

CNN 기본원리

2020.03.24 Hanyang univ. AILAB 정지은

INDEX.

1. CNN 등장 배경
2. Convolution 직관적 이해
3. Convolution 수식의 이해
4. CNN 구조의 이해
5. CNN 관련 용어 : Channel, Stride, Padding ...

CNN 왜 생겼지?

CNN 등장배경

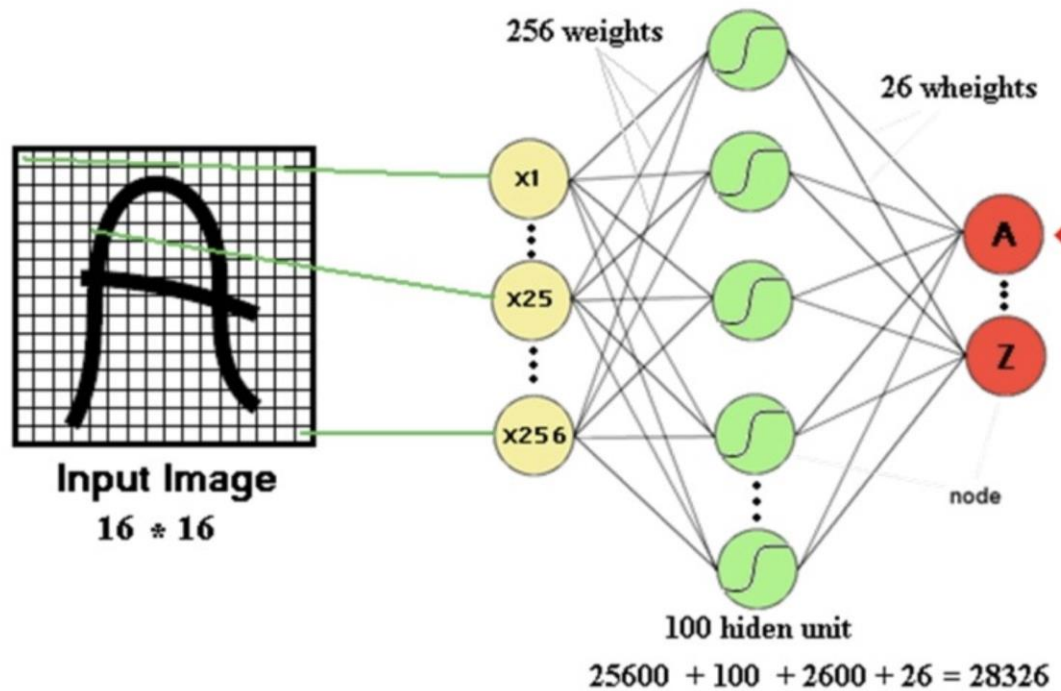
Zip code 필기체를 인식하기 위한 프로젝트에서 첫 등장 "Backpropagation applied to handwritten zip code recognition" (LeCun, 1989)

필기체 인식 문제를 Fully-connected Neural Network 로 풀어보면?

CNN 등장배경

Zip code 필기체를 인식하기 위한 프로젝트에서 첫 등장 "Backpropagation applied to handwritten zip code recognition" (LeCun, 1989)

필기체 인식 문제를 Fully-connected Neural Network 로 풀어보면?



할 수는 있다!

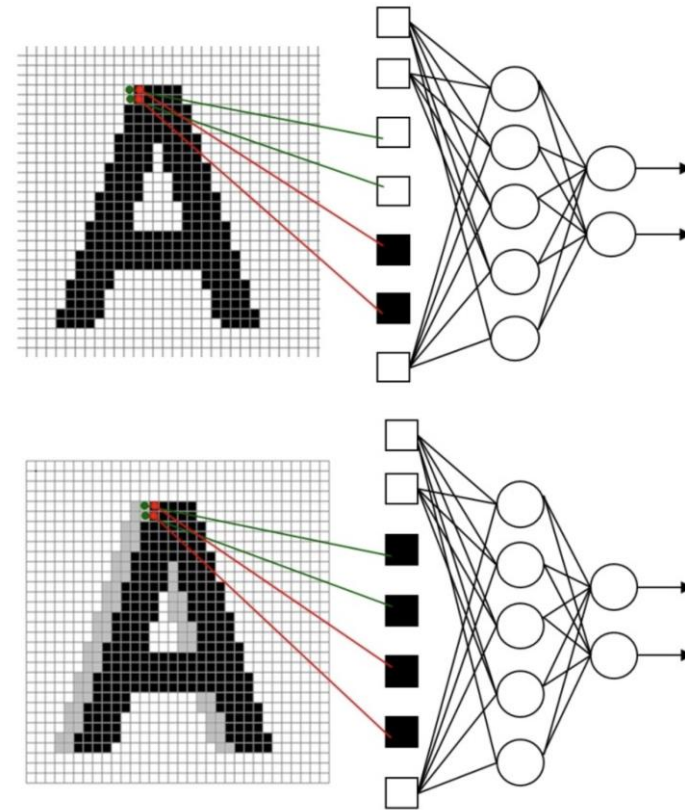
그런데 매우 비효율적이다.

CNN 등장배경

Zip code 필기체를 인식하기 위한 프로젝트에서 첫 등장 "Backpropagation applied to handwritten zip code recognition" (LeCun, 1989)



뒤틀림, 굵김 등 다양한 변화가 있음



CNN 등장배경

Fully-connected layer의 구조적 문제로 인한 문제 발생

CNN 등장배경

Fully-connected layer의 구조적 문제로 인한 문제 발생

- 이미지는 (가로, 세로, 채널)의 형태를 가지는 3차원 배열
- FC Layer의 입력은 항상 1차원 배열
- FC Layer는 모든 값들이 완전 연결되어 있으므로 전체 픽셀의 모든 관계를 다 계산 해야함

CNN 등장배경

Fully-connected layer의 구조적 문제로 인한 문제 발생

- 이미지는 (가로, 세로, 채널)의 형태를 가지는 3차원 배열
- FC Layer의 입력은 항상 1차원 배열
- FC Layer는 모든 값들이 완전 연결되어 있으므로 전체 픽셀의 모든 관계를 다 계산 해야함

↓
이미지의 3차원 배열 형상을 무시하고 1차원 배열로 flatten해서 학습

CNN 등장배경

Fully-connected layer의 구조적 문제로 인한 문제 발생

- 이미지는 (가로, 세로, 채널)의 형태를 가지는 3차원 배열
- FC Layer의 입력은 항상 1차원 배열
- FC Layer는 모든 값들이 완전 연결되어 있으므로 전체 픽셀의 모든 관계를 다 계산 해야함

↓
이미지의 3차원 배열 형상을 무시하고 1차원 배열로 flatten해서 학습

이미지의 전체적인 관계를 고려하지 못해서 **변형된 데이터에 매우 취약함 (Topology)**

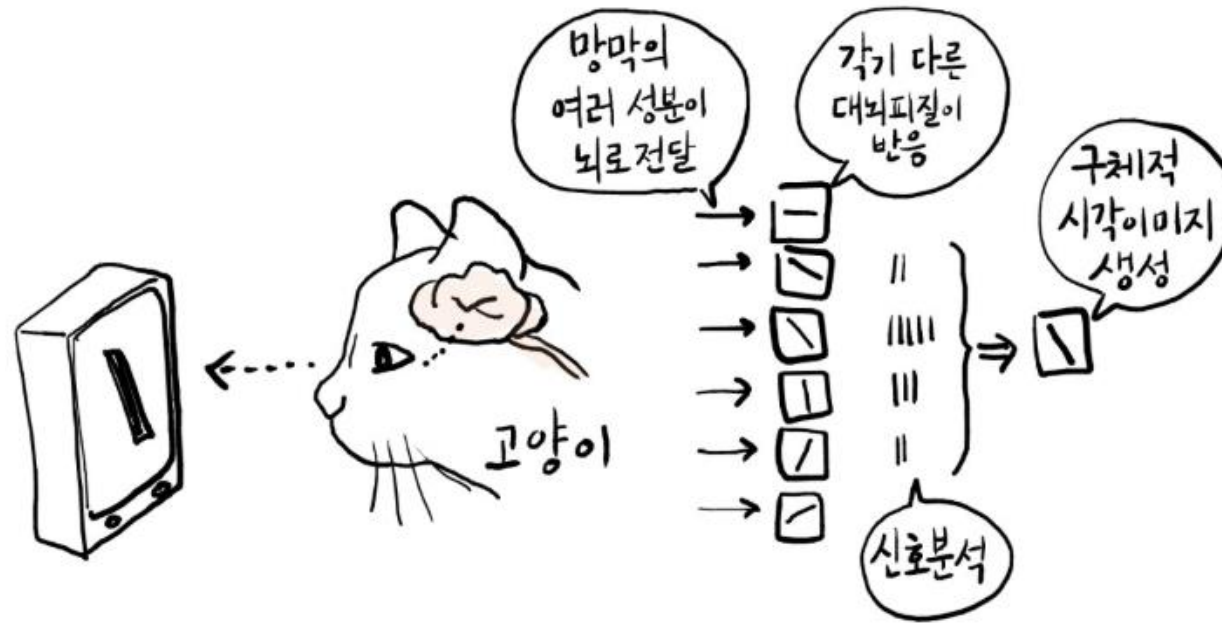
이미지의 특정 픽셀은 **주변 픽셀과 관련이 있다는 특성을 잃어버림 (Locality)**

=> 이미지를 조금만 변형해도 아예 다른 Object 로 인식하게 됨

모든 이미지마다 학습해야 해서 **망의 크기, 변수의 개수, 학습 시간 ↑**

CNN 등장배경

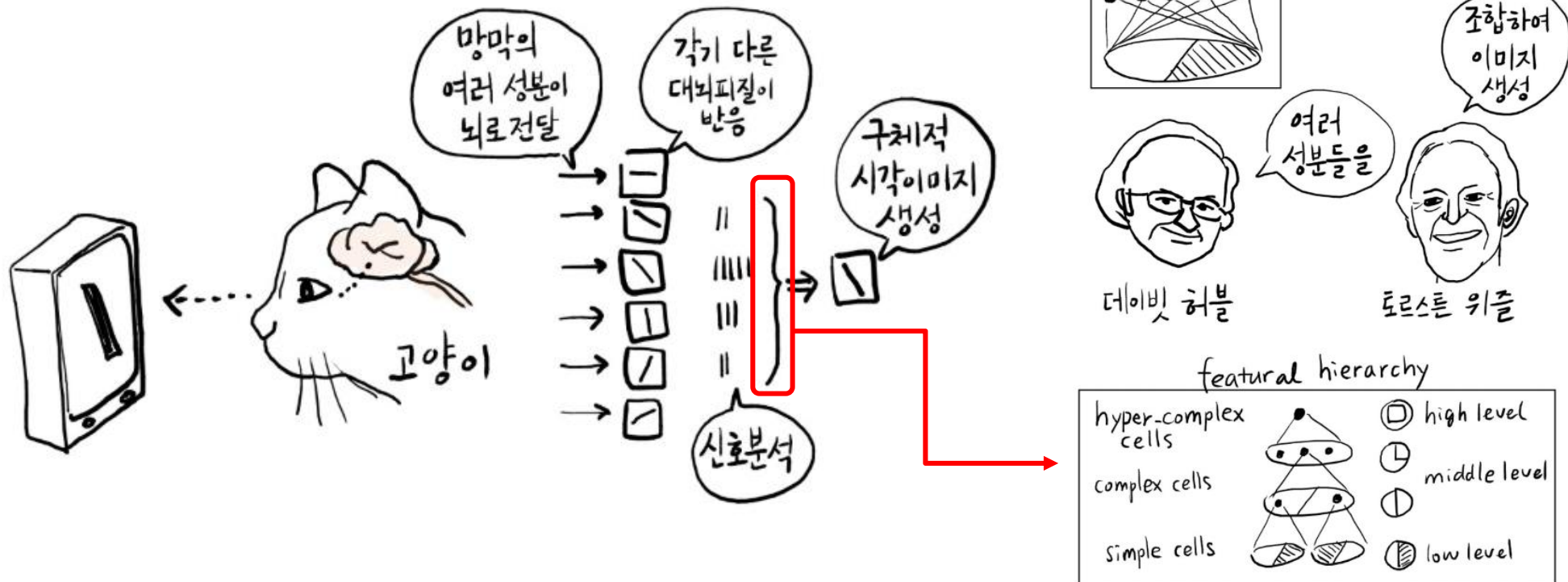
인간은 이미지를 어떻게 인식하지?



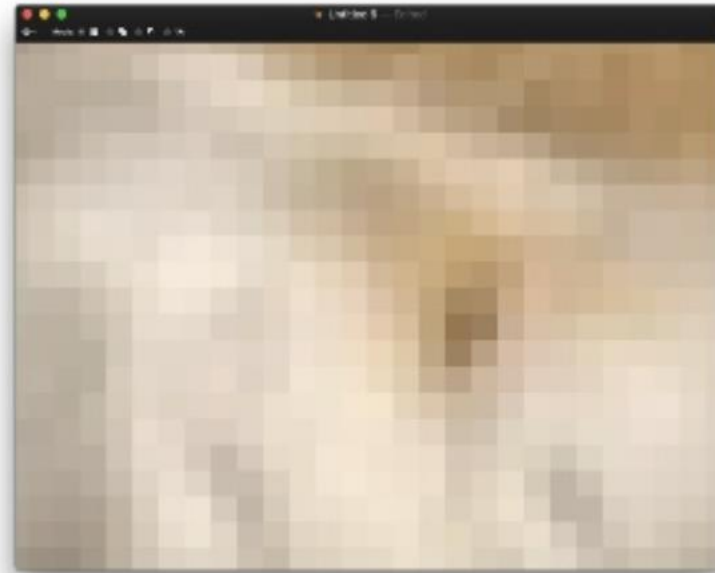
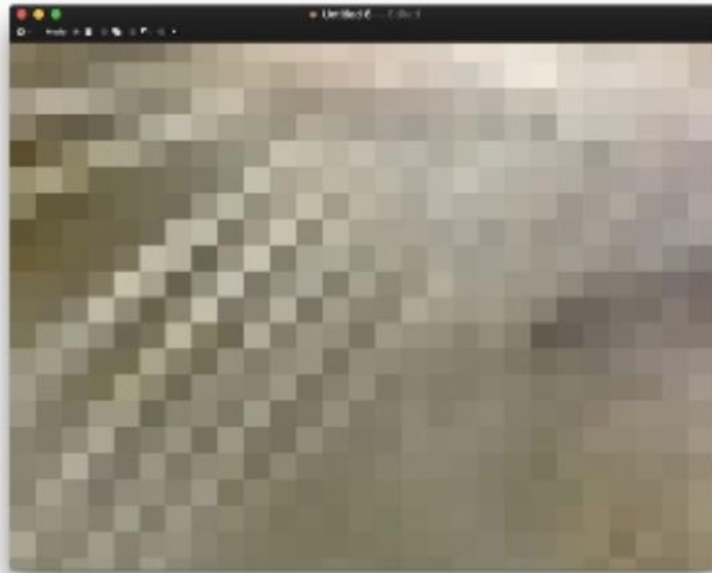
David H. Hubel & Torsten Wiesel (1958-59) 고양이 실험

CNN 등장배경

인간은 이미지를 어떻게 인식하지?

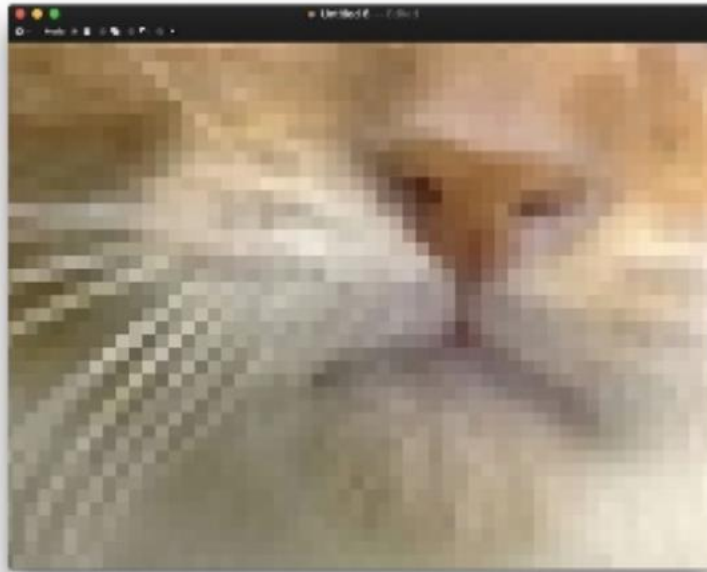


그림을 눈앞 1cm거리에서 본다고 생각해보자.



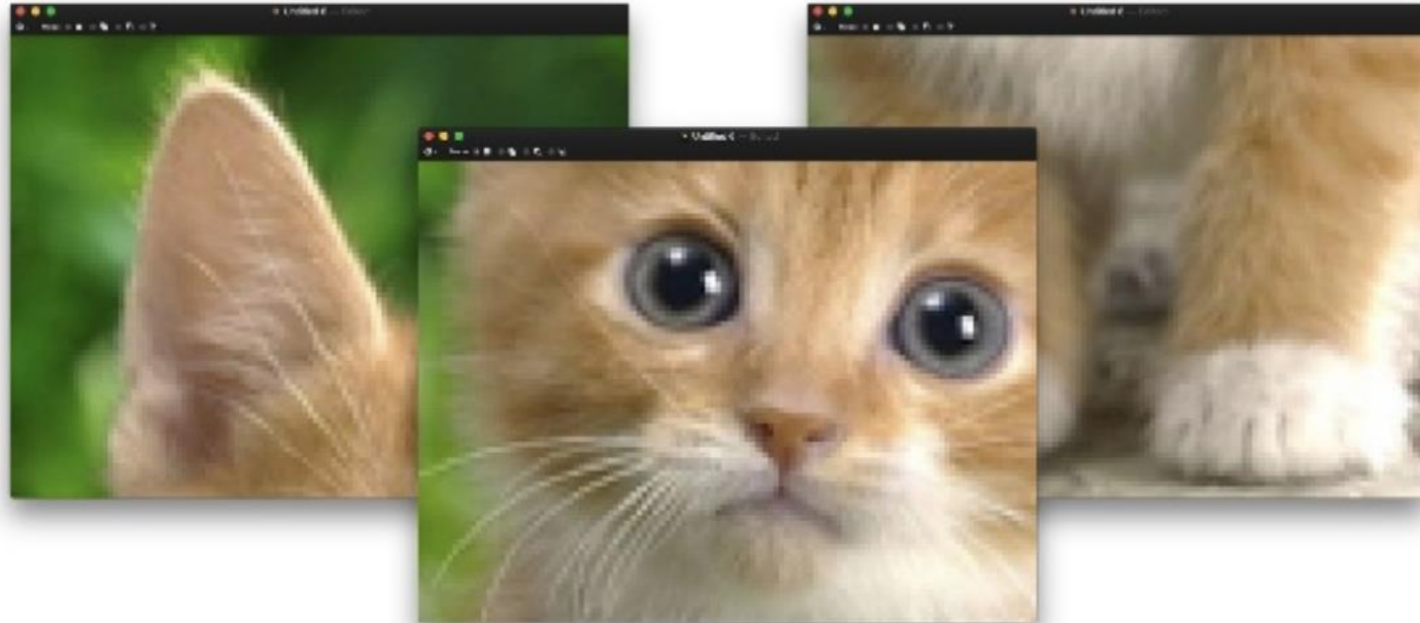
처음에는 점과 선, 이상한 질감 몇개 밖에 모르겠다.

점과 선, 질감을 충분히 배우고, 조금 떨어져서 보자.



점과 선이 질감이 합쳐져 삼각형, 동그라미, 북실함이 보인다.

삼각형, 원, 사각형, 복실함등을 조합해서 보니



뽀족귀와 땡그란눈과 복실한 발을 배웠다.

더 멀리서 보니, 그것들이 모아져있다. 이것은?

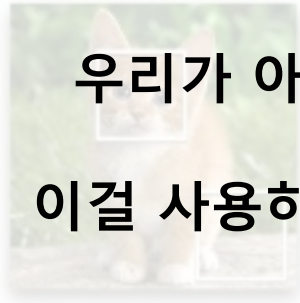


고양이!!



이미지에서는 인근 픽셀끼리만 상관있지 않나?

가까운 것들끼리만 묶어서 계산하면 의미도 있고
계산량도 줄겠는데?



우리가 아는 연산자중에 “convolution” 이 있는데
이걸 사용하면 컴퓨터가 비슷한 역할을 할 수 있겠다!

이미지에서는 인접 픽셀끼리만 상관있지 않나?
가까운 것들끼리만 묶어서 계산하면 의미도 있고
계산량도 줄겠는 거?




Convolution 이란 ?

Convolution 직관적 이해

원래 Convolution 이란?

= 필터 (Filter) 로 긁어내는 연산

= 이미지 각 픽셀에 대해 필터의 값들을 곱한 후 합치는 과정

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

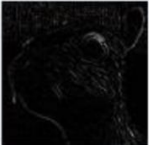


Convolution 직관적 이해

원래 Convolution 이란?

= 필터 (Filter) 로 긁어내는 연산

= 이미지 각 픽셀에 대해 필터의 값들을 곱한 후 합치는 과정

마스크(kernel)의 계수(weight)에 따라

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	



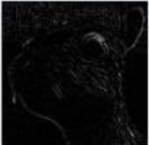



Convolution 직관적 이해

원래 Convolution 이란?

= 필터 (Filter) 로 긁어내는 연산

= 이미지 각 픽셀에 대해 필터의 값들을 곱한 후 합치는 과정

마스크(kernel)의 계수(weight)에 따라
필터와 유사한 이미지의 영역을 강조하는 결과 이미지를
얻어낼 수 있음

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Convolution 직관적 이해

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

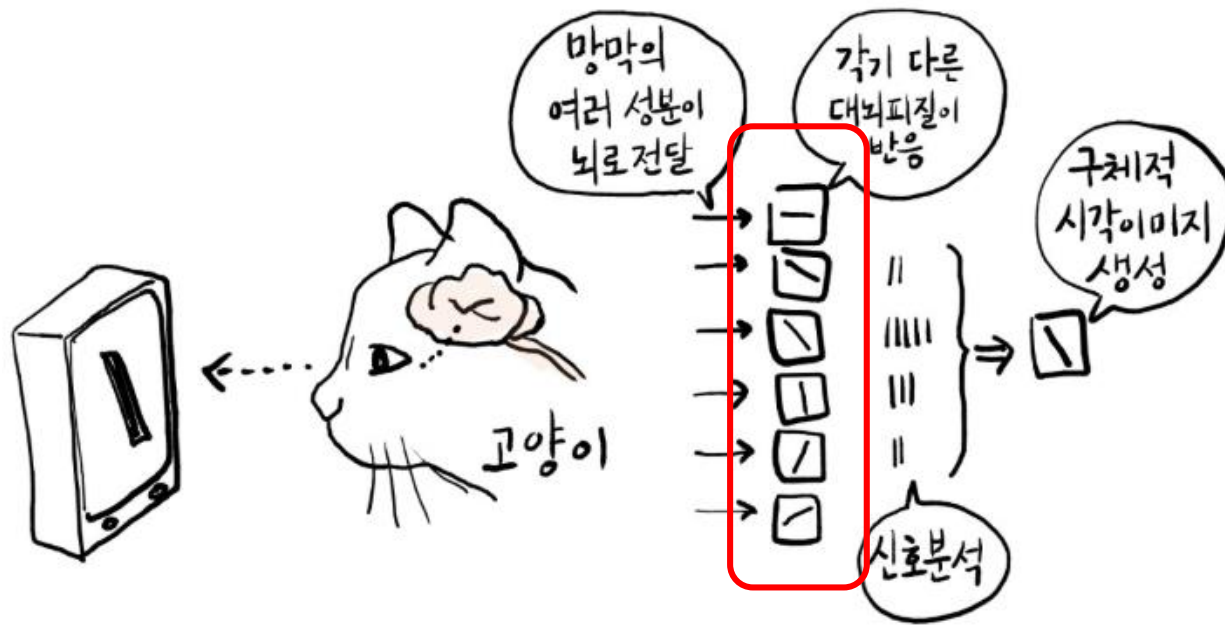
Image

4		

Convolved
Feature

Convolution 직관적 이해

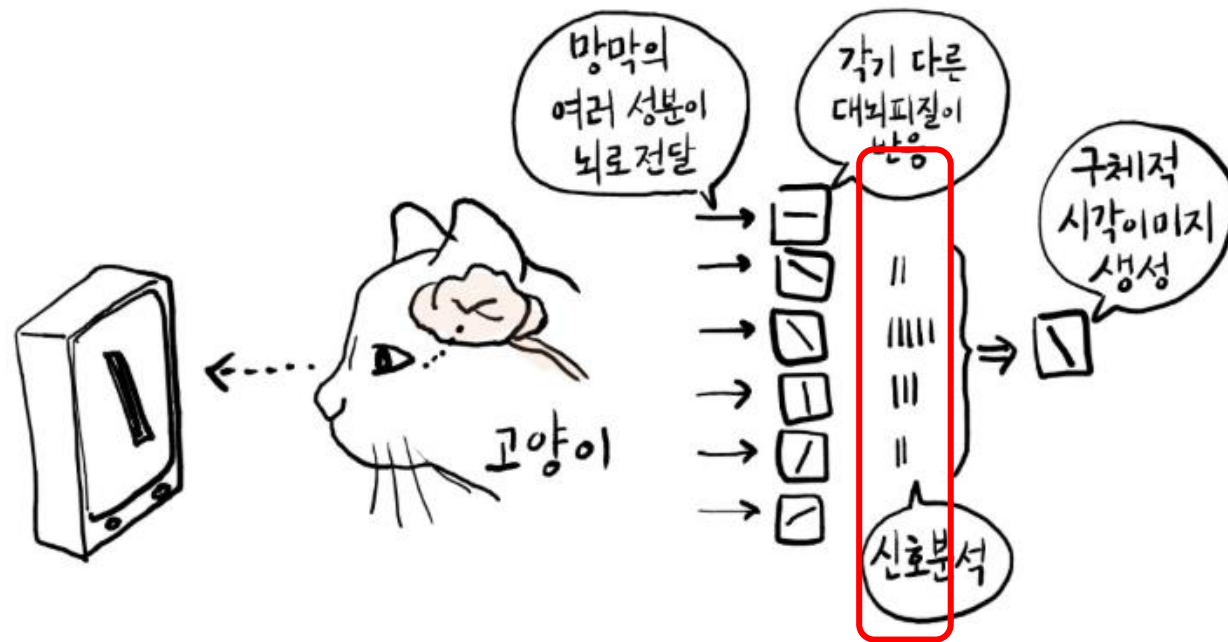
컴퓨터가 이미지를 어떻게 인식하게 하지?



Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Convolution 직관적 이해

컴퓨터가 이미지를 어떻게 인식하게 하지?



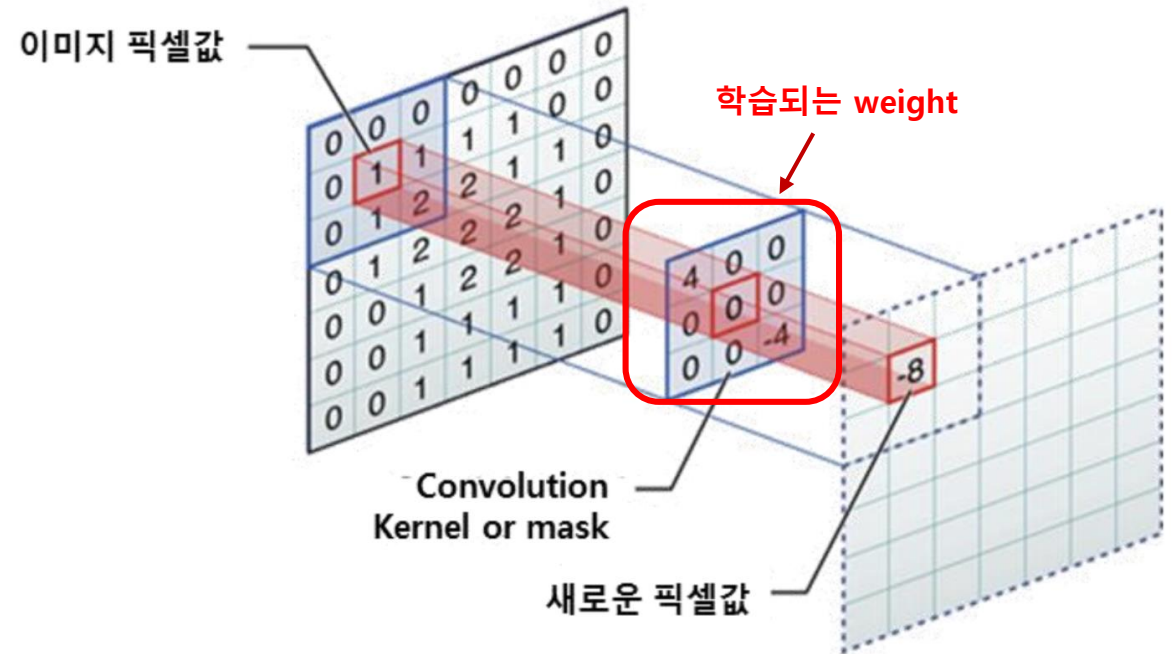
Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Convolution 직관적 이해

Convolution 과 CNN Convolution의 차이

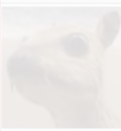
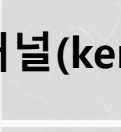
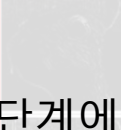
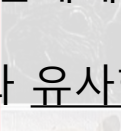
Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

사람이 설계



신경망이 설계해줌

Convolution 직관적 이해

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

필터(filter) or 커널(kernel) = Convolution layer 의 가중치 파라미터

학습단계에서 적절한 필터를 찾도록 학습하고

Convolution layer에서 필터와 유사한 이미지의 영역을 강조하는 **특성맵(feature map)**을 만듦

이미지 픽셀값

학습되는 weight

Convolution
Kernel or mask

새로운 픽셀값

사람이 설계

신경망이 설계해줌

Convolution 수식의 이해

필터로 곱는 것을 수식적으로는 어떻게 표현하는데?

Convolution 수식의 이해

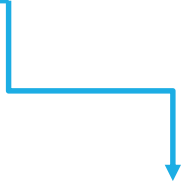
필터로 곱는 것을 수식적으로는 어떻게 표현하는데?

합성곱

위키백과, 우리 모두의 백과사전.

합성곱(合成-, convolution, 콘벌루션)은 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분하여 새로운 함수를 구하는 수학 연산자이다.

덧셈, 곱셈과 같은 '수학 연산자'


$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

CNN 에서 함수 f , g 는 이미지 자체, t 는 위치를 의미함

$f(r)g(r) \rightarrow$ 좌우 반전 $\rightarrow f(r)g(-r) \rightarrow t$ 만큼 이동 $\rightarrow f(r)g(-(r-t)) = f(r)g(t-r)$

Convolution 수식의 이해

필터로 곱는 것을 수식적으로는 어떻게 표현하는데?

Convolution

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

$$2D : (f * g)(i, j) = \sum_{x=0}^{h-1} \sum_{y=0}^{w-1} f(x, y)g(i - x, j - y)$$

좌/우 반전시킨 g 함수를 i, j 만큼 평행이동

h, w : 이미지의 높이, 너비

i, j : 필터의 높이, 너비

Convolution 수식의 이해

필터로 곱는 것을 수식적으로는 어떻게 표현하는데?

Convolution

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

$$2D : (f * g)(i, j) = \sum_{x=0}^{h-1} \sum_{y=0}^{w-1} f(x, y)g(i - x, j - y)$$

좌/우 반전시킨 g 함수를 i, j 만큼 평행이동

h, w : 이미지의 높이, 너비

i, j : 필터의 높이, 너비

Cross-correlation

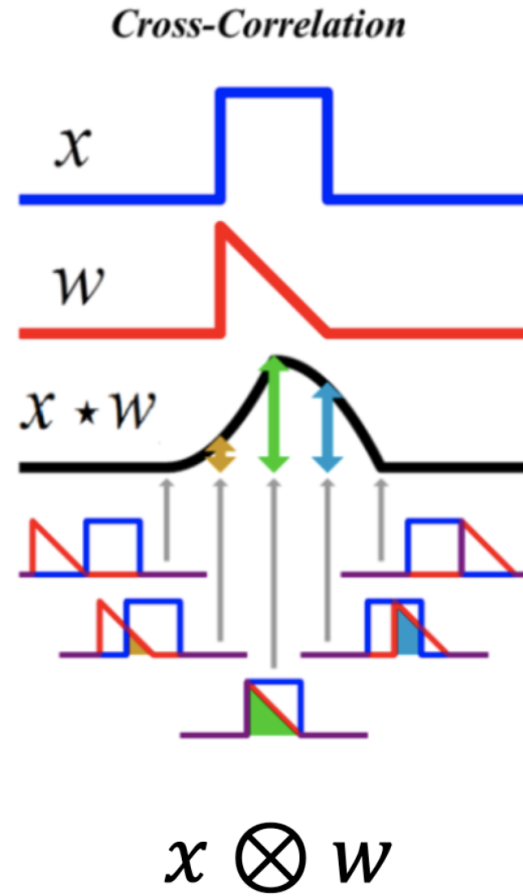
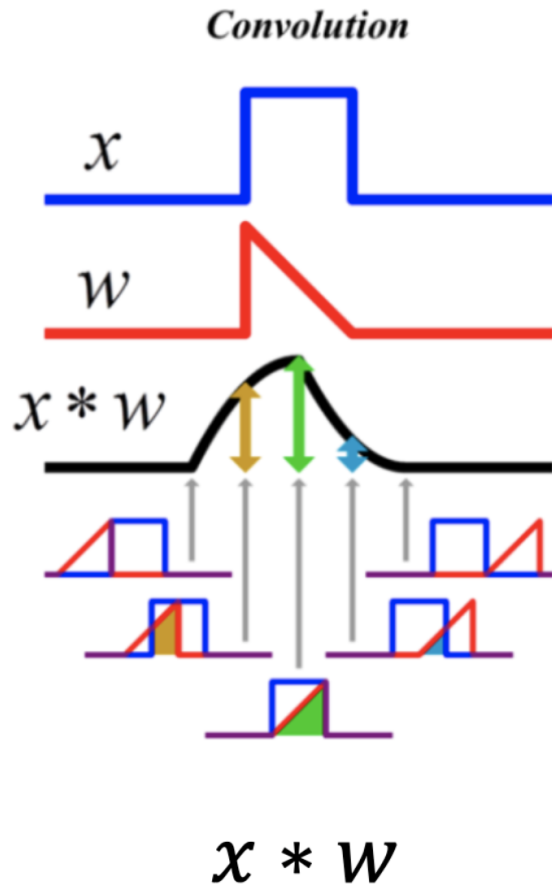
$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t + \tau)d\tau$$

$$(f * g)(i, j) = \sum_{x=0}^{h-1} \sum_{y=0}^{w-1} f(x, y)g(i + x, j + y)$$

함수 반전 X

부호만 다름

Convolution 수식의 이해



Convolution 수식의 이해

Convolution

image		
1	2	3
4	5	6
7	8	9

*

kernel		
A	B	C
D	E	F
G	H	I

$$(1 * I) + (2 * H) + (3 * G) + (4 * F) + \dots + (9 * A)$$

Cross-correlation

image		
1	2	3
4	5	6
7	8	9

\otimes

kernel		
A	B	C
D	E	F
G	H	I

$$(1 * A) + (2 * B) + (3 * C) + (4 * D) + \dots + (9 * I)$$

Convolution 수식의 이해

Convolution

image			kernel		
1	2	3	A	B	C
4	5	6	D	E	F
7	8	9	G	H	I

*

$$(1 * I) + (2 * H) + (3 * G) + (4 * F) + \dots + (9 * A)$$

Cross-correlation

image			kernel		
1	2	3	A	B	C
4	5	6	D	E	F
7	8	9	G	H	I

⊗

실제 Convolution 구현은 Cross-correlation 으로 함

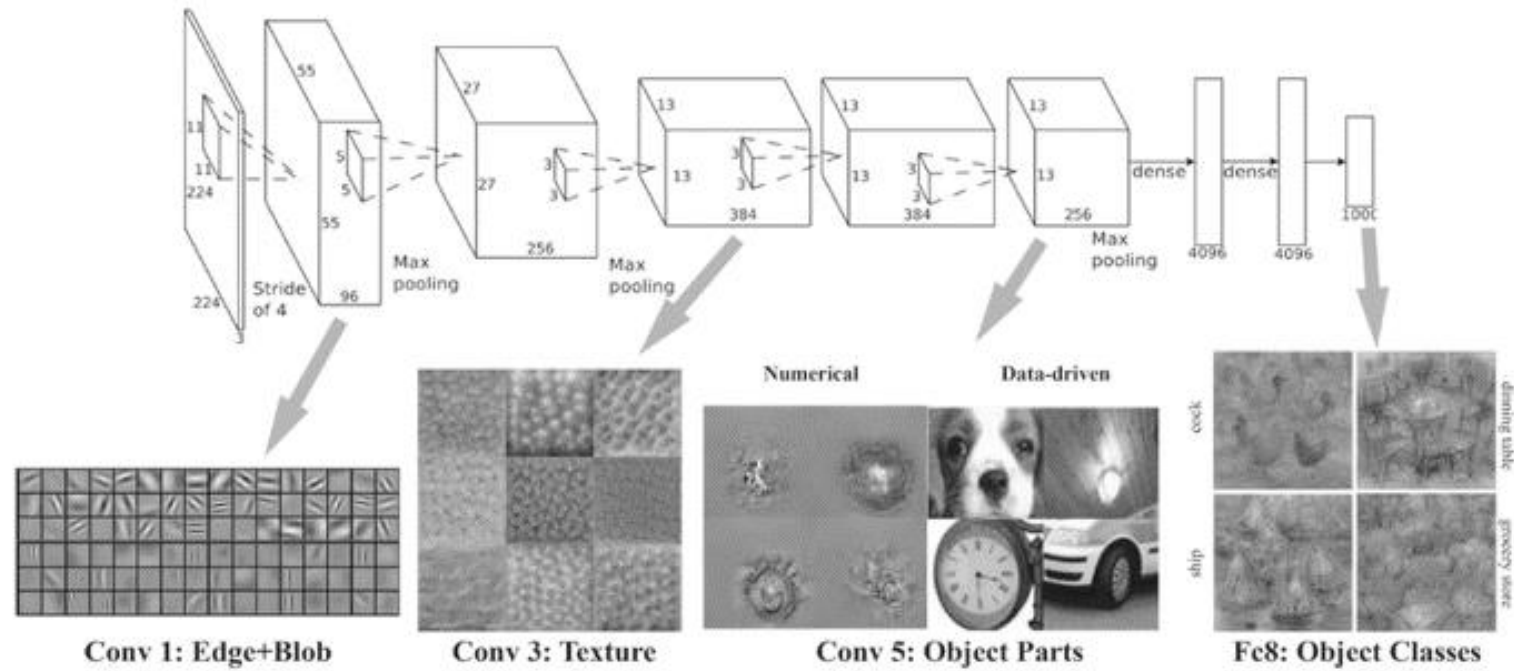
$$(1 * A) + (2 * B) + (3 * C) + (4 * D) + \dots + (9 * I)$$

CNN은 어떤 구조인가?

CNN 구조의 이해

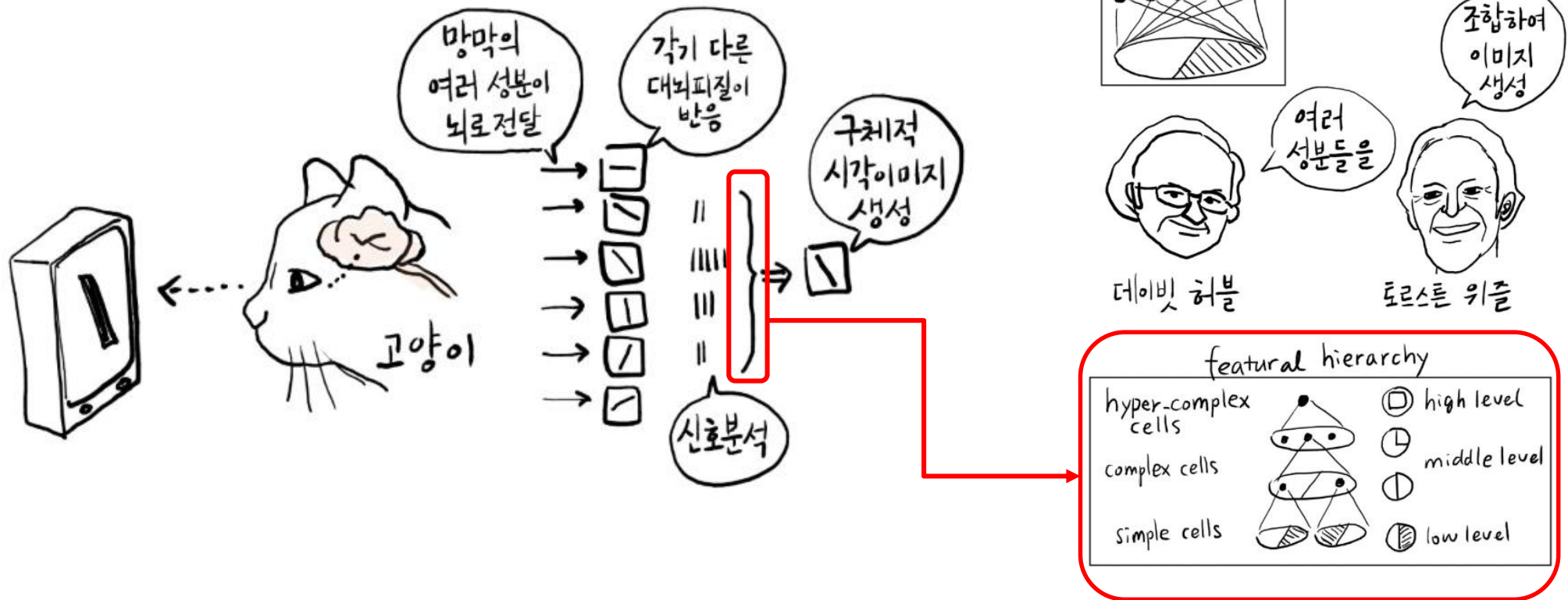
Convolution 을 사용한

Neural Network 구조는 어떻게 생겼는가?



CNN 구조의 이해

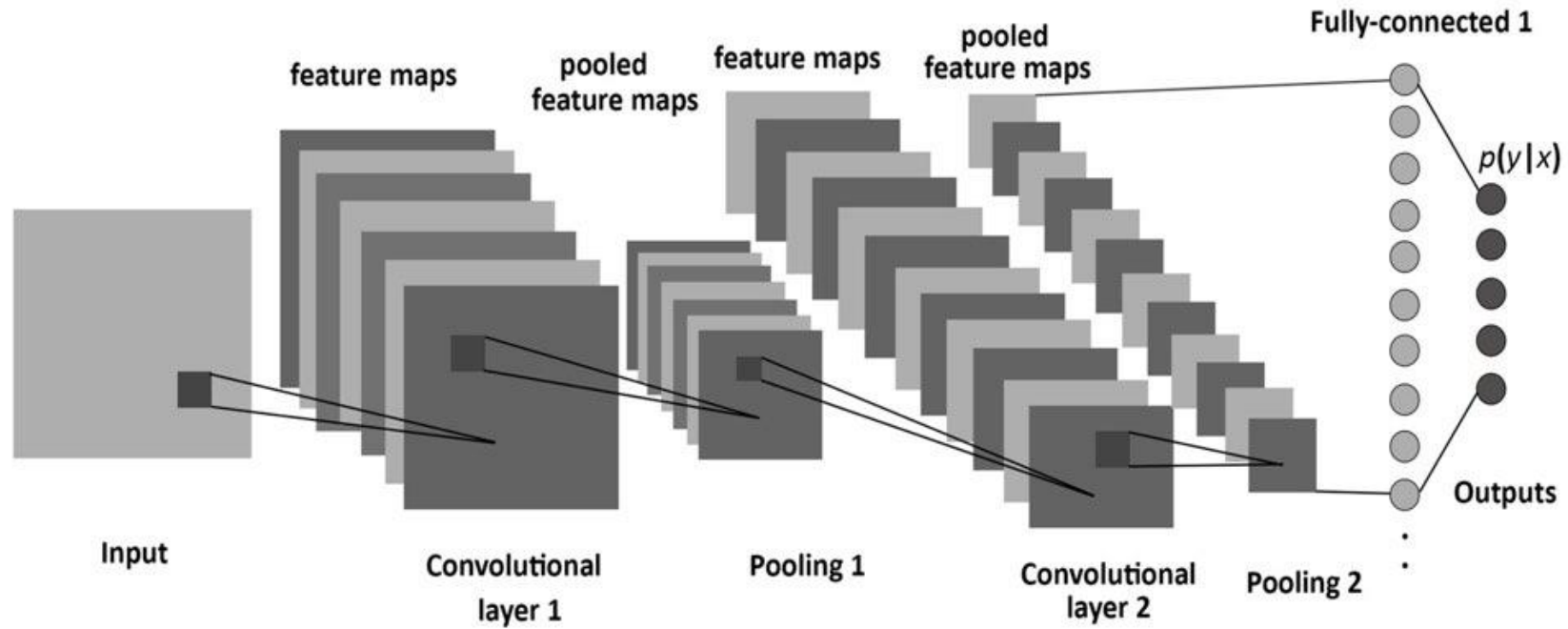
인간은 이미지를 어떻게 인식하지?



이부분을 어떻게 모델링할까?

CNN 구조의 이해

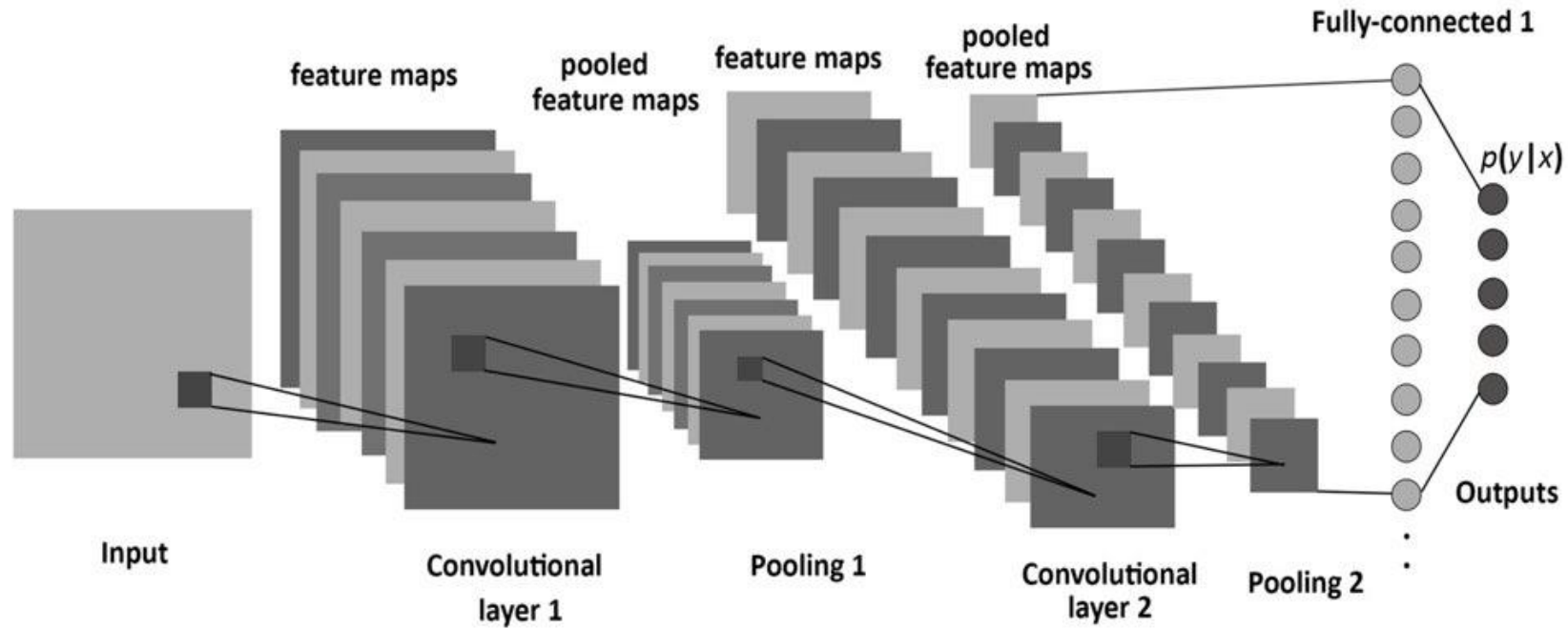
CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)



CNN 구조의 이해

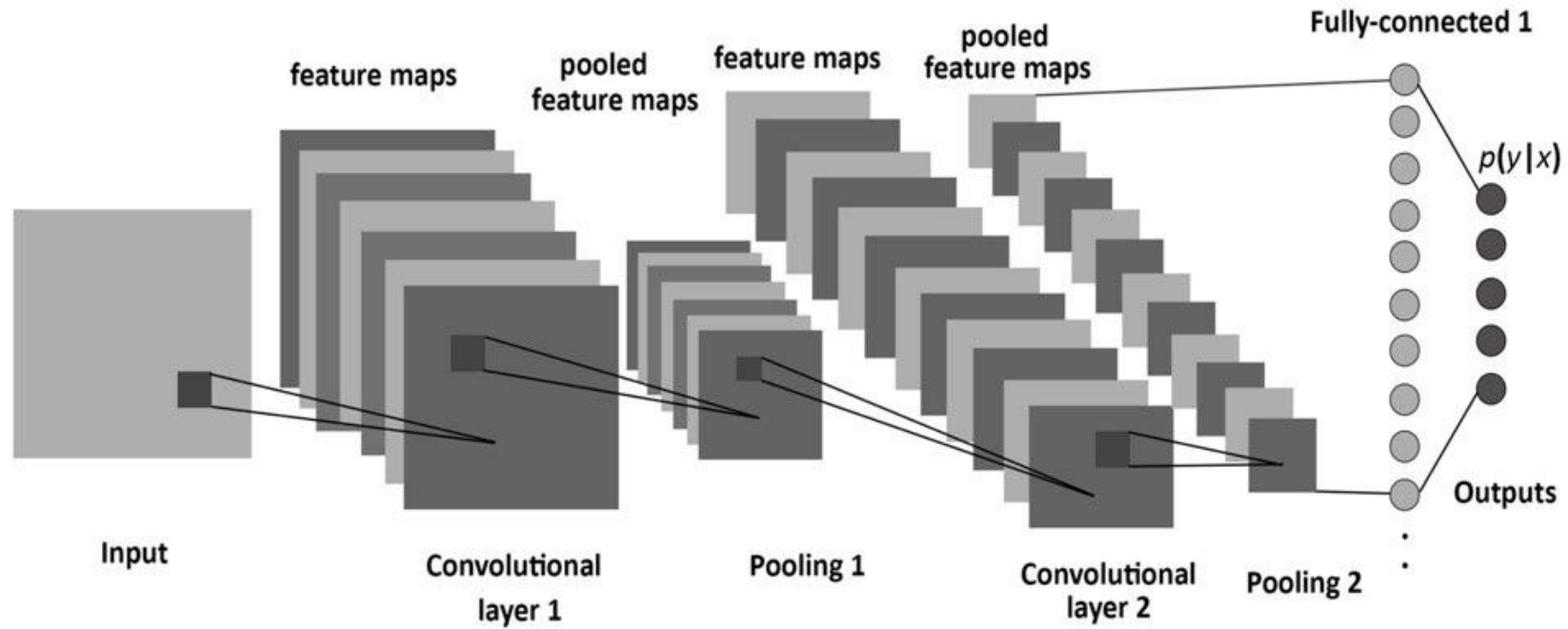
CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)

Feature map 추출



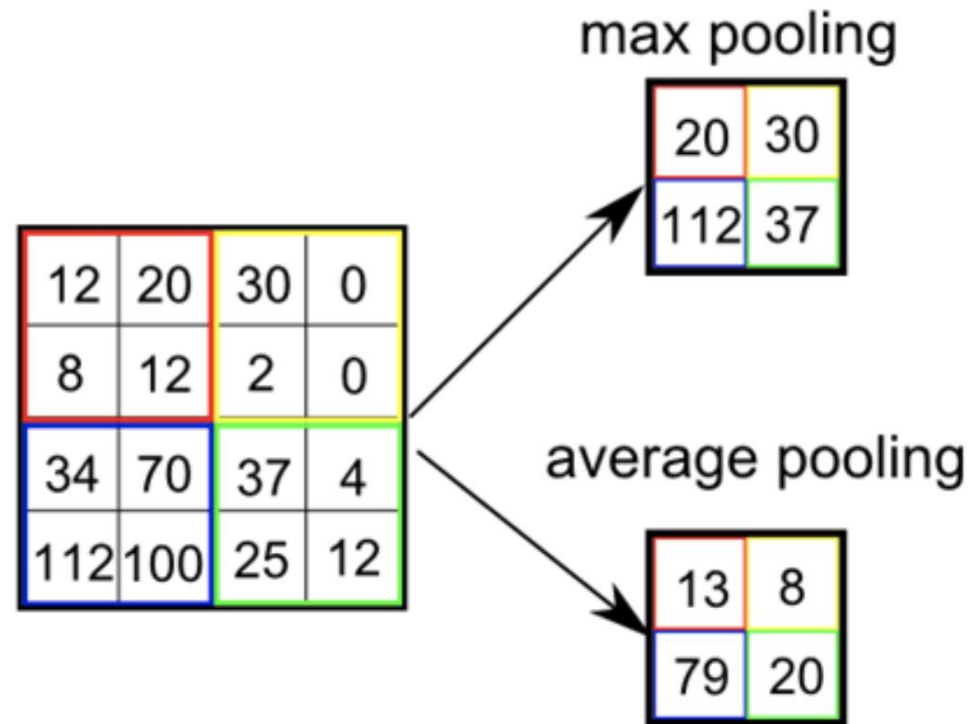
CNN 구조의 이해

CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)



CNN 구조의 이해

CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)

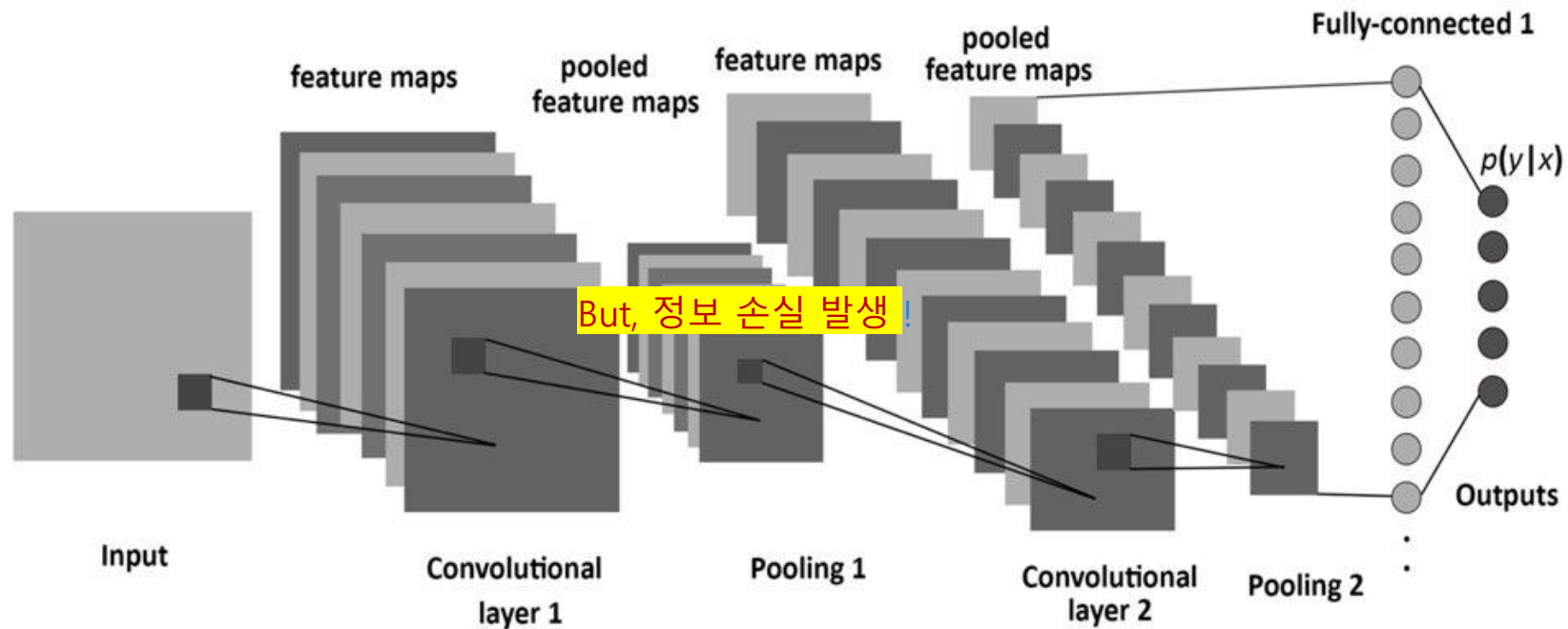


CNN 구조의 이해

CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)

Feature map 사이즈 줄여서 연산량 줄이기

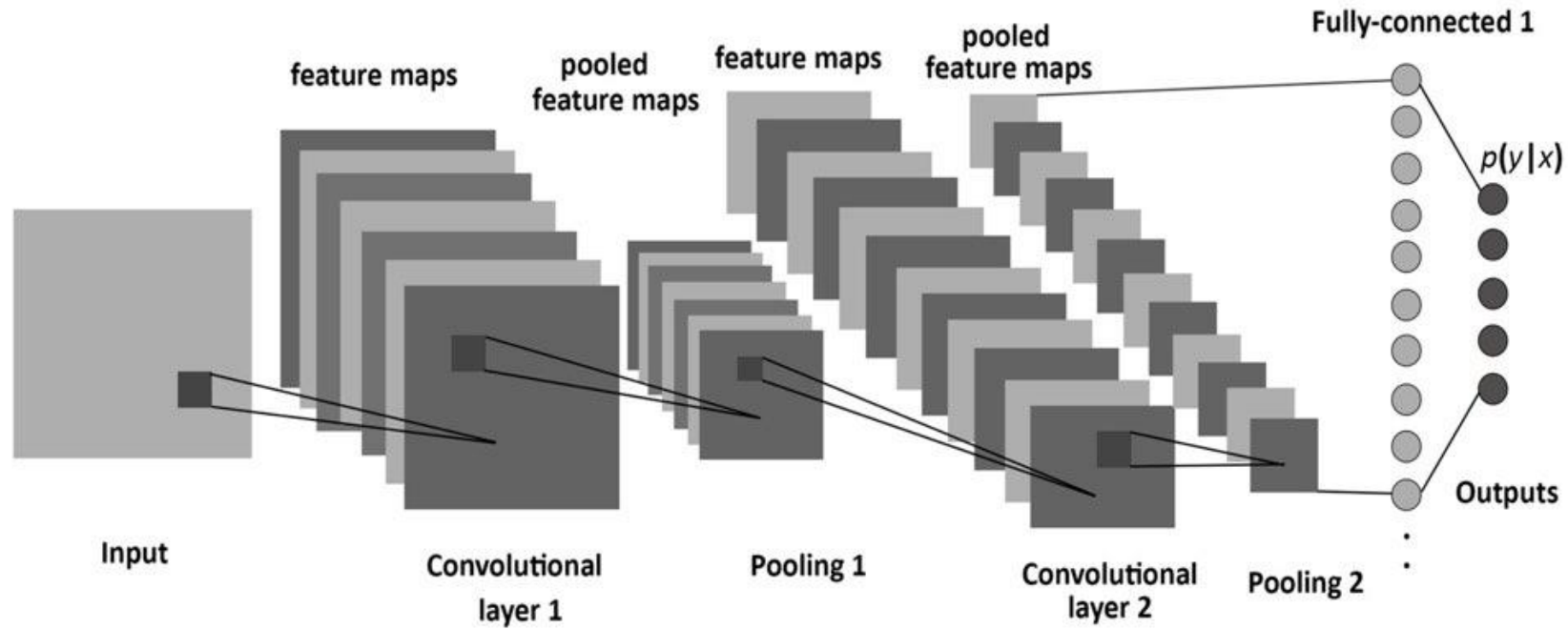
Strong, Global feature 위주로 뽑도록 학습하여 입력에 대한 Invariance 얻기



CNN 구조의 이해

CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)

사람? 운전대? Classification

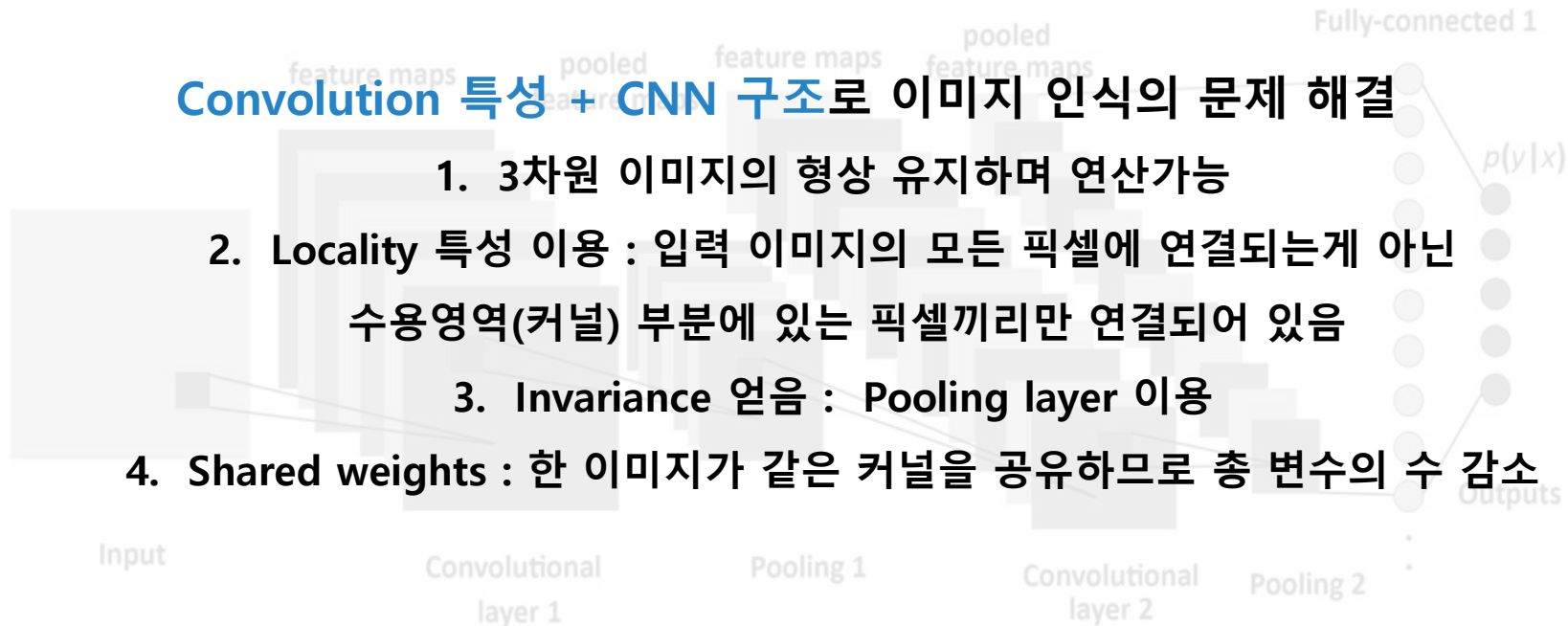


CNN 구조의 이해

CNN 구성요소 = (1) Convolution layer (2) Sub-sampling(pooling) layer (3) FC layer(Affine)

Convolution 특성 + CNN 구조로 이미지 인식의 문제 해결

1. 3차원 이미지의 형상 유지하며 연산가능
2. Locality 특성 이용 : 입력 이미지의 모든 픽셀에 연결되는게 아닌 수용영역(커널) 부분에 있는 픽셀끼리만 연결되어 있음
3. Invariance 얻음 : Pooling layer 이용
4. Shared weights : 한 이미지가 같은 커널을 공유하므로 총 변수의 수 감소



CNN 관련용어

관련 용어 : Channel

이미지에서의 채널

RED Channel



Green Channel



Blue Channel

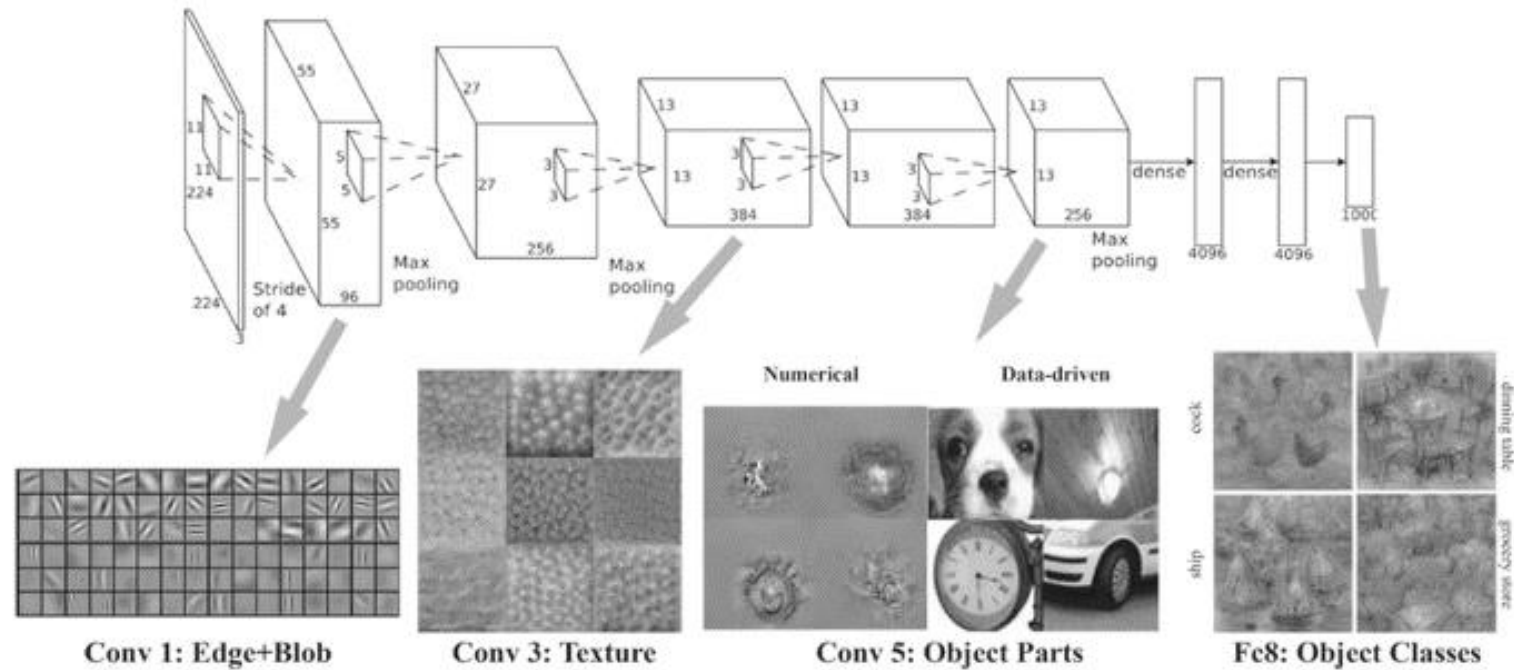


이미지는 3차원 배열 (가로, 세로, **채널**)

관련 용어 : Channel

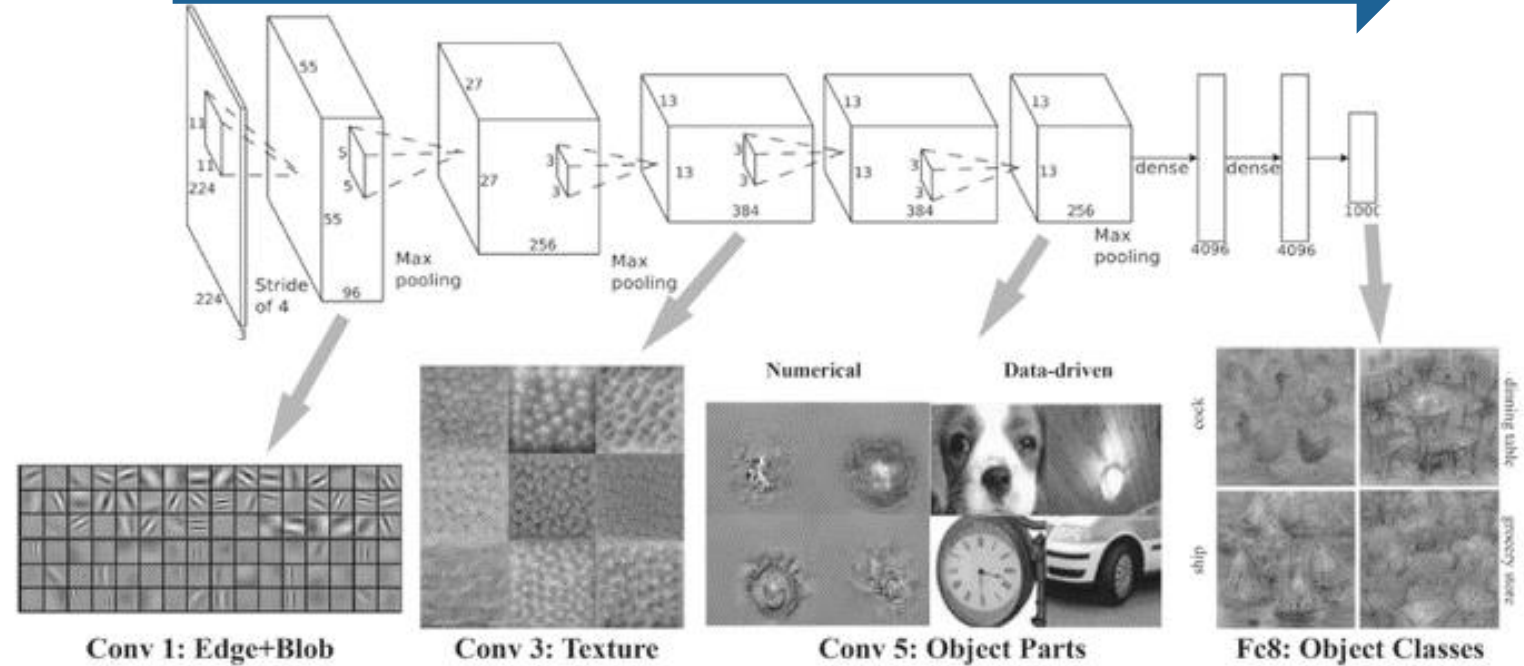
필터에서의 채널?

- 필터의 채널을 12개 쓴다 = 서로 다른 12개의 특징을 학습하겠다는 의미
- CNN은 모델의 뒷부분으로 갈수록 필터 채널의 개수를 늘리는 이유는?



연산량 줄어듦

Feature map의 가로, 세로 크기가 줄어듦



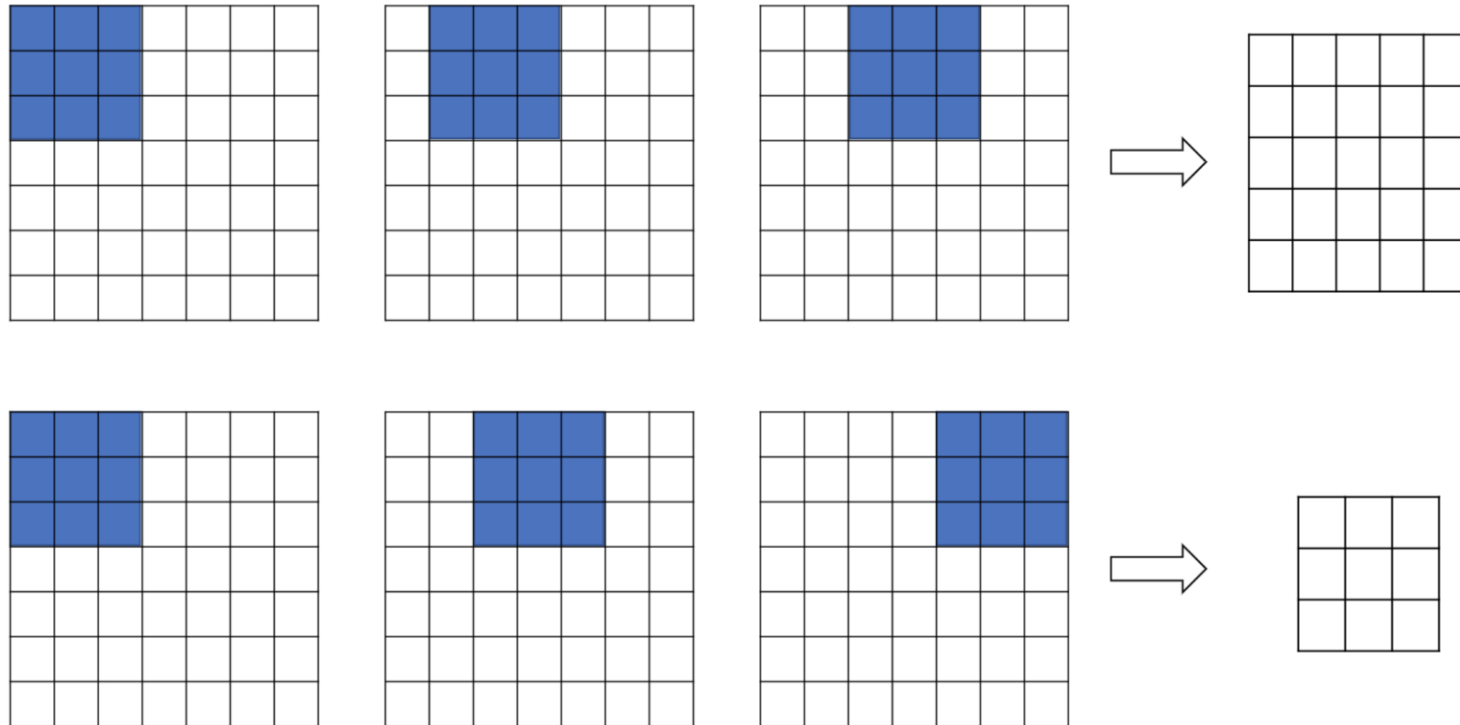
고차원 feature 표현

추상적이고, 복잡한 Feature map를 얻음

관련 용어 : Stride

Stride : Convolution 수행 시, 필터의 이동 간격

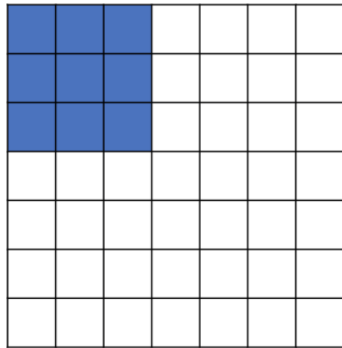
 : image  : filter



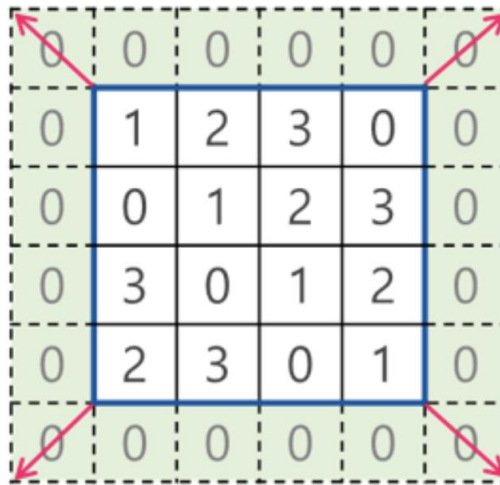
관련 용어 : Padding

Padding : 입력데이터의 주변을 특정 값으로 채워 늘리는 것

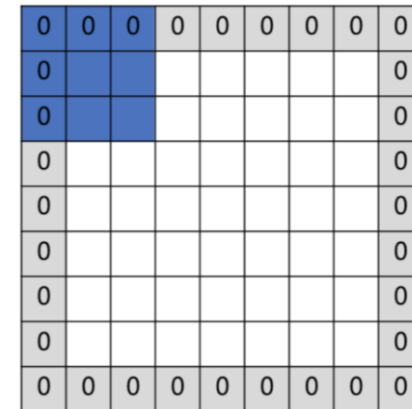
(일반적으로 0 으로 채운 Zero Padding 사용)



$7 \times 7 \xrightarrow{\text{conv}} 5 \times 5$



1픽사리 zero- padding



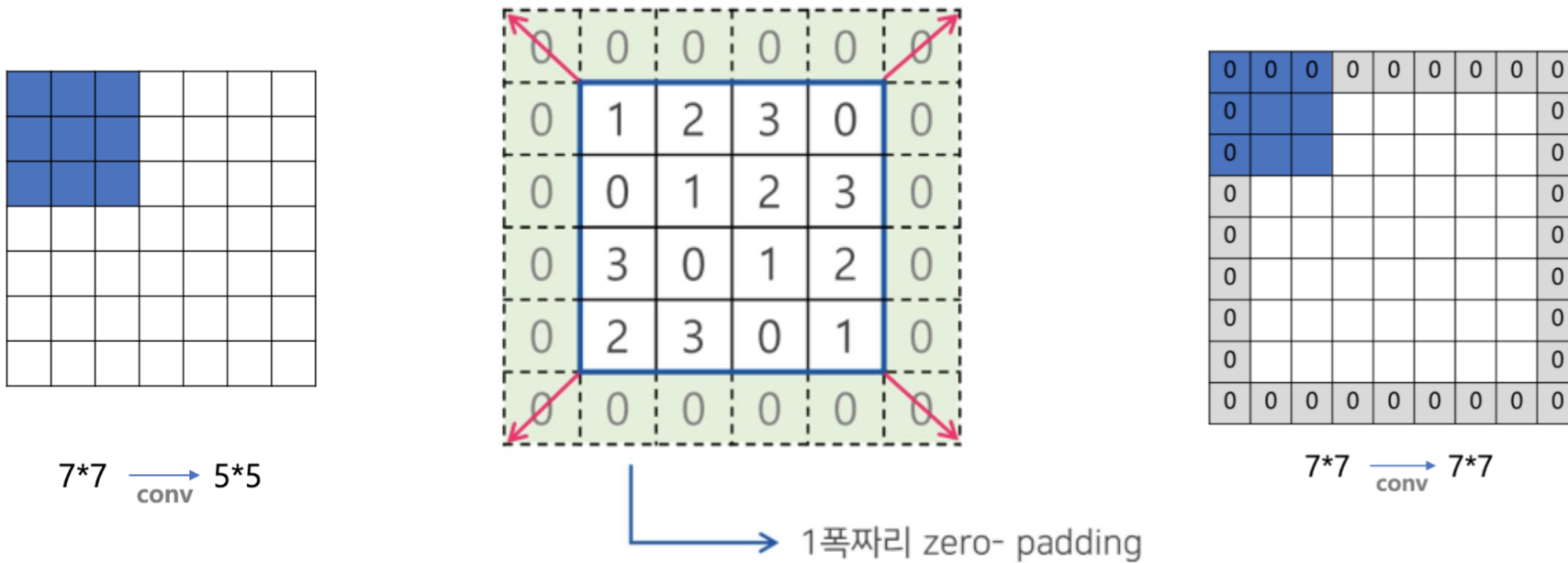
$7 \times 7 \xrightarrow{\text{conv}} 7 \times 7$

관련 용어 : Padding

Padding : 입력데이터의 주변을 특정 값으로 채워 늘리는 것

(일반적으로 0 으로 채운 Zero Padding 사용)

- 입력데이터 모서리 부분의 정보 손실 방지
- 출력데이터(feature map)의 크기 조절 (주로 입력과 출력 크기를 동일하게 해주기 위해 1 사용)



CNN이 사용되는 대표적인 Task

- Image Classification
- Semantic Segmentation
- Object Detection
- Object Localization
- Visual QnA
- Image Captioning

Q&A
