

Convolutional Neural Network

Week 2

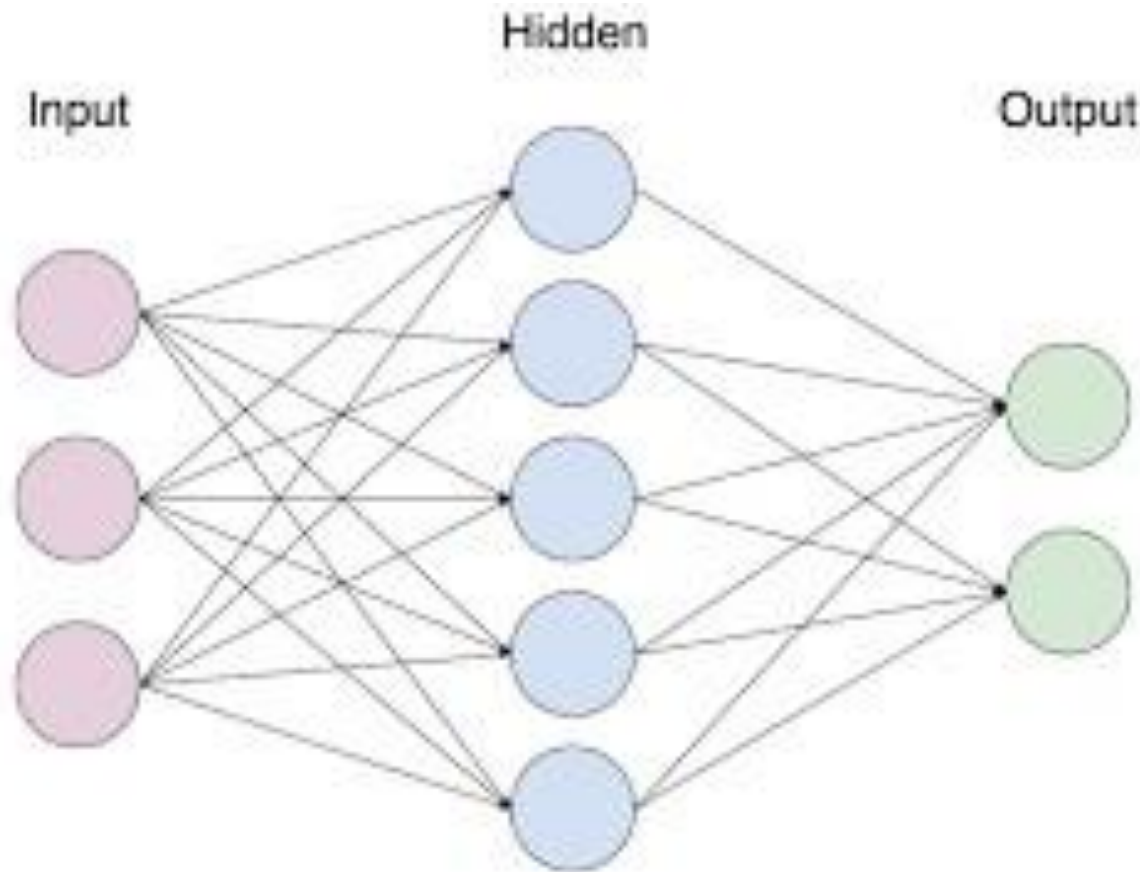
발표자 이원용

INDEX

- 앞강의에서 가져올 개념
 - NN
 - Activation Functions
- CNN
 - CNN LAYER
 - FILTER
 - STRIDE
 - PADDING
 - POOLING
 - MAXPOOLING
 - FC
 - GOOD CNN

Neural Network [신경망]

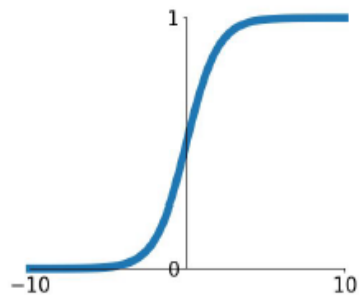
지난 시간에 Neural Network의 존재를 배웠습니다.



Activation Function [활성함수]

Sigmoid

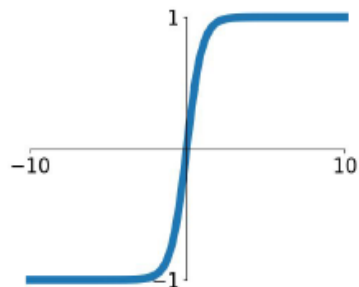
$$\sigma(x) = \frac{1}{1+e^{-x}}$$



$0 < y < 1$ 값을 반환
확률값의 의미
Classification 에서 자주 사용

tanh

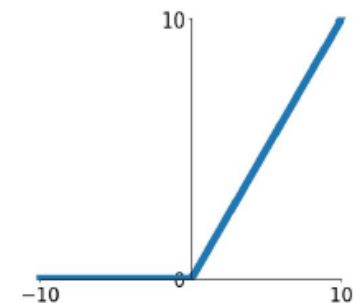
$$\tanh(x)$$



$-1 < y < 1$ 값을 반환

ReLU

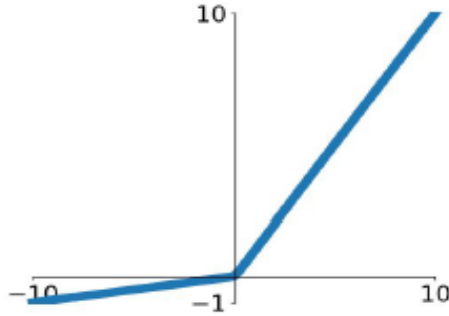
$$\max(0, x)$$



$y \geq 0$ 값을 반환
음의 값을 가지는 값은 모두 0으로 만들고
양의 값을 가지는 값은 그대로 적용
특징 추출에 유용하다고 한다.
간편하게 작동하고, 계산이 빠르다

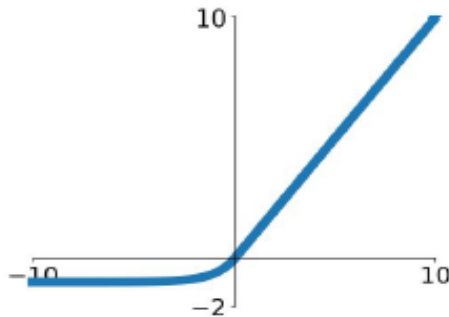
Activation Function [활성함수]

Leaky ReLU
 $\max(0.1x, x)$



Maxout
 $\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



CNN에 들어가기 앞서

Hierarchical Organization [계층/위계적 조직]

단순한 인식에서 복잡한 인식으로 가는 과정

CNN의 구조가 단순한 feature 추출에서 복잡한 feature 추출로

가는 것이라 생각하면 좋을것 같습니다.

CNN are Everywhere

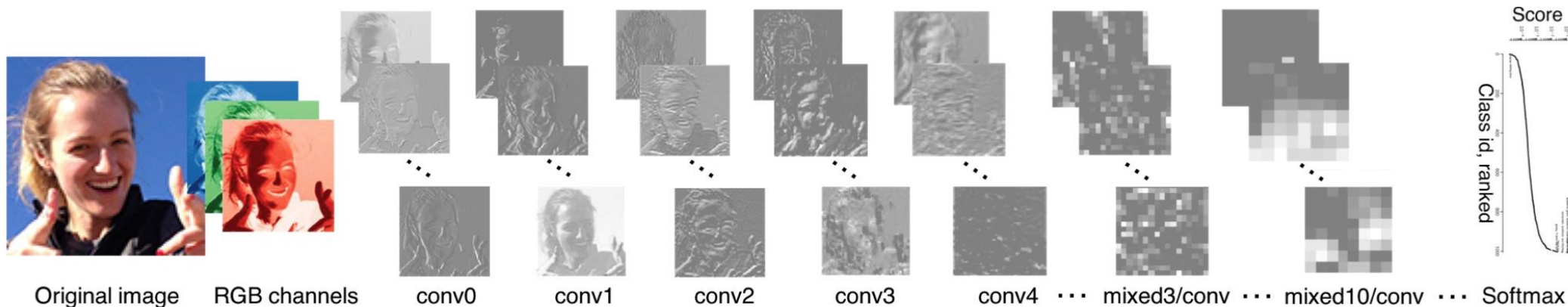
Classification [분류]



Retrieval [검색]

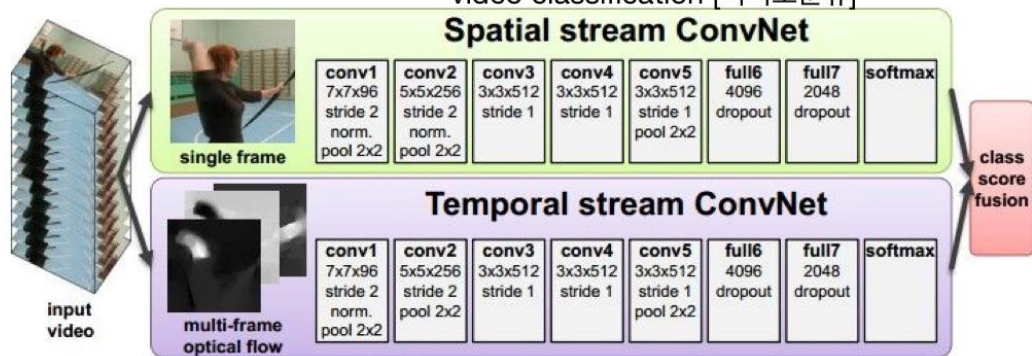


Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.



[Taigman et al. 2014] face recognition[얼굴인식] /
video classification [비디오분류]

Activations of [inception-v3 architecture](#) [Szegedy et al. 2015] to image of Emma McIntosh, used with permission. Figure and architecture not from Taigman et al. 2014.



[Simonyan et al. 2014]

Figures copyright Simonyan et al., 2014.
Reproduced with permission.

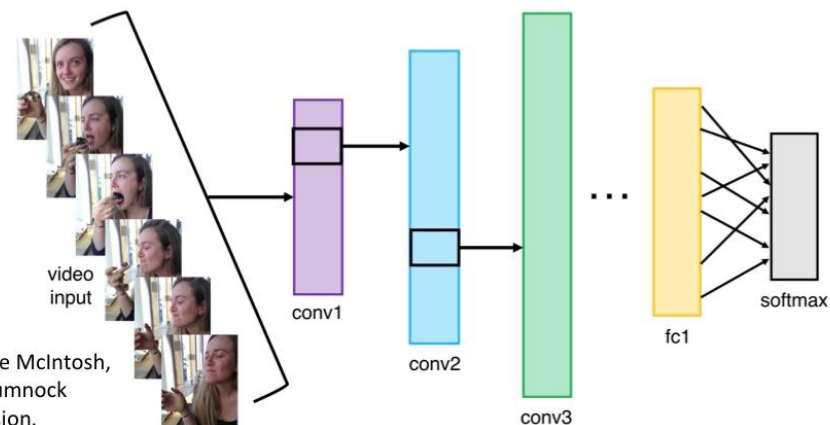
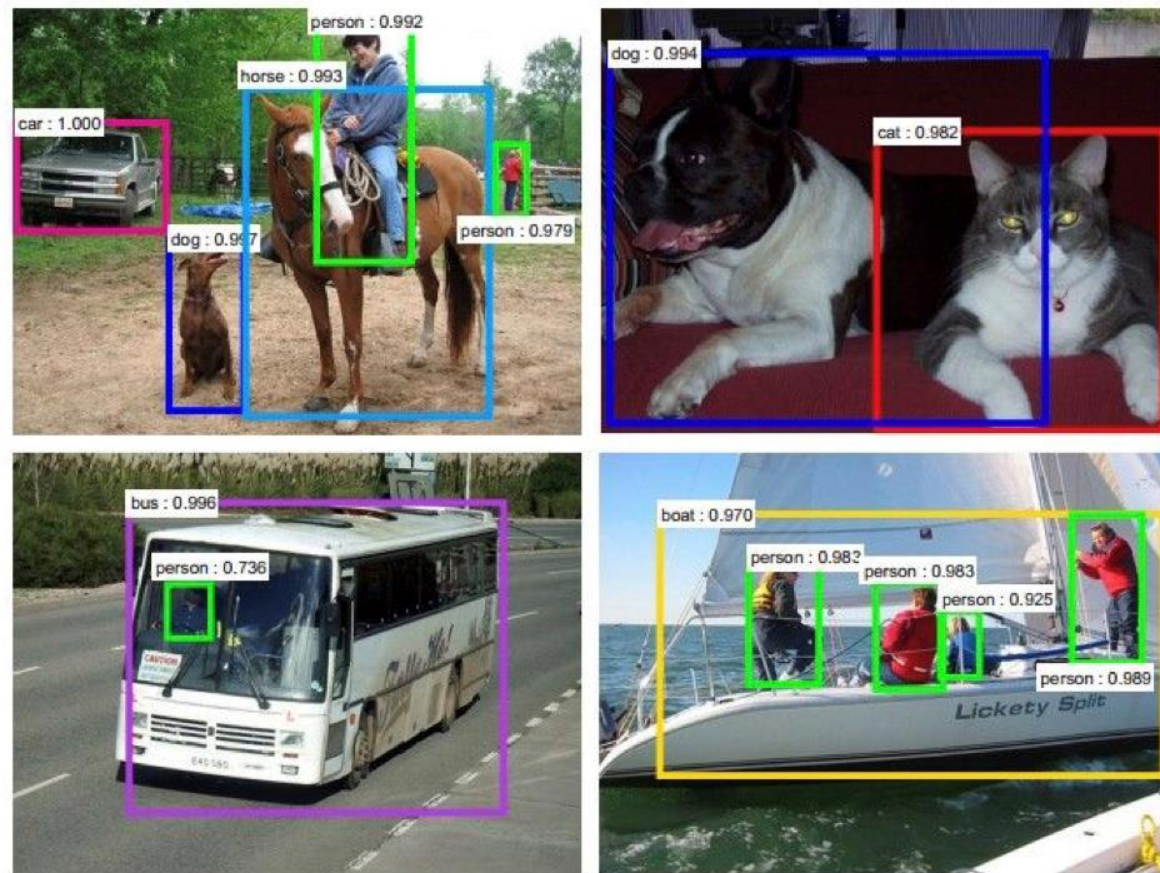


Illustration by Lane McIntosh,
photos of Katie Cumnock
used with permission.

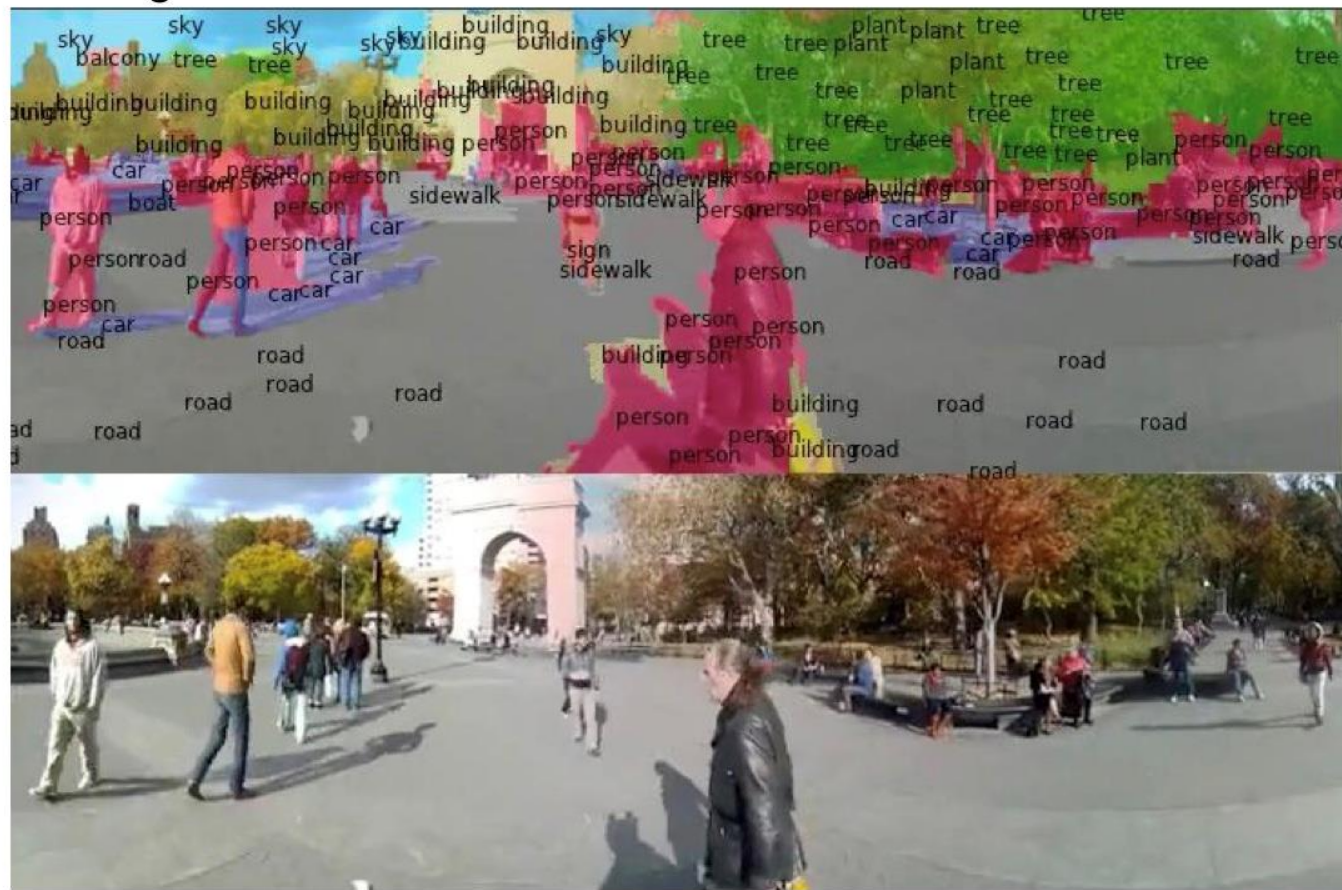
Detection 탐지



Figures copyright Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation 분할



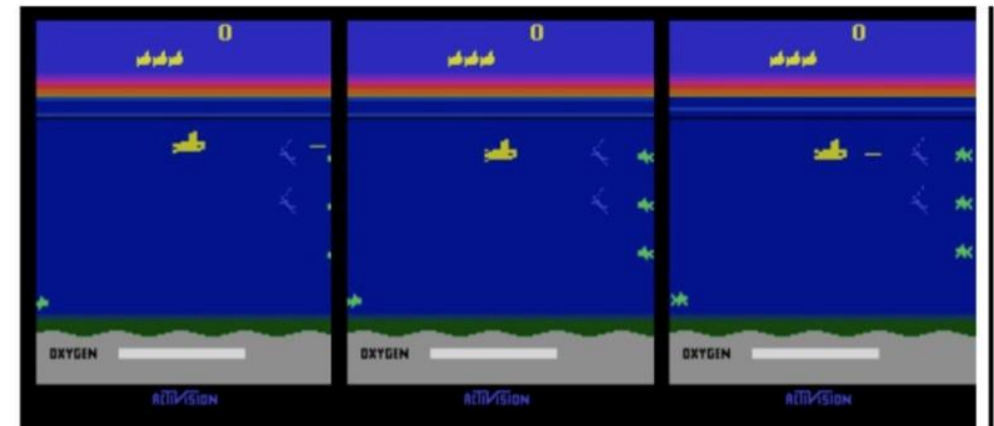
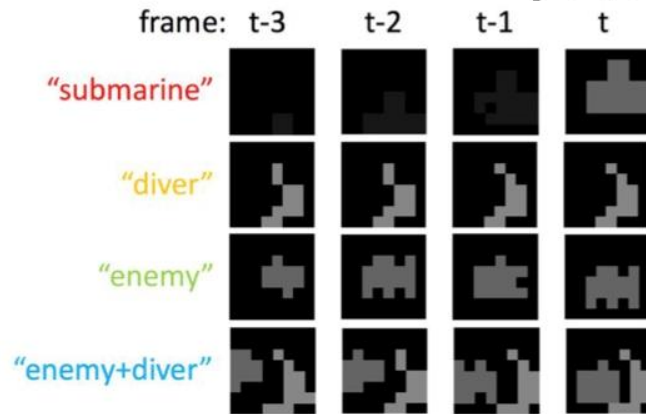
Figures copyright Clement Farabet, 2012. Reproduced with permission.

[Farabet et al., 2012]



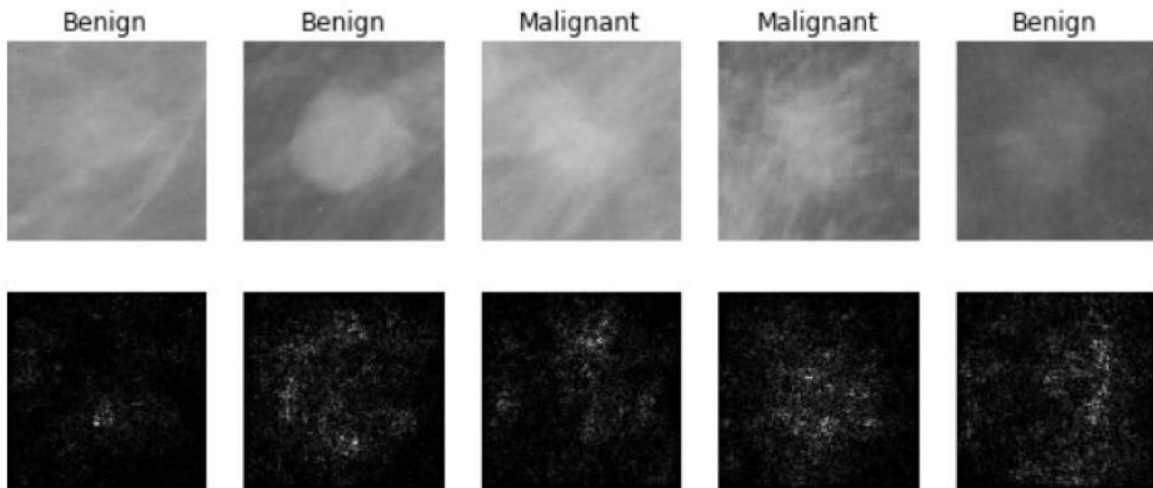
Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014] Pose recognition
[포즈인식]



[Guo et al. 2014] Game with deep

Figures copyright Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard Lewis, and Xiaoshi Wang, 2014. Reproduced with permission.



[Levy et al. 2016]

Figure copyright Levy et al. 2016.
Reproduced with permission.



[Dieleman et al. 2014]

From left to right: [public domain by NASA](#), usage [permitted](#) by ESA/Hubble, [public domain by NASA](#), and [public domain](#).

참고 : CNN을 이용한 피부암 진단 모델 /
<https://github.com/udacity/dermatologist-ai>



[Sermanet et al. 2011]
[Ciresan et al.]

Photos by Lane McIntosh.
Copyright CS231n 2017.

No errors



A white teddy bear sitting in the grass

Minor errors



A man in a baseball uniform throwing a ball

Somewhat related



A woman is holding a cat in her hand

Image Captioning

*[Vinyals et al., 2015]
[Karpathy and Fei-Fei, 2015]*



A man riding a wave on top of a surfboard



A cat sitting on a suitcase on the floor



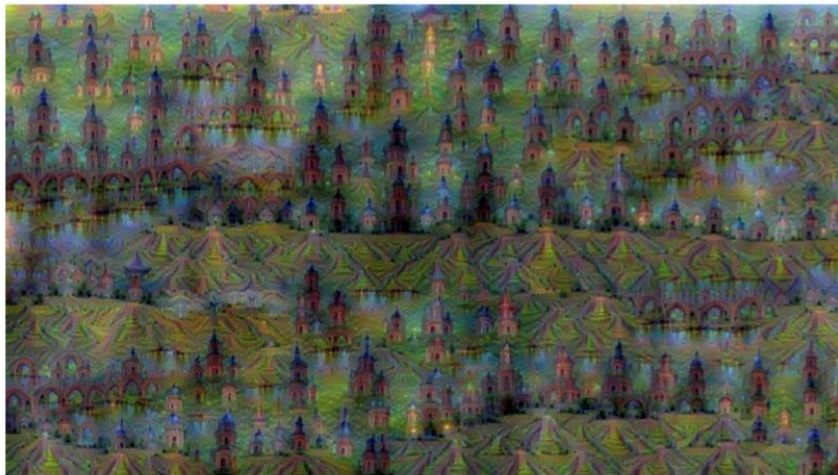
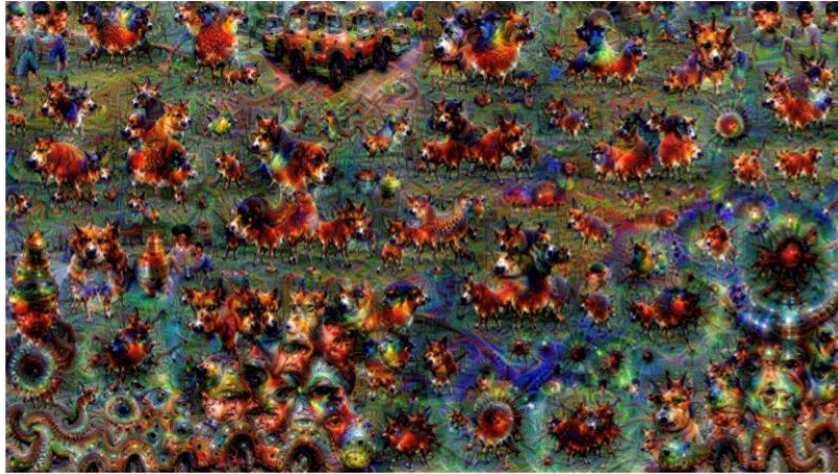
A woman standing on a beach holding a surfboard

All images are CC0 Public domain:

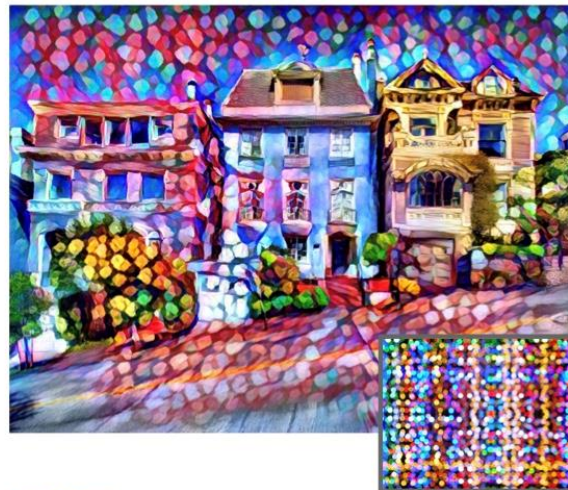
<https://pixabay.com/en/luggage-antique-cat-1643010/>
<https://pixabay.com/en/teddy-plush-bears-cute-teddy-bear-1623436/>
<https://pixabay.com/en/surf-wave-summer-sport-litoral-1668716/>
<https://pixabay.com/en/woman-female-model-portrait-adult-983967/>
<https://pixabay.com/en/handstand-lake-meditation-496008/>
<https://pixabay.com/en/baseball-player-shortstop-infield-1045263/>

Captions generated by Justin Johnson using [NeuralTalk2](#)

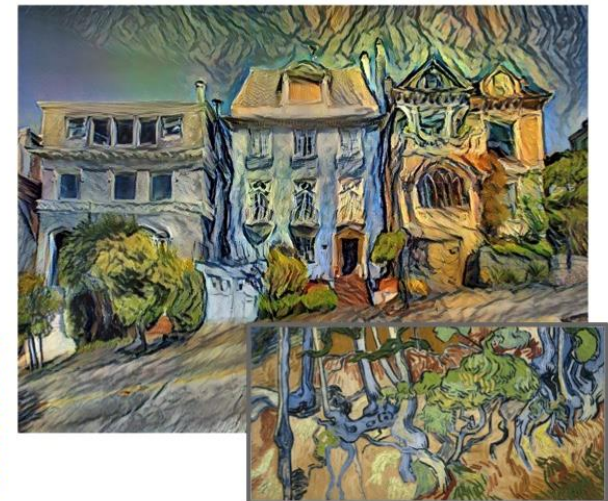
Style Transfer



Figures copyright Justin Johnson, 2015. Reproduced with permission. Generated using the Inceptionism approach from a [blog post](#) by Google Research.



Original image is CC0 public domain
[Starry Night](#) and [Tree Roots](#) by Van Gogh are in the public domain
[Bokeh image](#) is in the public domain
Stylized images copyright Justin Johnson, 2017;



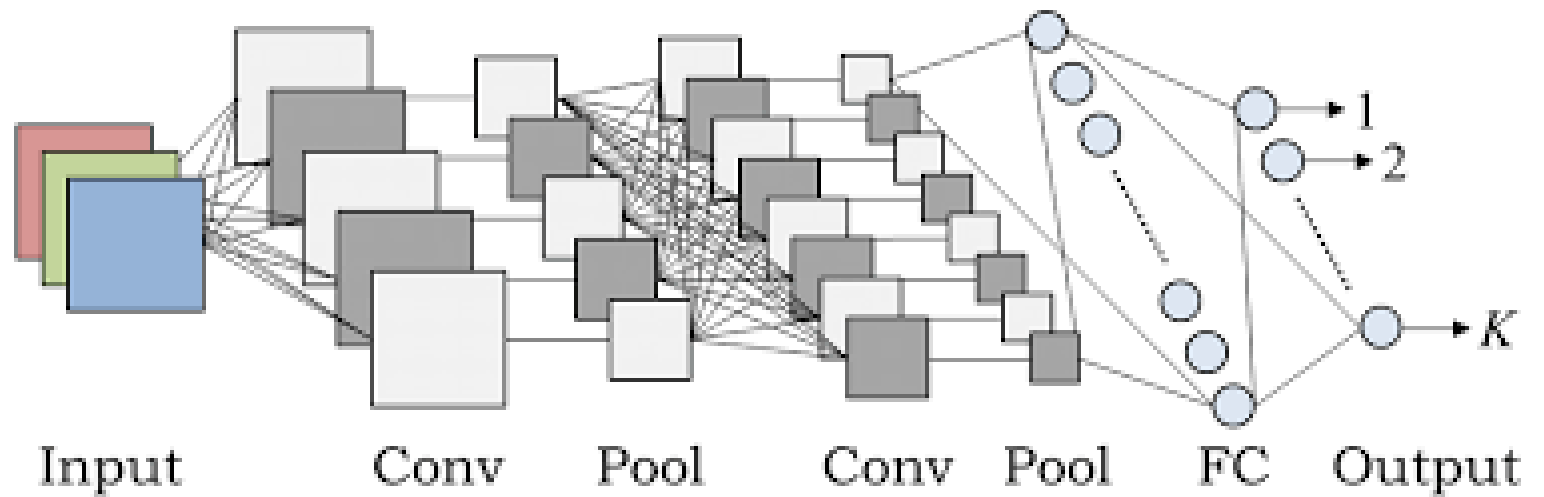
Gatys et al, "Image Style Transfer using Convolutional Neural Networks", CVPR 2016
Gatys et al, "Controlling Perceptual Factors in Neural Style Transfer", CVPR 2017

CNN

Convolutional Neural Networks
[합성곱 신경망]

여러 필터의 합성곱으로 이루어진 신경망

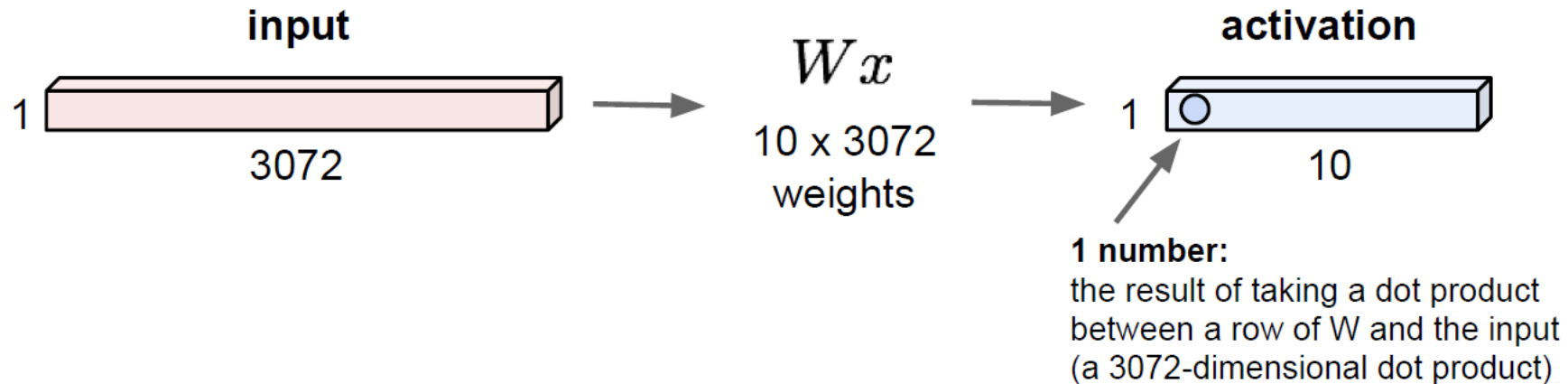
여러 개의 Convolutional Layer를 중첩시켜 Feature map(특징)을 추출



다시한번 Fully Connected 에서는...

Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1

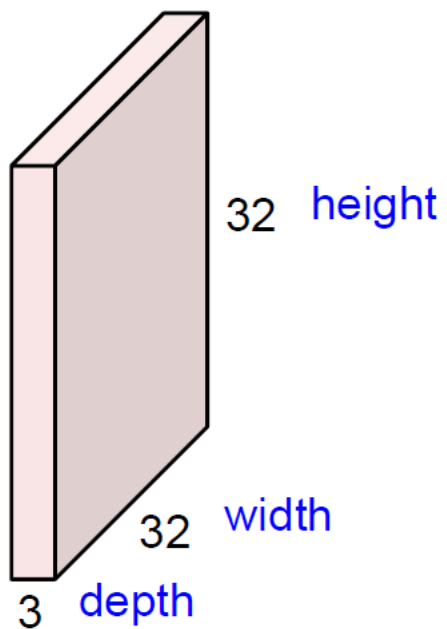


Convolution Layer

32x32x3 image -> preserve spatial structure

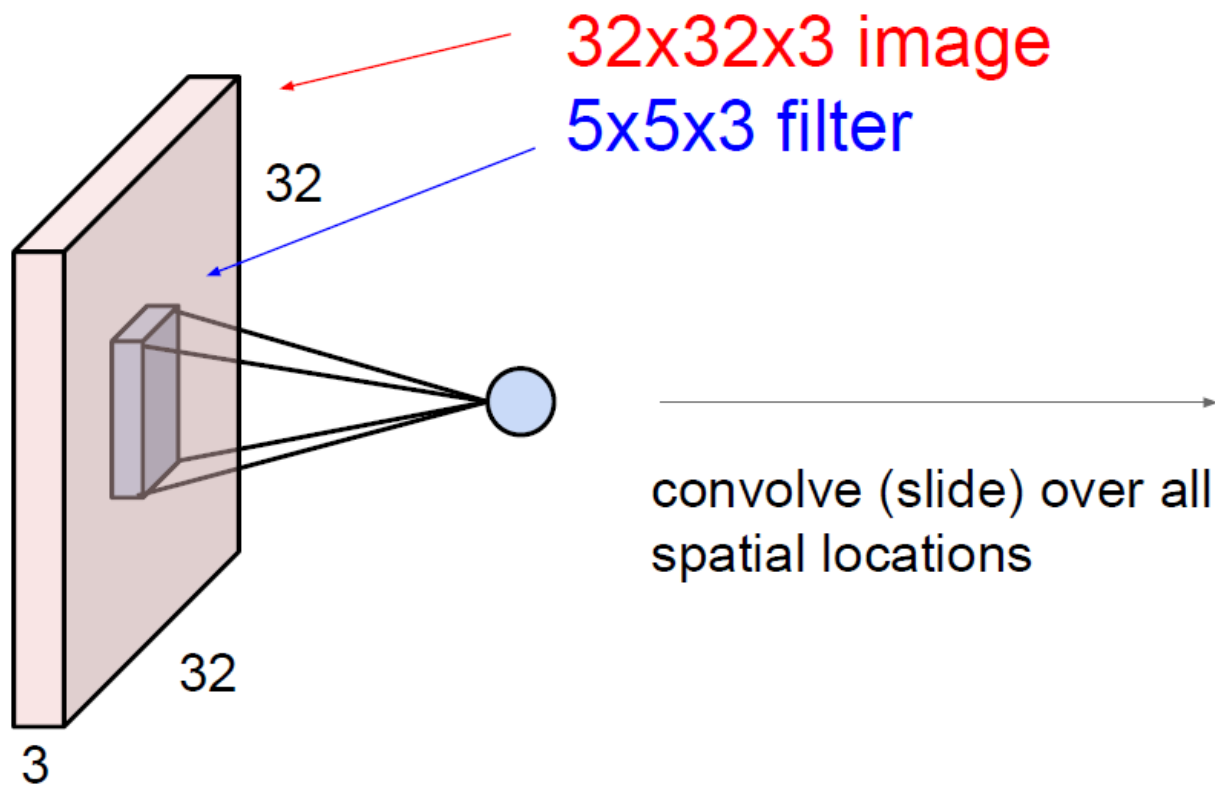
압축하지않고, 데이터의 구조 그대로 사용

Input과 filter의 volume은 항상 같아야 한다.

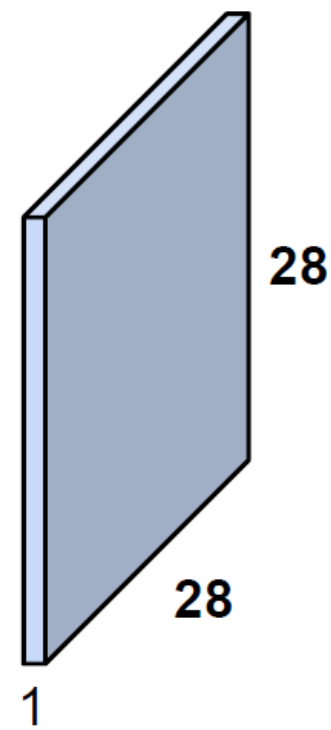


5x5x3 filter





activation map

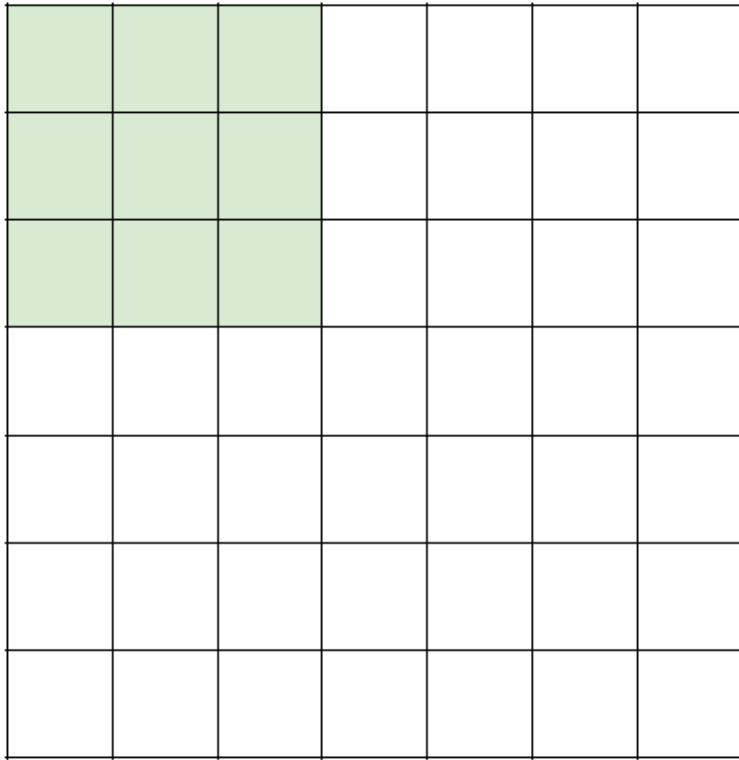


필터가 훑고 가면
크기가 작아진다?!

28x28x1의 output
생성

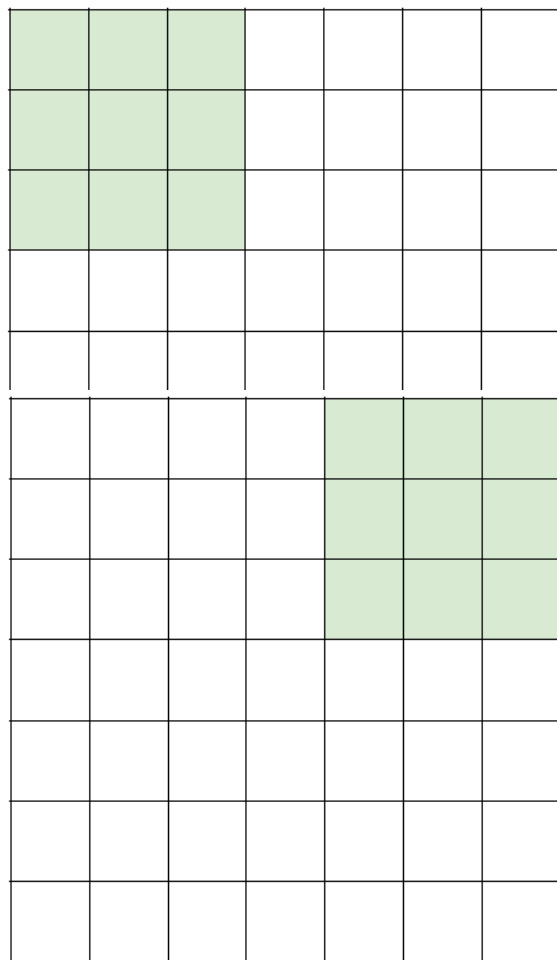
7x7 input with 3x3 filter

7

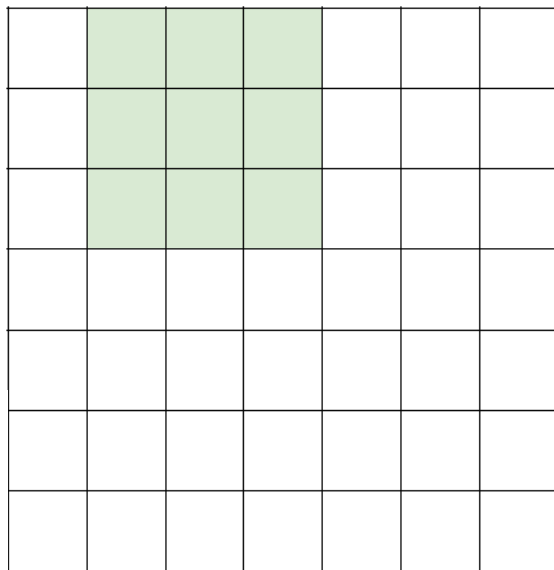


7

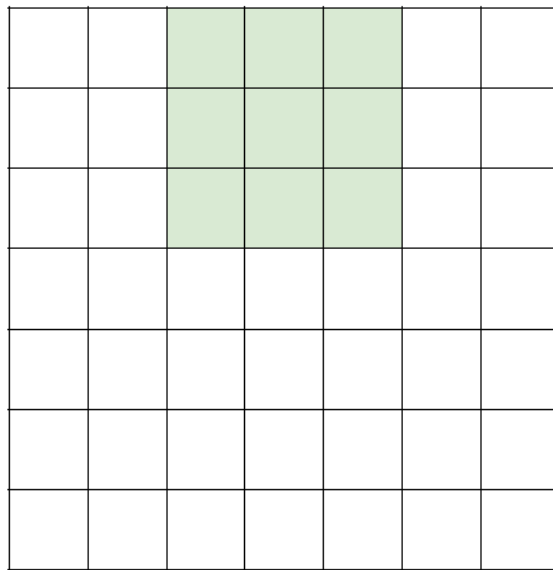
7



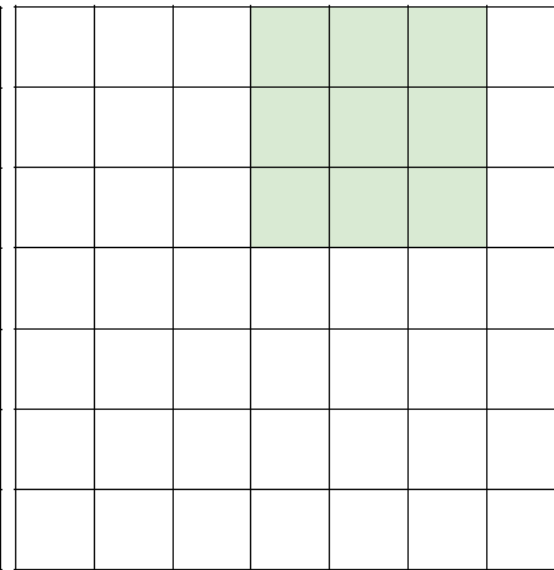
7



7



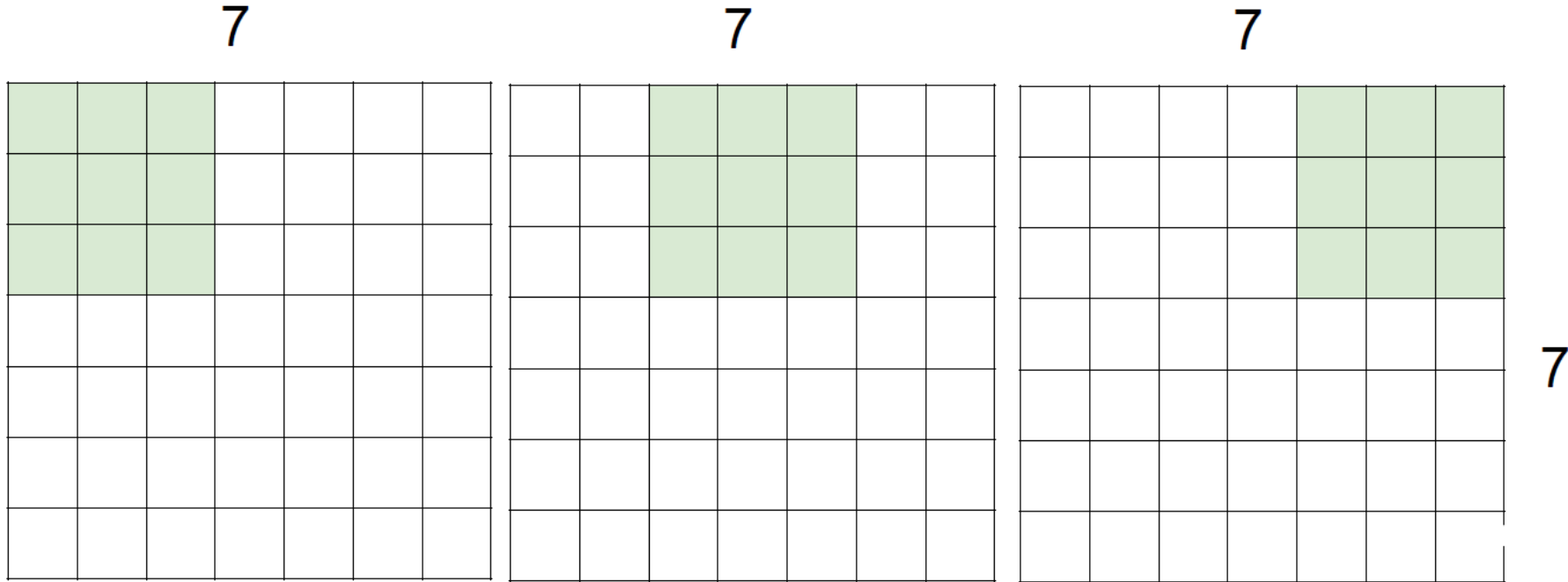
7



7

7x7 input을 3x3 필터로 훑게되면 5x5가 된다.

이번엔 두칸씩 넘어간다면?

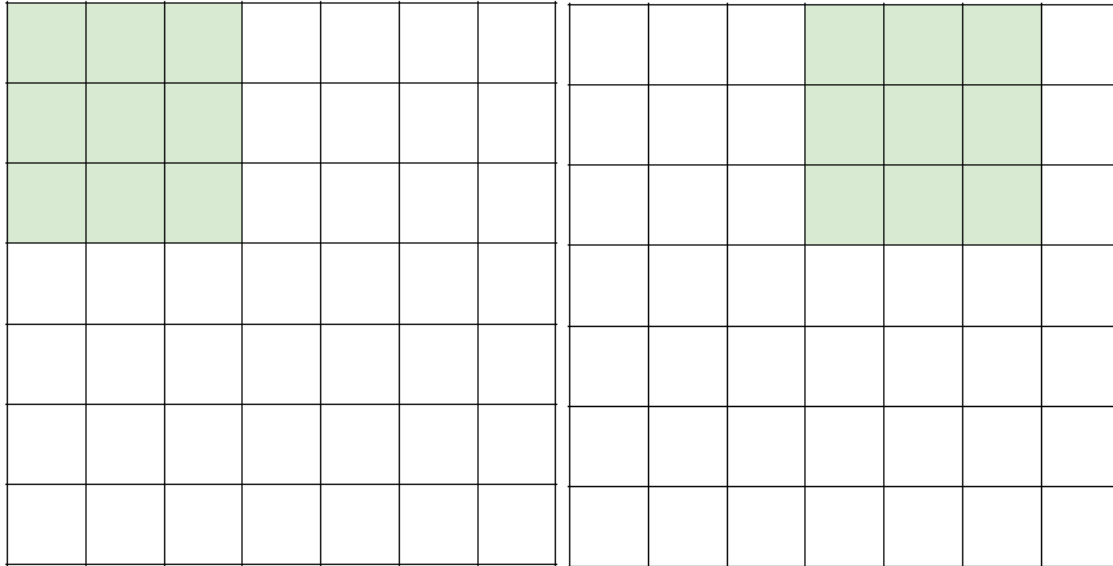


7x7 input을 3x3 필터로 두칸 간격으로 훑게되면 3x3가 된다.
두칸 간격 : stride = 2

세칸도 가능?

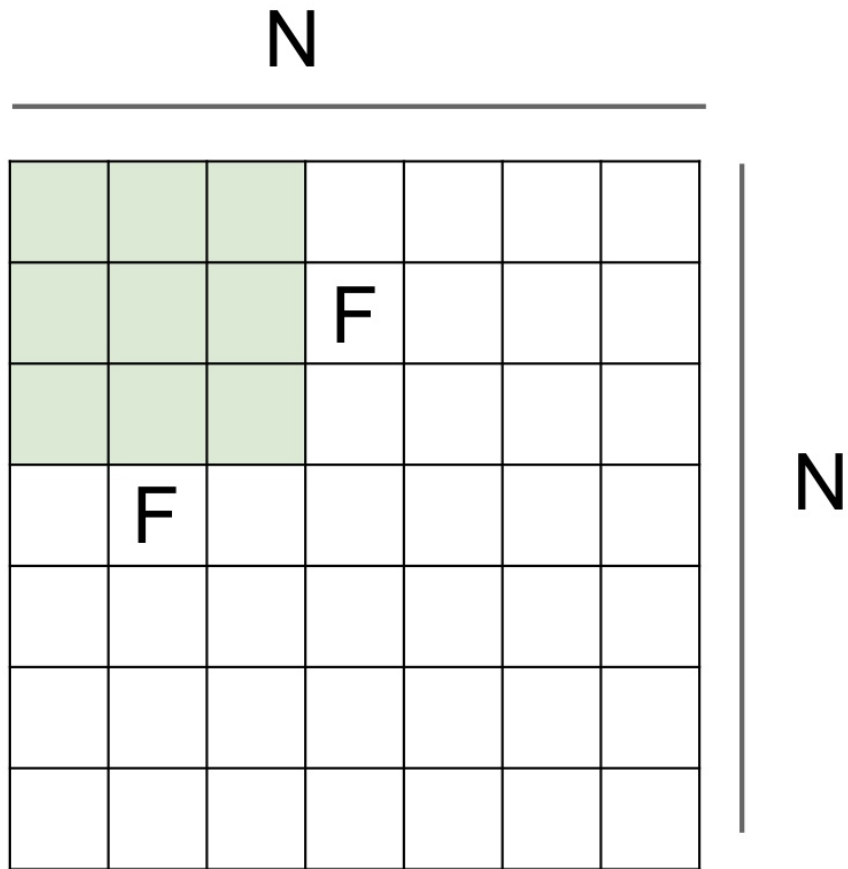
7

7



7

불가능...



Output size:

$$(N - F) / \text{stride} + 1$$

(한쪽크기 - 필터크기) / 걸음걸이 + 1
값이 자연수로 나와야한다!

e.g. $N = 7, F = 3$:

$$\text{stride } 1 \Rightarrow (7 - 3) / 1 + 1 = 5$$

$$\text{stride } 2 \Rightarrow (7 - 3) / 2 + 1 = 3$$

$$\text{stride } 3 \Rightarrow (7 - 3) / 3 + 1 = 2.33$$

당연한 소리지만 stride = 1 이면 항상 가능하겠죠?
하지만 stride가 작으면 보폭이 작아서
시간이 너무 오래걸려요!

Pad the Border [덧대기]

Image

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | | | | | | 0 |
| 0 | | | | | | 0 |
| 0 | | | | | | 0 |
| 0 | | | | | | 0 |
| 0 | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Filter를 거칠 때 마다
Output의 크기가
작아지는것을 방지

Zero pad가 가장 일반적

Output size =
 $(N + \#pad * 2 - F) / stride + 1$

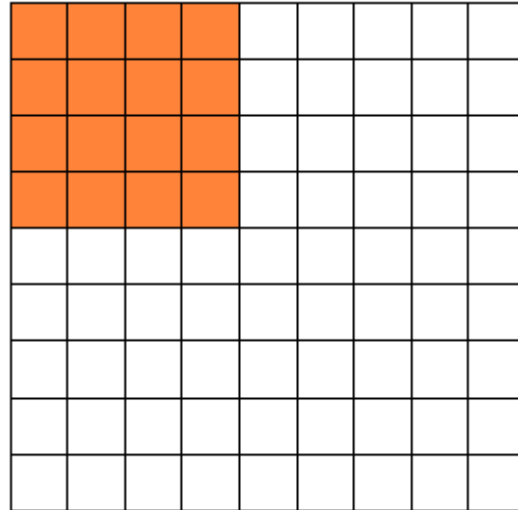
Quiz

Input : 9x9

Filter : 4x4

Stride : 1

Pad : X



일 때 Output size는?

답 : 6x6

Quiz

Input : 8x8

Filter : 4x4

Stride : 2

Pad : 1

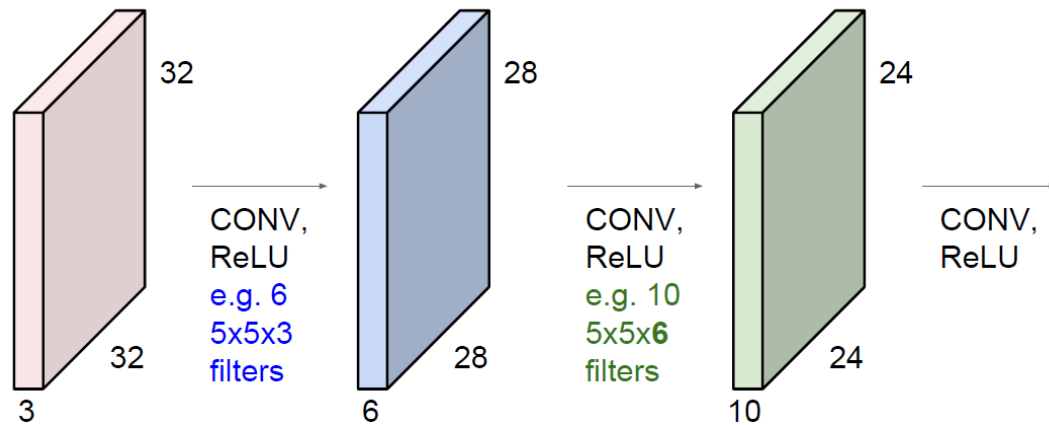
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | | | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

일 때 Output Size는?

답 : 4x4

기억해야 할 점은

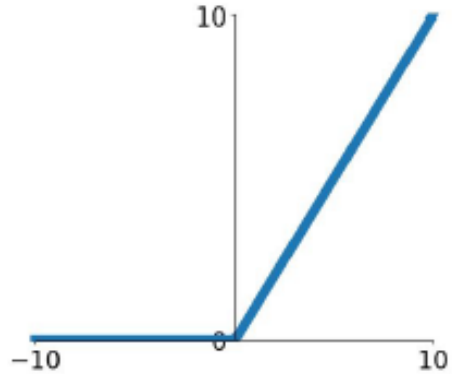
디자인을 어떻게 하냐에
따라서 output의 크기가 바뀌니까
신경쓰면서 디자인 할것!



Activation Function : ReLu

ReLU

$$\max(0, x)$$



하나의 Convolutional Layer를
지날때 마다 활성화함수 ReLu 적용

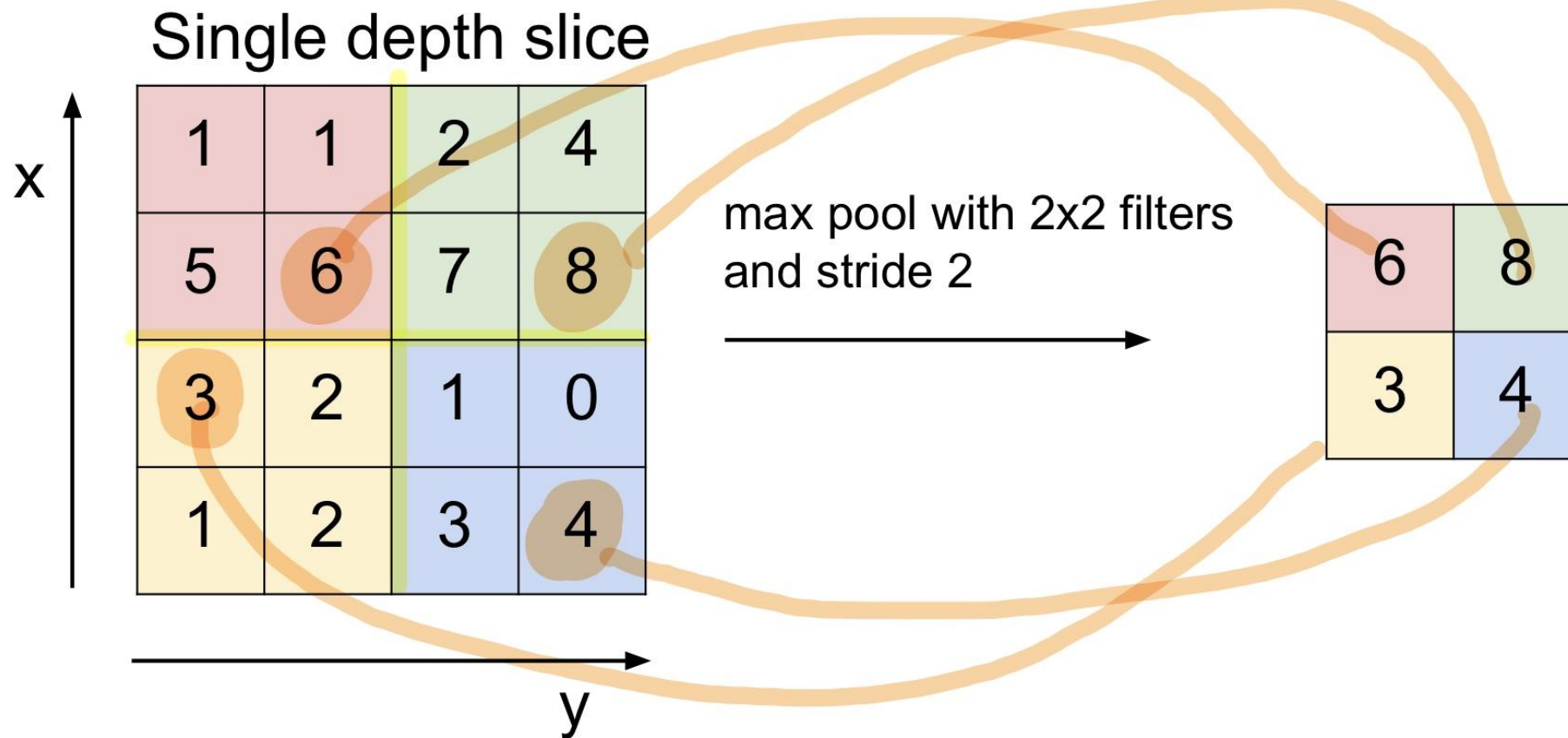
- 음의 값을 가지는 값은 모두 0으로 만들고
- 양의 값을 가지는 값은 그대로 적용
- 특징 추출에 유용하다고 한다.
- 간편하게 작동하고, 계산이 빠르다

Pooling Layer

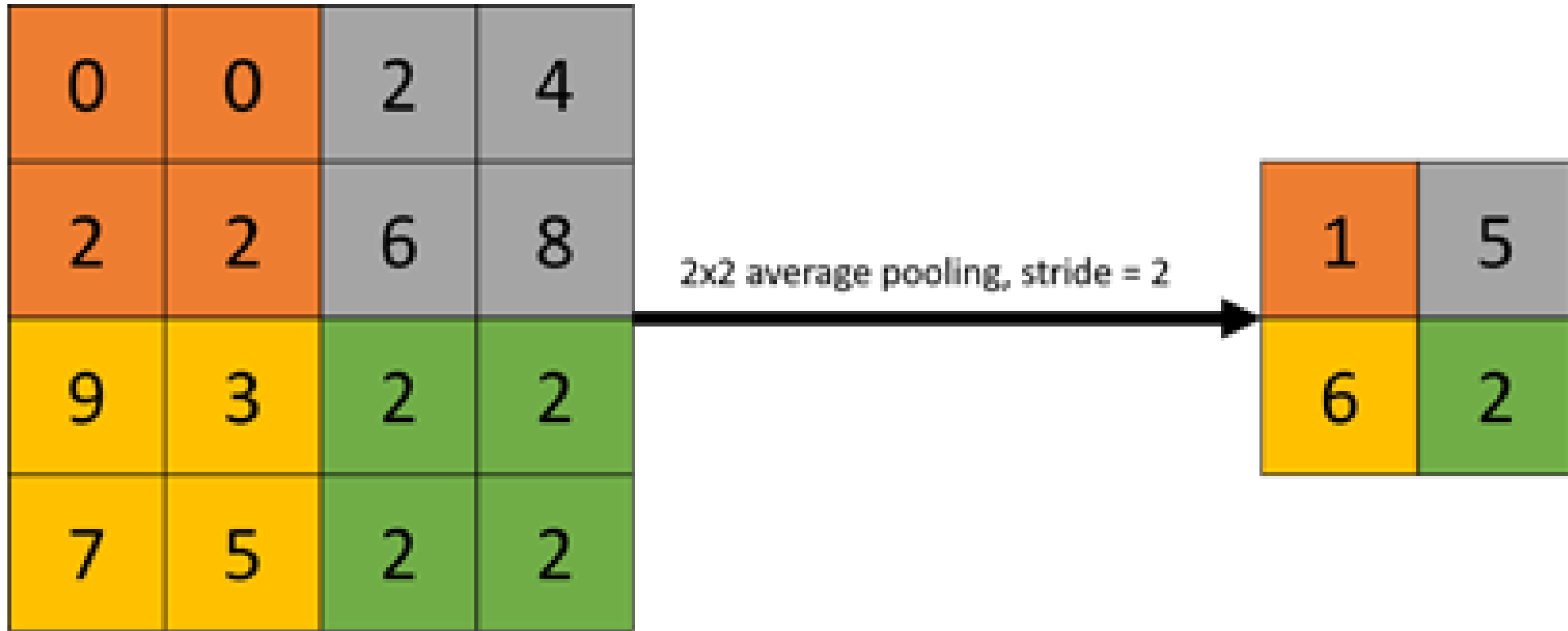
데이터를 압축하여 크기를 작게 만든다
설정한 필터 크기 안에서 대푯값 하나만 선택한다고 볼 수 있다

CNN에서 MAX POOLING을 쓰는 이유는 feature map의 값을
극대화 하기 위함

MAX POOLING



cf) AVERAGE POOLING

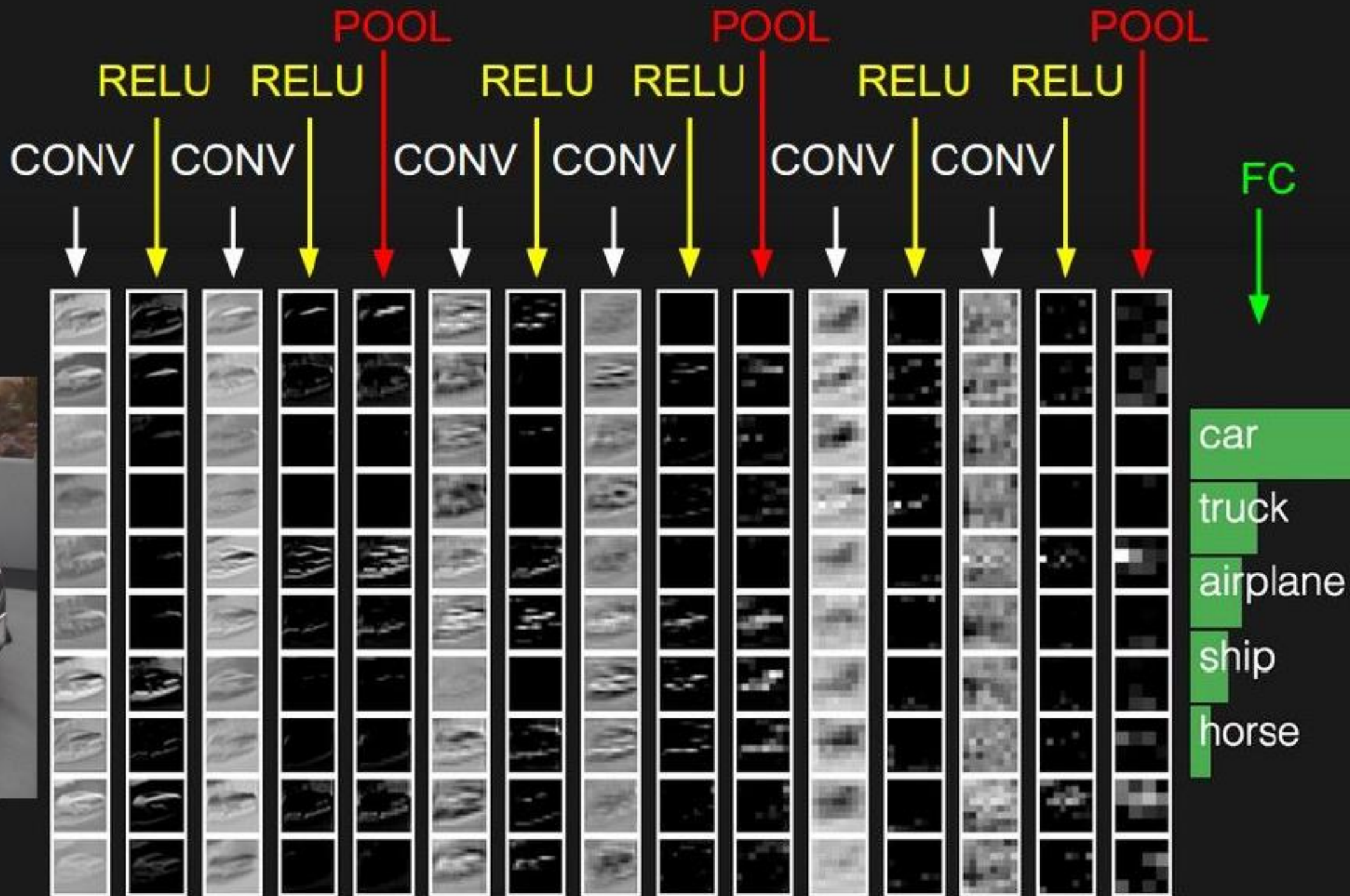


Fully Connected Layer [완전 연결층]

마지막은 FC층을 만들어 Classification 하는 구조를 만든다.

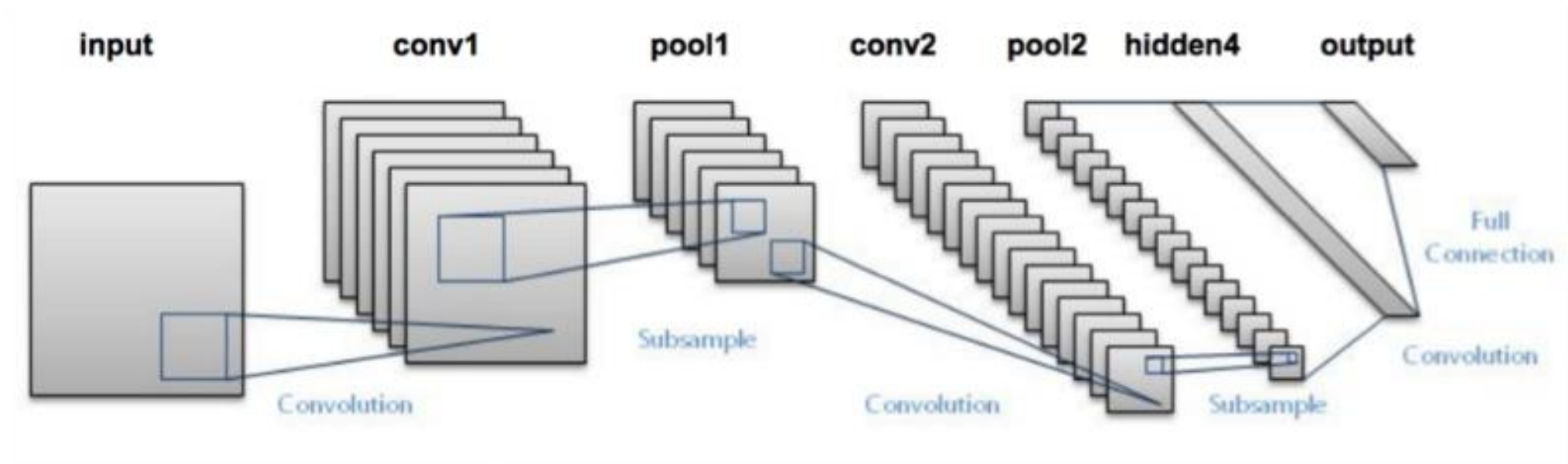
그동안은 Spatial(공간적)인 정보를 유지하면서 학습을 했고

마지막에는 aggregate(종합)해서 score output을 산출!

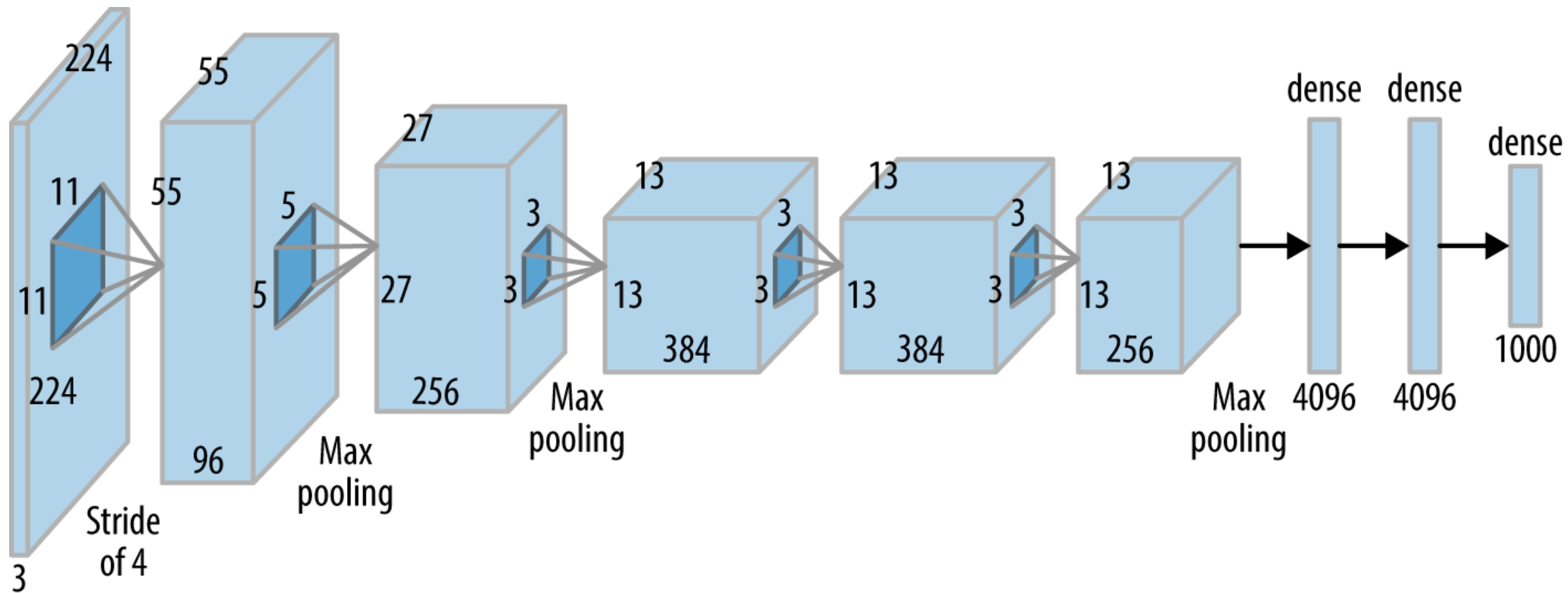


잘 만들어진 CNN모델들

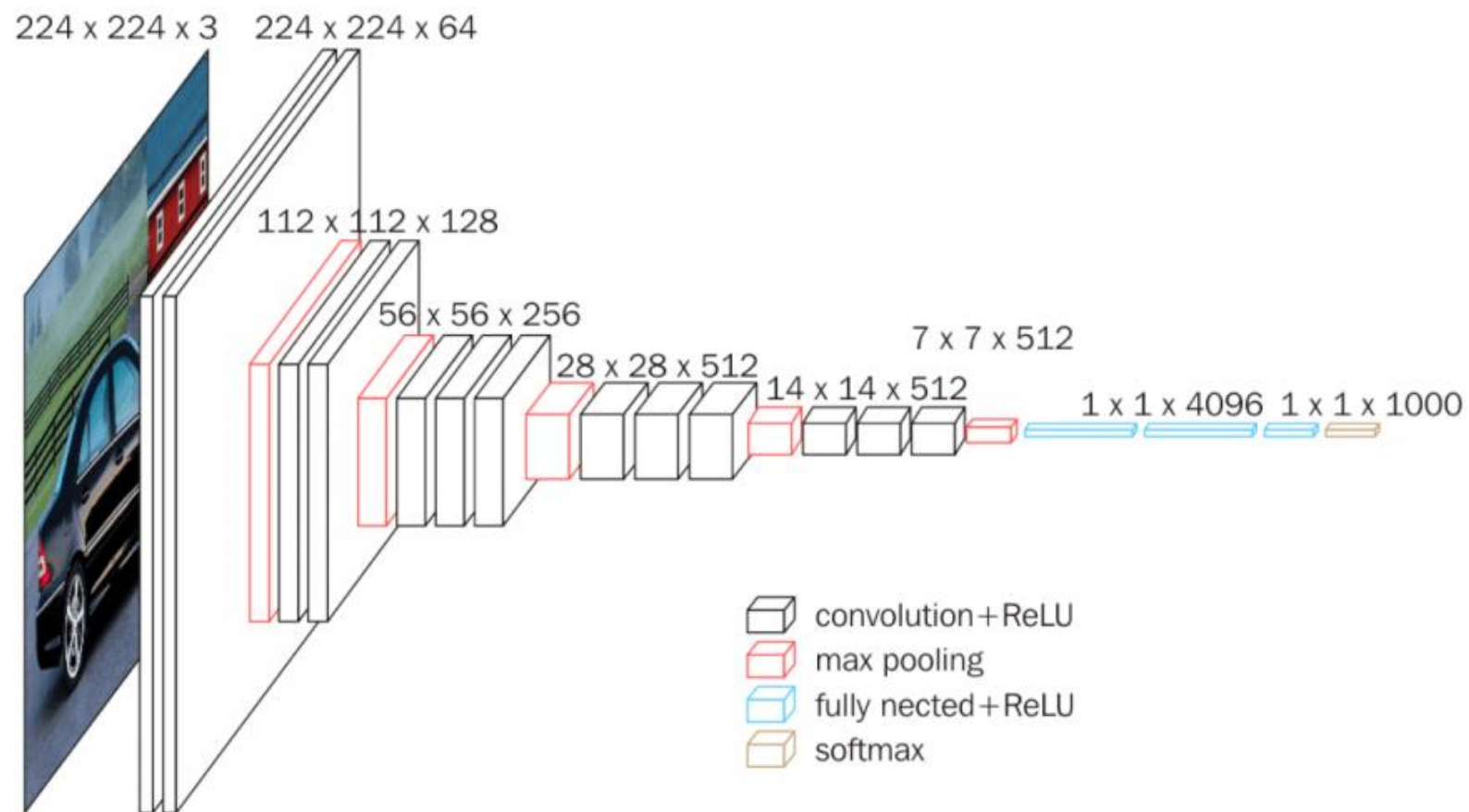
LeNet



AlexNet



VGGNet



감사합니다.
