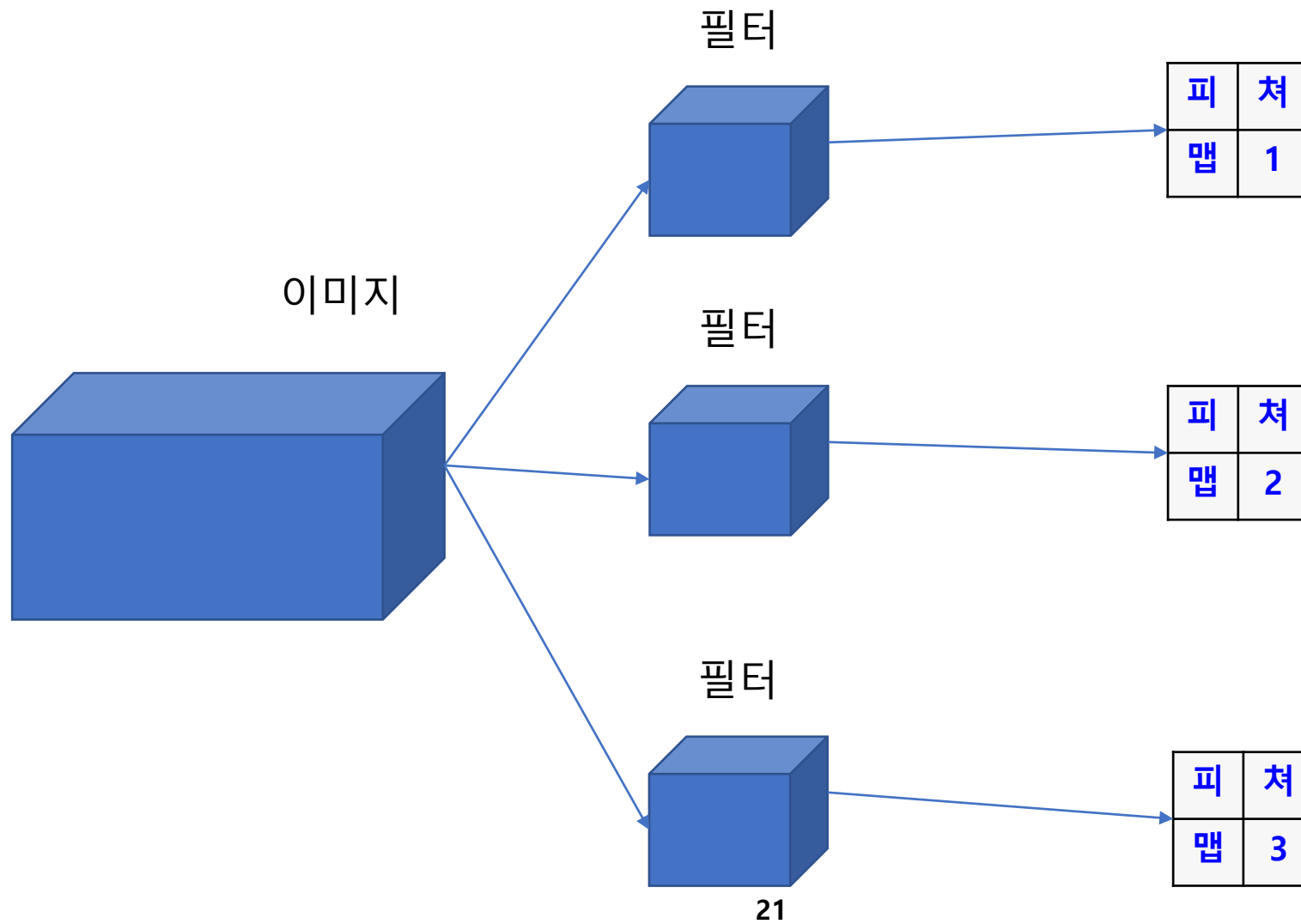
어떤 필터로 보느냐에 따라 대상이 다르게 처리됨
Convolution: 이미지 처리의 필터(=Kernel) 연산을 의미

채널별로 합성곱 연산



참고. 이미지 분류를 위한 CNN

3차원 데이터의 합성곱: 여러 필터!



Pooling: 2차원 공간을 줄여주는 연산

- Pooling 절차: Convolution의 결과들의 공간을 줄여주며 주요 값을 더 강조할 수 있음
- Pooling의 여러 방법: Max Pooling과 Average Pooling등이 있으며, 이미지 처리에서는 Max Pooling을 많이 사용함
- Pooling의 특징: 하이퍼파라미터 학습이 없고, 입력값의 변화에 강건한 특징이 있음

1	2	3
0	2	1
2	0	3

이 피쳐맵을 줄여보기!

Pooling: 2차원 공간을 줄여주는 연산: 2X2 영역별로 최대값을 구하기

2

1	2	3
0	2	1
2	0	3

3

1	2	3
0	2	1
2	0	3

2

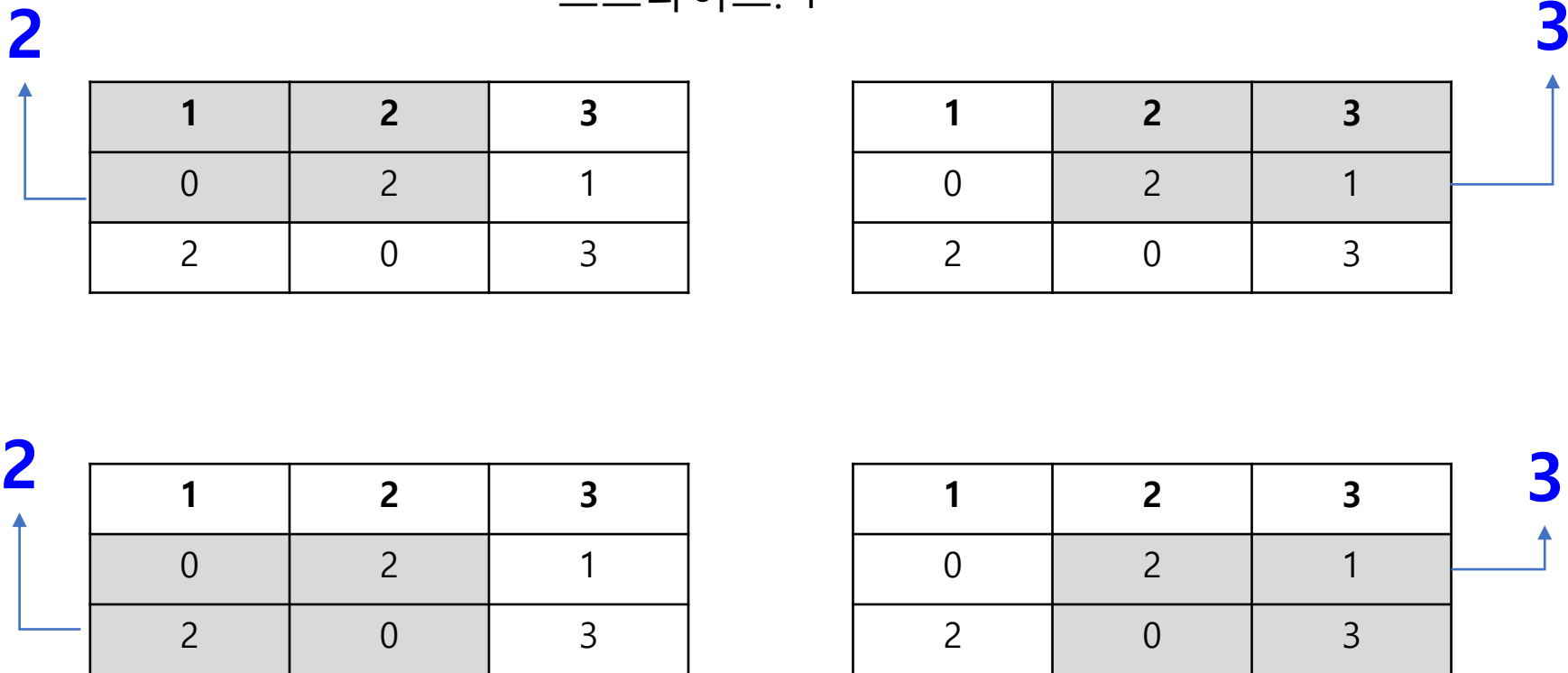
1	2	3
0	2	1
2	0	3

3

1	2	3
0	2	1
2	0	3

Pooling: 2차원 공간을 줄여주는 연산: 2X2 영역별로 최대값을 구하기

스트라이드: 1

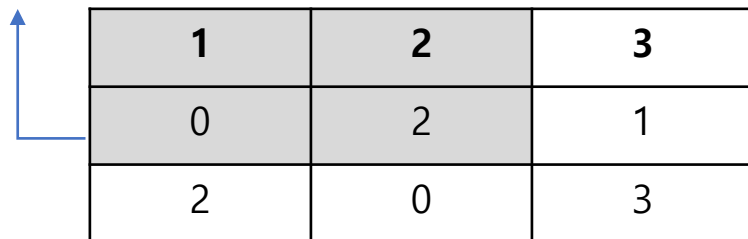


2X2 Max Pooling

Pooling: 2차원 공간을 줄여주는 연산: 2X2 영역별로 최대값을 구하기

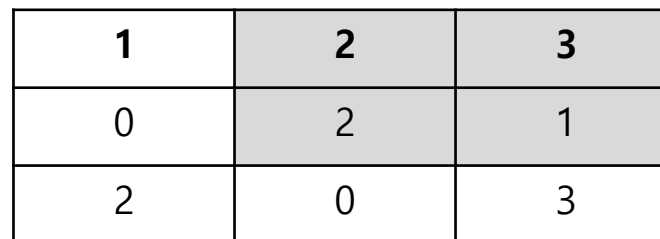
스트라이드: 1

1.25



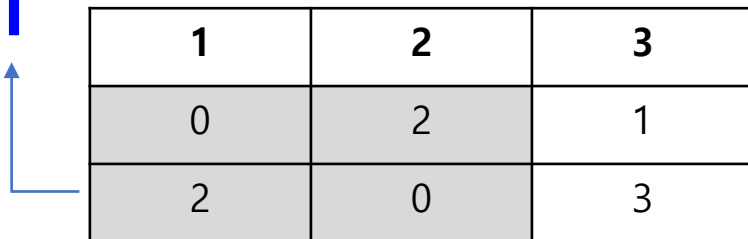
1	2	3
0	2	1
2	0	3

2



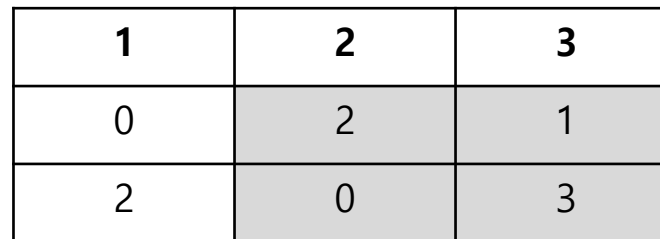
1	2	3
0	2	1
2	0	3

1



1	2	3
0	2	1
2	0	3

1.5



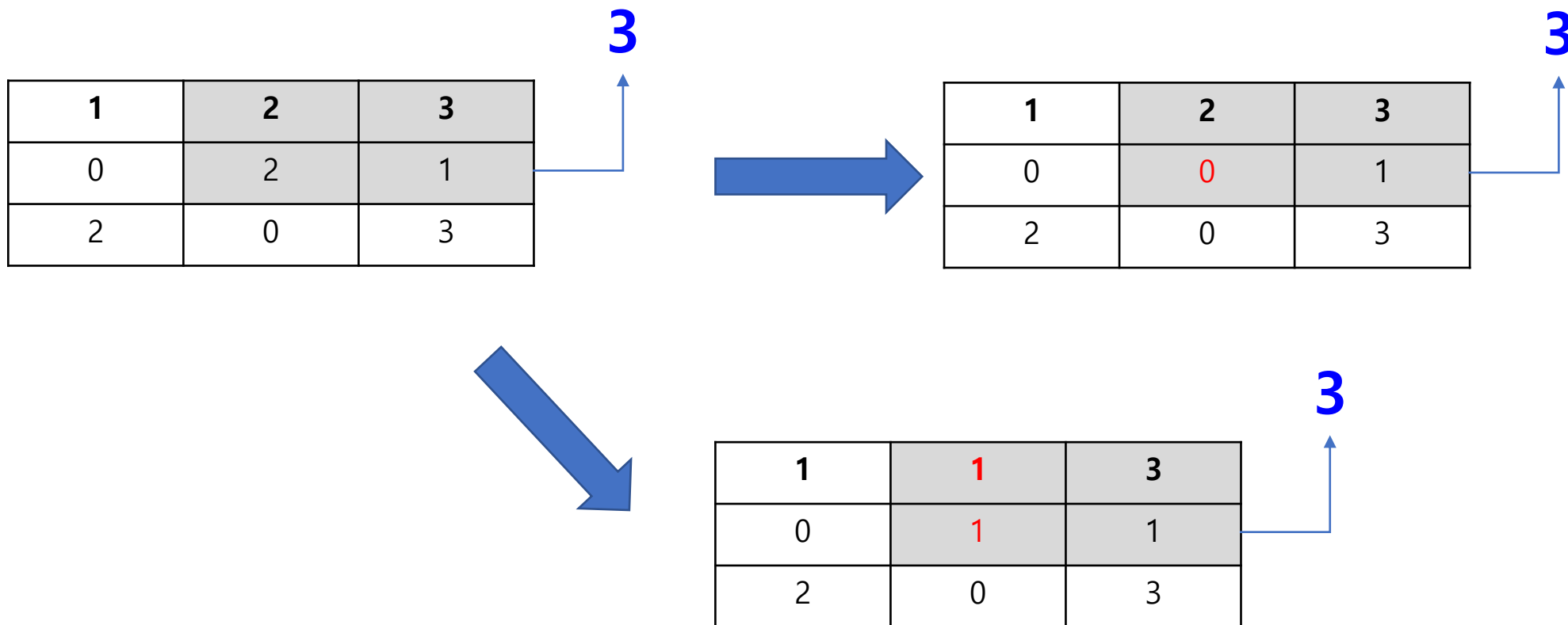
1	2	3
0	2	1
2	0	3

2X2 Average Pooling

Pooling

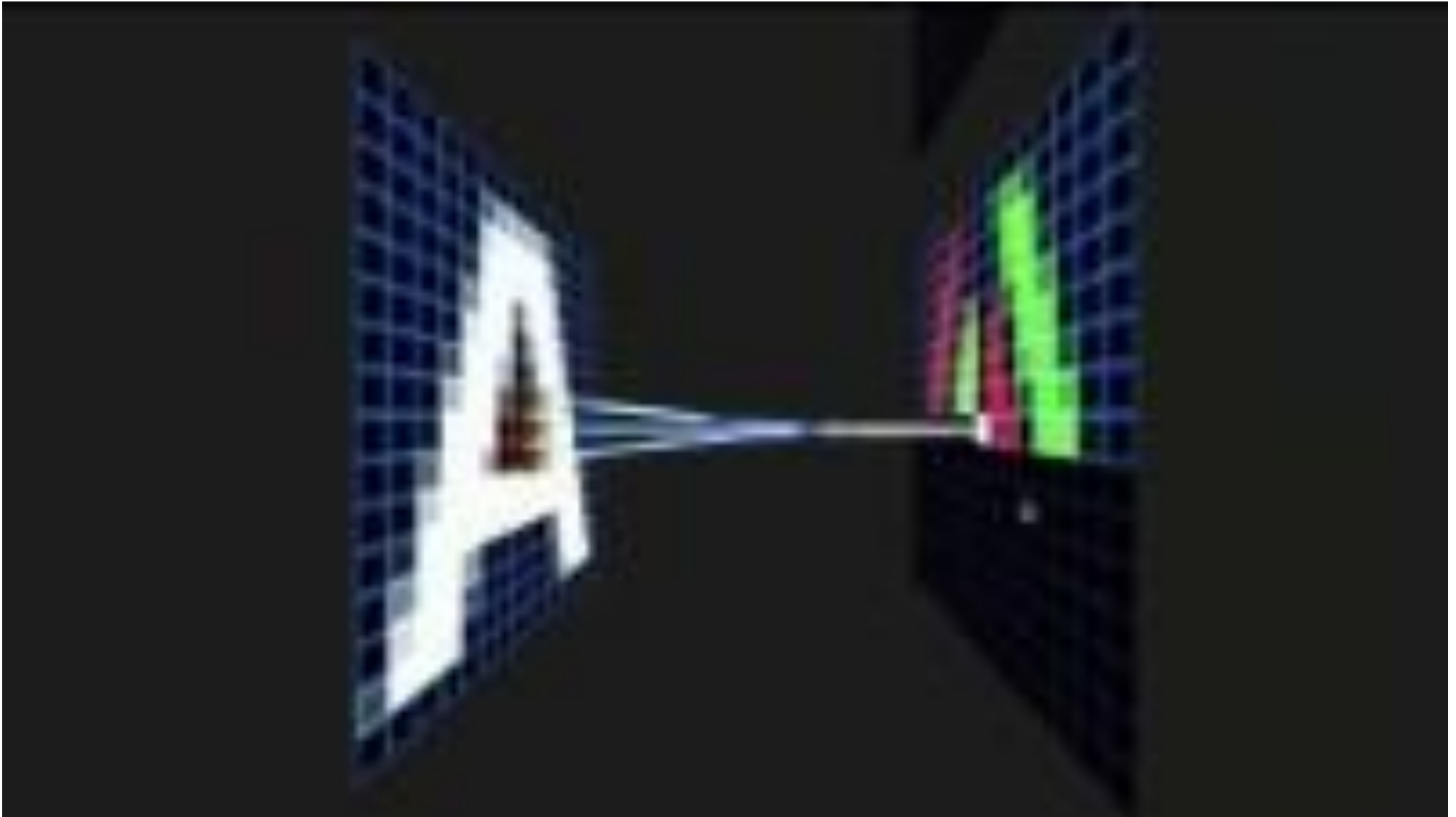
- 학습할 하이퍼 파라미터 없음
- 채널수 고정
- 입력의 변화에 영향을 적게 받음(Robust)

Pooling의 강건함: 입력값의 변화를 Pooling이 흡수, Robust한 결과를 제공



2X2 Max Pooling

이미지 처리에서는 주로
Max Pooling 사용!



<https://www.youtube.com/watch?v=f0t-OCG79-U>