- 사진강의 → image Convolution
- CNN
  - convolutional layer
  - max pooling layer

* 출력 이미지 사이즈 결정.
  ① 입력 이미지 size.
  - {kernel size}
  = padding
  = stride

Inclass 26: Convolutional Neural Network

[SCS4049] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

Department of Artificial Intelligence, Dongguk University

FC layer
- neuron 개수 명시.

다음주 월요일 12/06 : 비대면 기말고사

공지

- 카메라 세팅 기준: 오너터, 손, 얼굴, 책상위.
- 중간고사 문제 + 중간고사이후 강의 전부

Cover. 12/03 (금) 오후 2시 ~
(조금 늦을수 있음)

eclass
강의 zoom
(4주차 3강)

- 출석체크 안함, 필수X 의무X.
- 다시보기 제공 O,

- 기말고사 문제 내는중.
  - 문제 多多
  - 코딩 그렇게 많지 않을 수도?

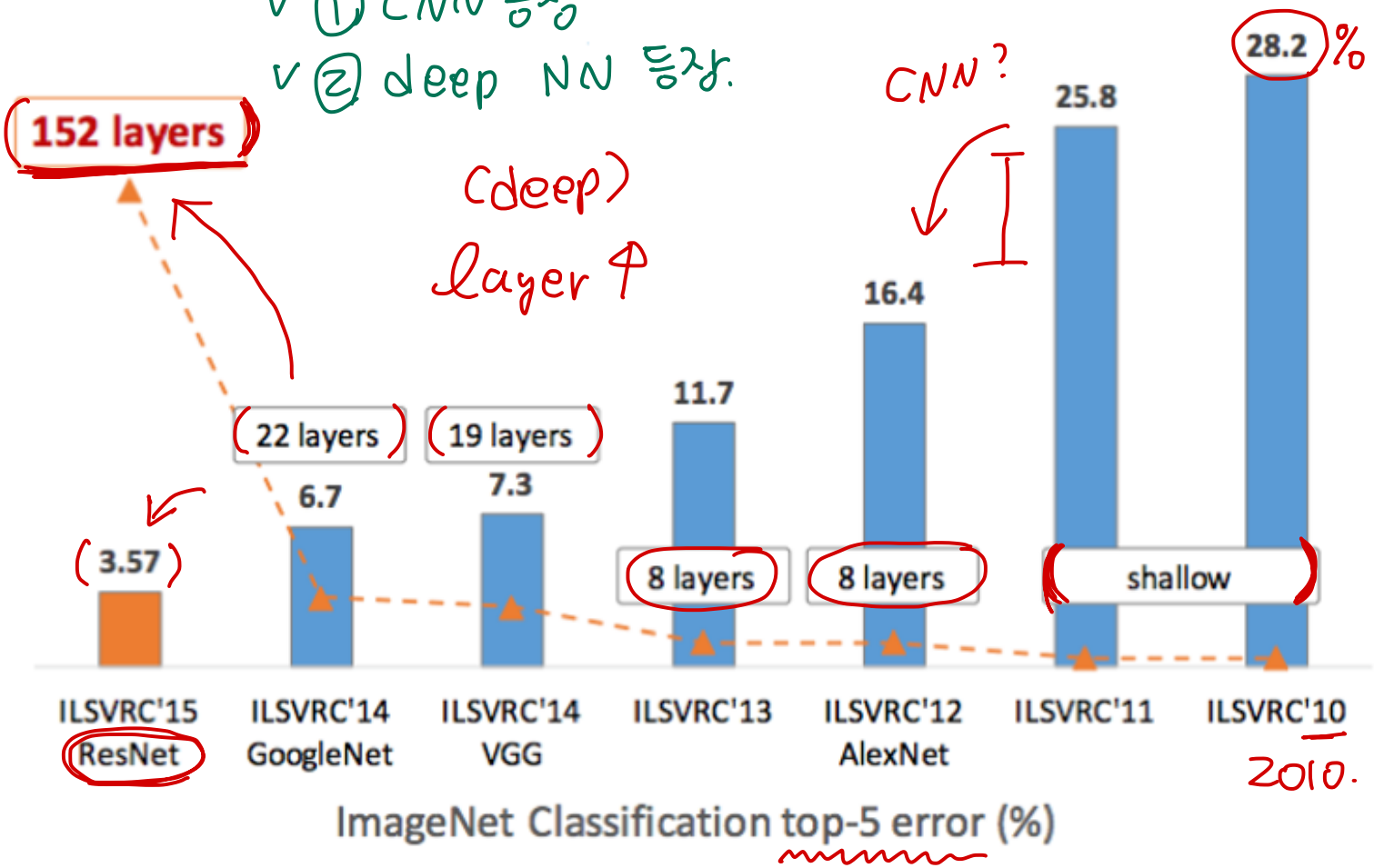
금요일에 줄다리기 할수도?

ImageNet challenge

image classification 유명한 데이터셋 대회인지.

v ① CNN 등장

v ② deep NN 등장.

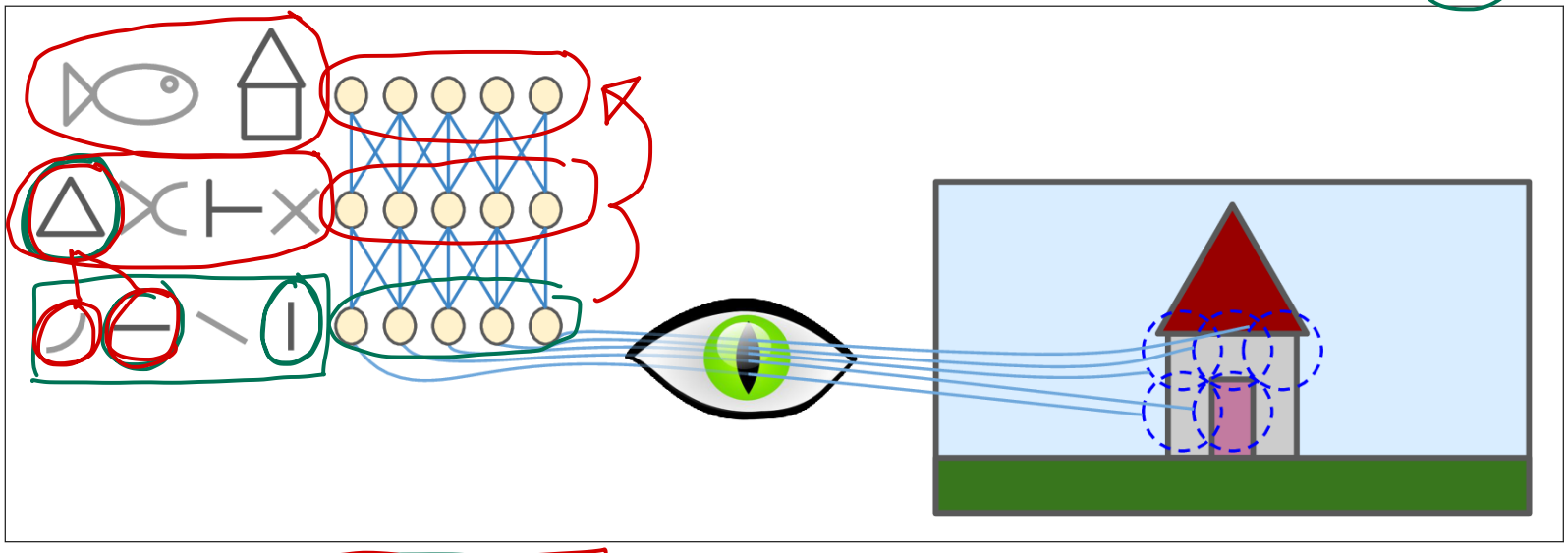


시간 ←

The architecture of the visual cortex

CNN : 우리의 시각에 있는 어떤 원리를 구현.

고차원



저차원

Figure 14-1. Local receptive fields in the visual cortex

⇒ image convolution. 구현

In particular, they showed that many neurons in the visual cortex have a small **local receptive field**, meaning they react only to visual stimuli located in a limited region of the visual field (see Figure 14-1, in which the local receptive fields of five neurons are represented by dashed circles). The receptive fields of different neurons may overlap, and together they tile the whole visual field.

Convolutional layer

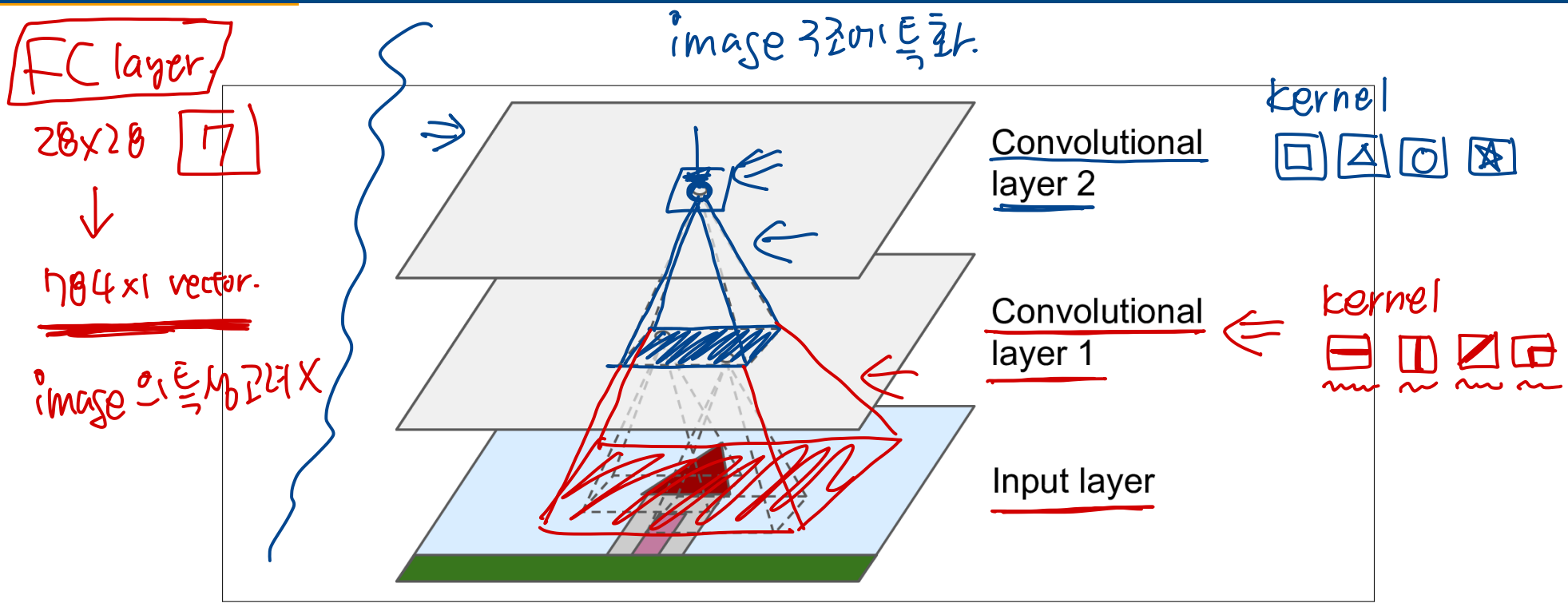


Figure 14-2. CNN layers with rectangular local receptive fields

The most important building block of a CNN is the **convolutional layer**:6 neurons in the first convolutional layer are not connected to every single pixel in the input image (like they were in previous chapters), but only to pixels in their receptive fields (see Figure 14-2). In turn, each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer. This architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on.

Convolutional neural network

- 컨볼루션 신경망(CNN)은 다층 신경망(multi-layer neural networks)의 특별한 형태
- CNN은 시각 피질의 국소 수용역(Local Receptive Field, LRF)과 시신경 세포의 방향 선택적 성격의 발견에 의해서 신경 생물학적인 영감으로 만들어짐
- 내재적이면서 자동적으로 유의미한 특징(feature)들을 추출하는 신경망 구조
- CNN 은 이미지로부터 위상학적 성질들을 추출할 수 있는 피드포워드(feed-forward) 신경망 convolution ~ linear operator
- 다른 모든 신경망들처럼 CNN 은 변형된 역전파 알고리즘을 사용하여 학습
- CNN 은 최초한의 전처리 만으로도 픽셀 이미지들로부터 시각적 패턴을 직접 인식할 수 있도록 설계
- 극도로 변화가 많은 패턴들도 인식 가능함 (예: 손글씨)

neural net → 학습데이터 backpropagation → weight W를 얻음.

CNN → 학습데이터 backpropagation → convolution kernel을 얻음.

주어진 이미지에서 어떤 특성을 뽑아야할지 고민할 필요가 없다.

MNIST 데이터 손글씨 → 어떤 특성을 추출해야 숫자를 잘 인식? → 고민할 필요X

Convolution

Given a filter kernel $\mathcal{H}$, the convolution of the kernel with image $\mathcal{F}$ is an image $\mathcal{R}$. The (i, j)-th component of $\mathcal{R}$ is given by

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}. \tag{1}$$

- **kernel** of the filter: the pattern of weights used for a linear filter
- **convolution**: the process of applying the filter

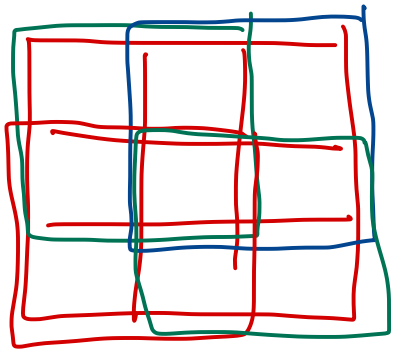
This operation is called **convolution**

$$R(f) = (h * f) \tag{2}$$

- *commutative*: $(g * h)(x) = (h * g)(x)$
- *associative*: $(f * (g * h)) = ((f * g) * h)$
- *distributive*: $f * (g + h) = f * g + f * h$

convolution

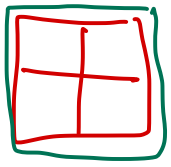
input



3x3

x

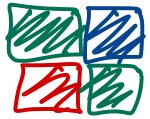
kernel



2x2

=

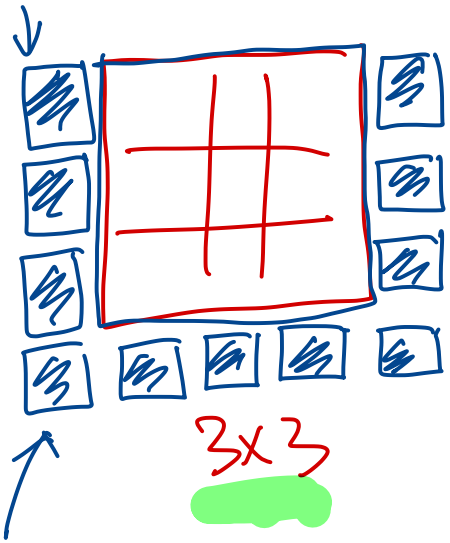
result



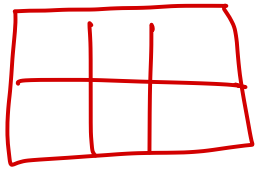
2x2.

+padding : input size와 output size를 동일하게 만들자.

input x kernel = output



3x3



2x3

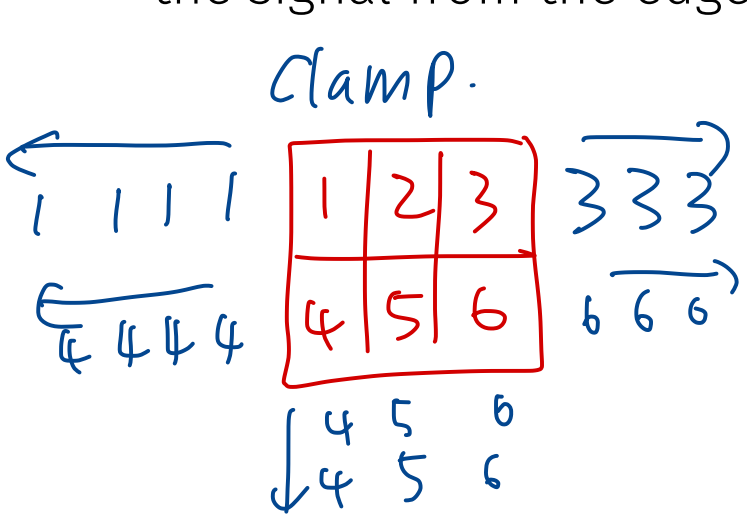
3x3

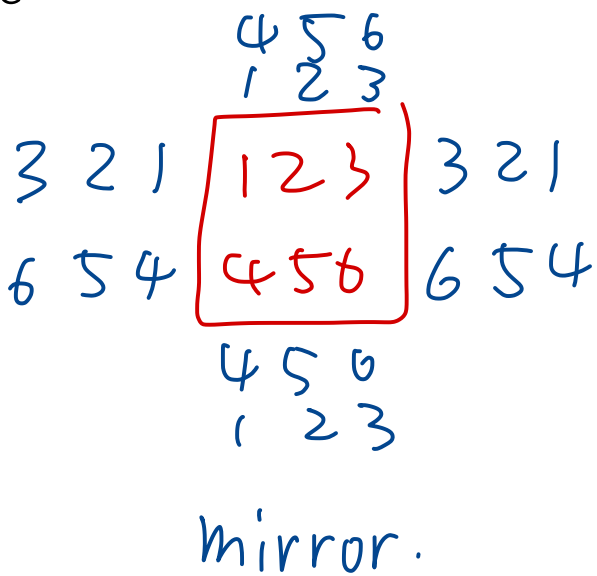
padding : 이미지의 경계에다가 값을 추가함.

필요한 만큼 (kernel size에 영향)

Padding (border effects)

- **zero**: set all pixels outside the source image to 0
- **constant**: set all pixels outside the source image to a specified border value
- **clamp**: repeat edge pixels indefinitely
- **wrap**: loop "around" the image in a "toroidal" configuration
- **mirror**: reflect pixels across the image edge
- **extend**: extend the signal by subtracting the mirrored version of the signal from the edge pixel value





Padding (border effects)

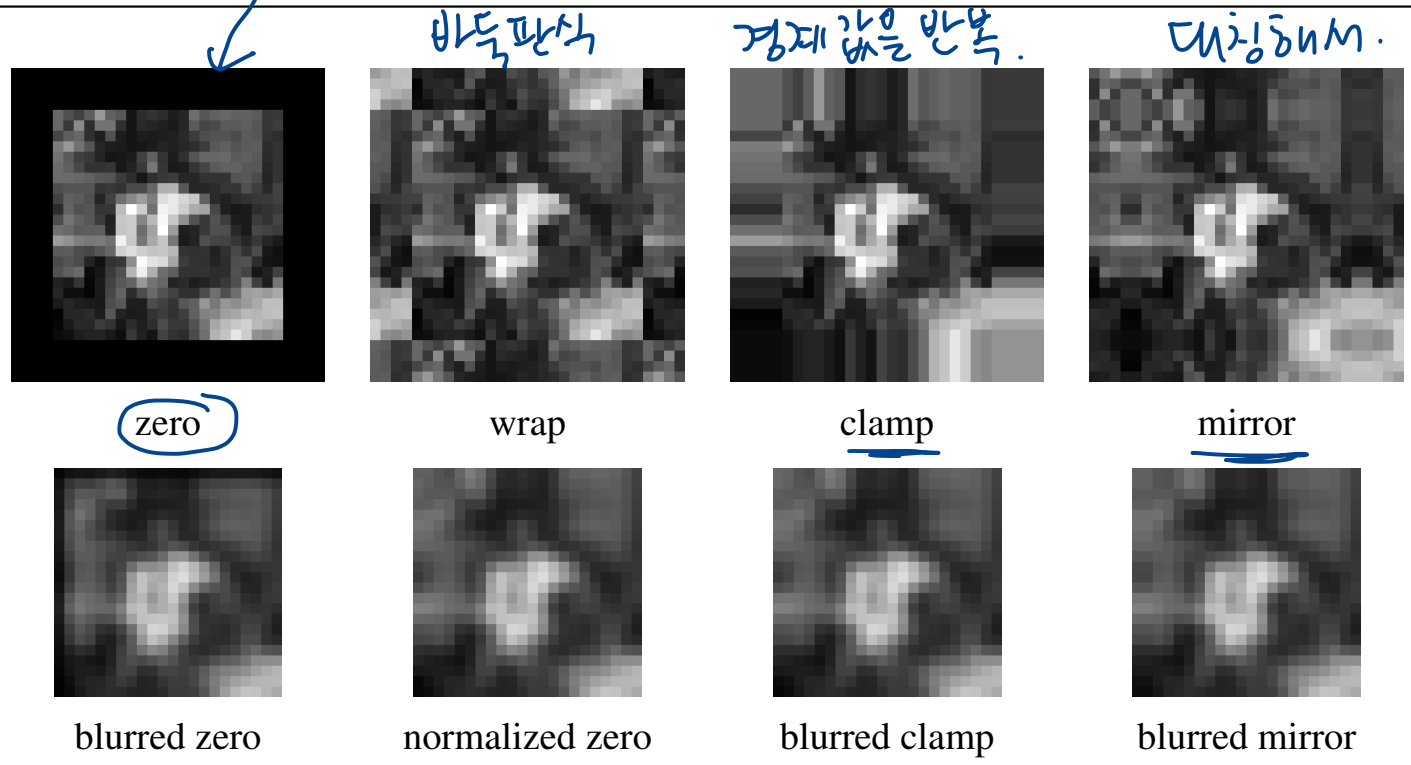


Figure 3.13 *Border padding (top row) and the results of blurring the padded image (bottom row). The normalized zero image is the result of dividing (normalizing) the blurred zero-padded RGBA image by its corresponding soft alpha value.*

Convolution: connections and padding

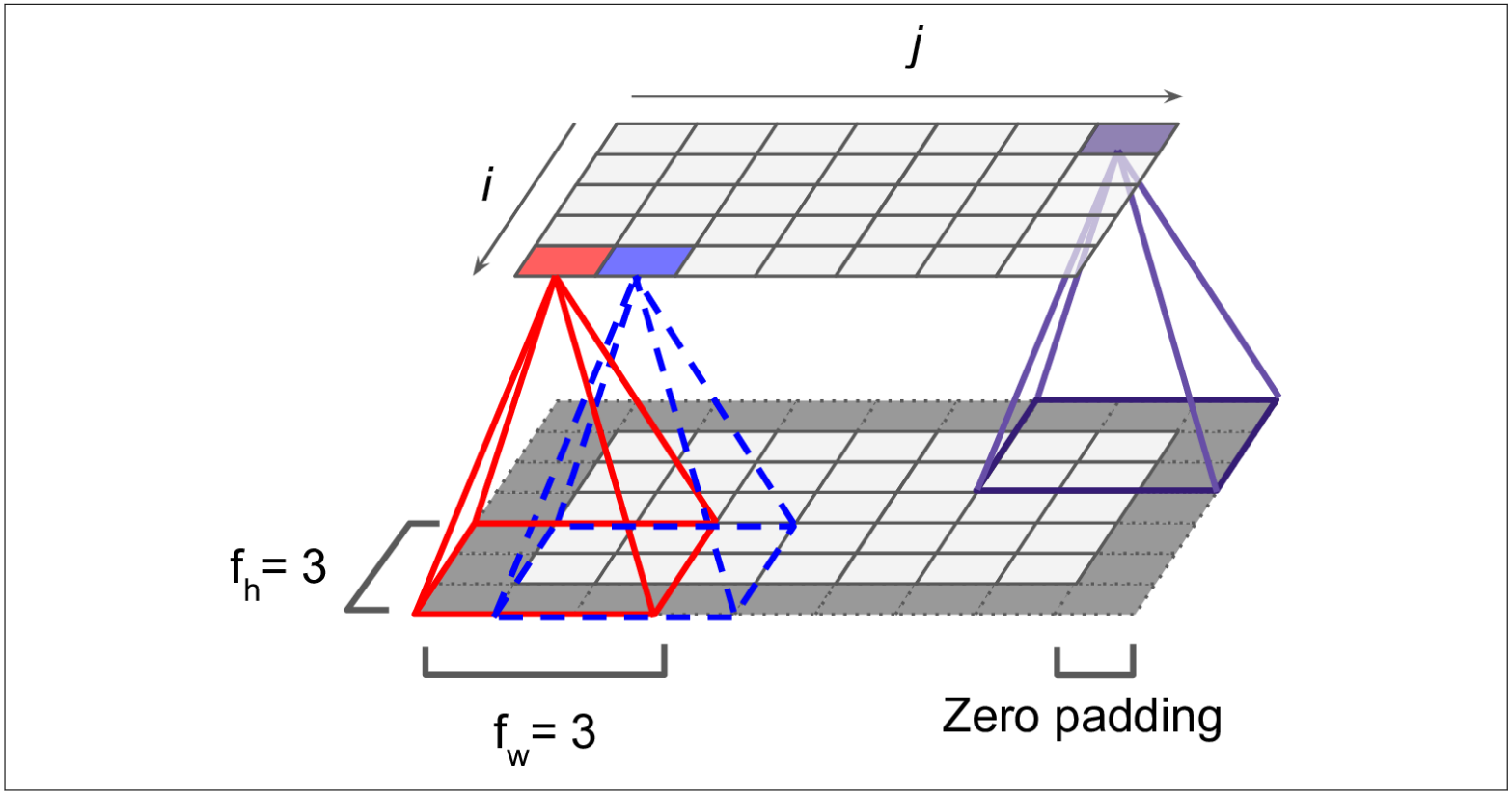


Figure 14-3. Connections between layers and zero padding

Convolution: stride and dimensional reduction

Stride: 가로·세로 방향으로 몇 칸씩 건너뛰며 convolution하나?

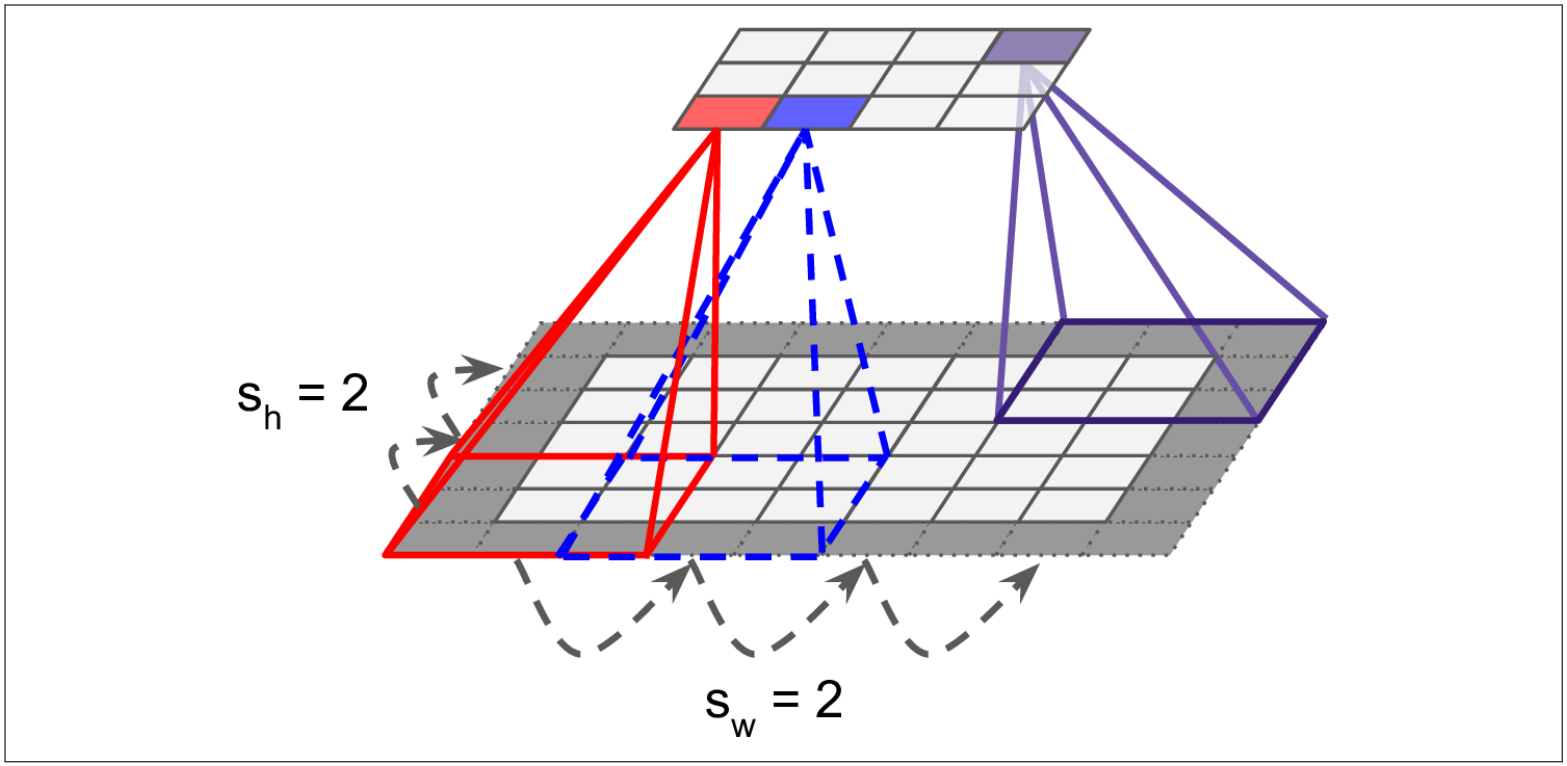
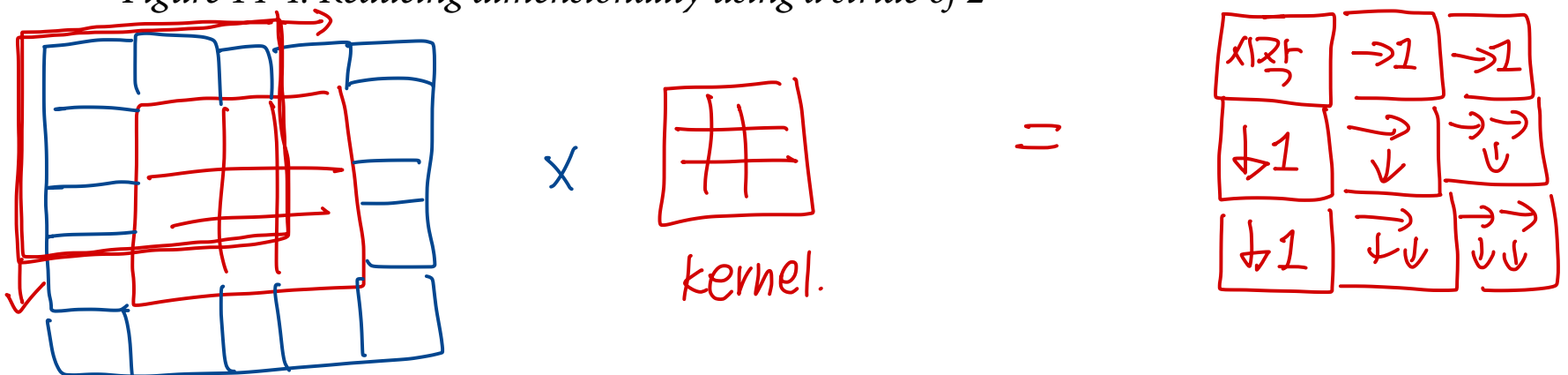


Figure 14-4. Reducing dimensionality using a stride of 2



우리가 convolution은 : 행은 +1칸씩, 열은 +1칸씩.

stride mxn은 : 행은 m칸씩, 열은 n칸씩.

input (padding적용)	kernel	stride.	output.
5x4	2x2	1x2	5x2
5x4	3x3	1x2	5x2
12x6	4x2	3x2	4x3

Padding : 기본적으로 input = output 동일하게.

필요한 만큼 들어가면 됨.

디폴트로 적용함. 신경쓸일 없다.

Stride : 행·열 방향으로 얼만큼 간격을 둘건지.

Stride에 반비례로 output size 줄어듦.

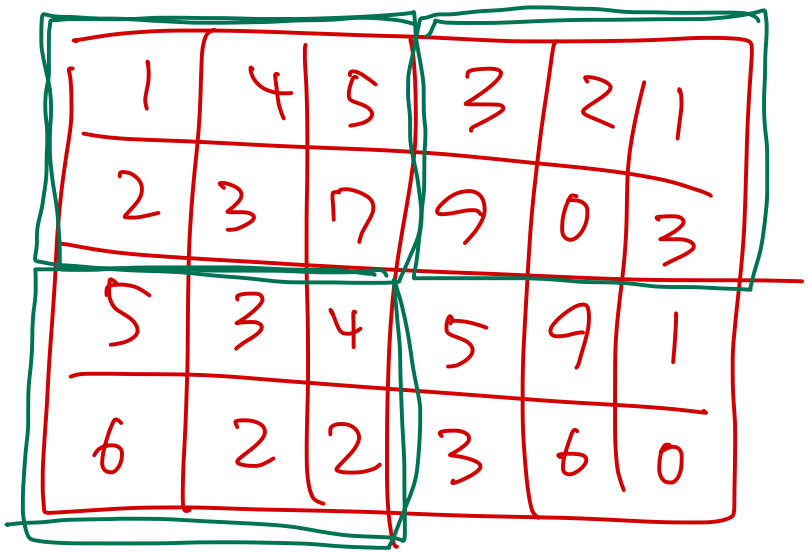
output size ← input size, stride size.

(padding 0)

kernel, stride.

pooling : 이미지의 size를 줄일 때 인접한 픽셀들의 대표값으로.

max pooling :



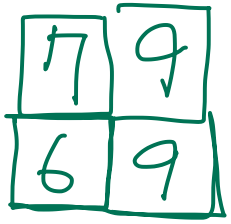
4x6

max pooling

size 2x3

(=stride 2x3)

동일한 stride끼리만.



2x2.

Convolution: example

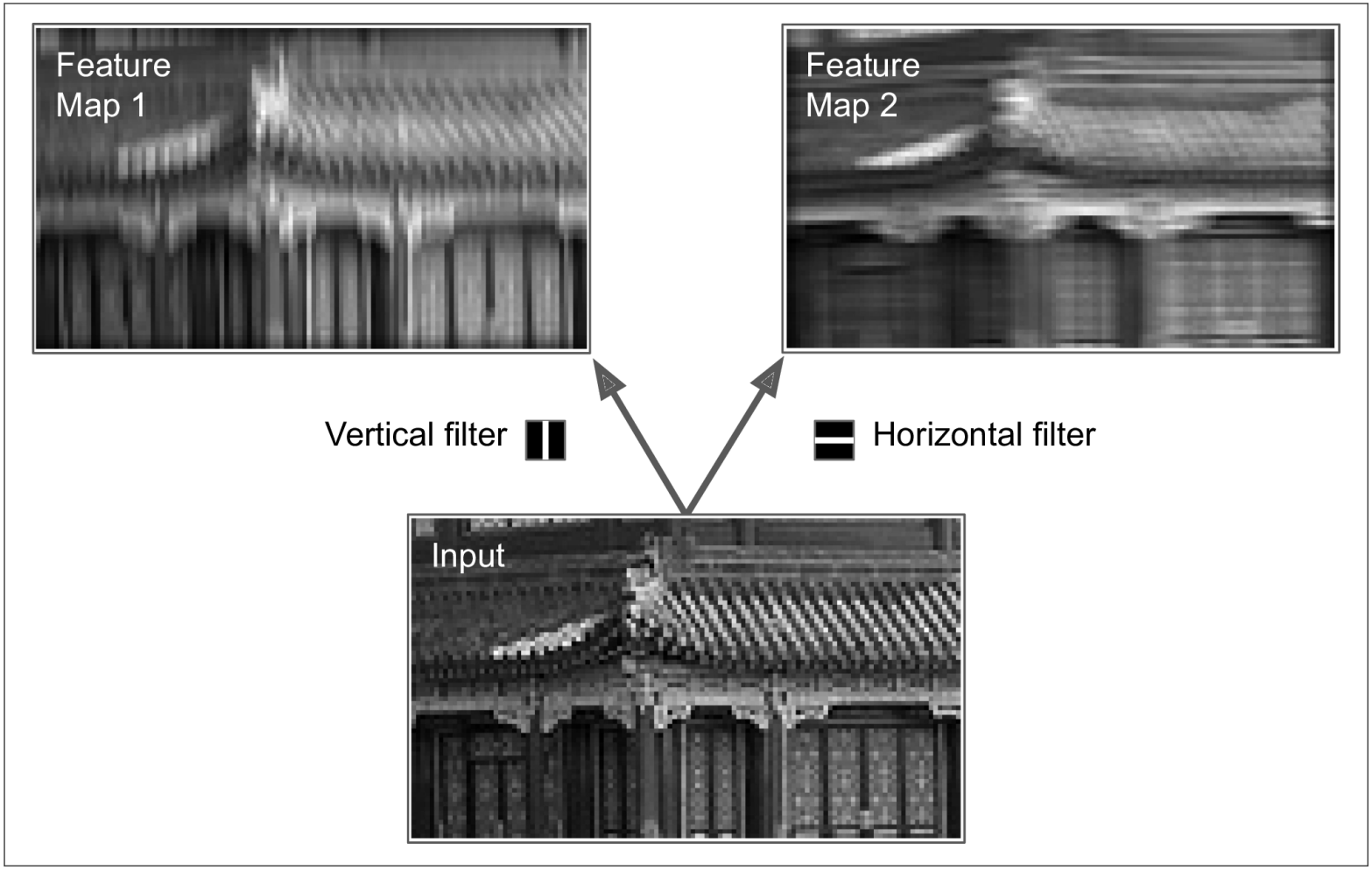
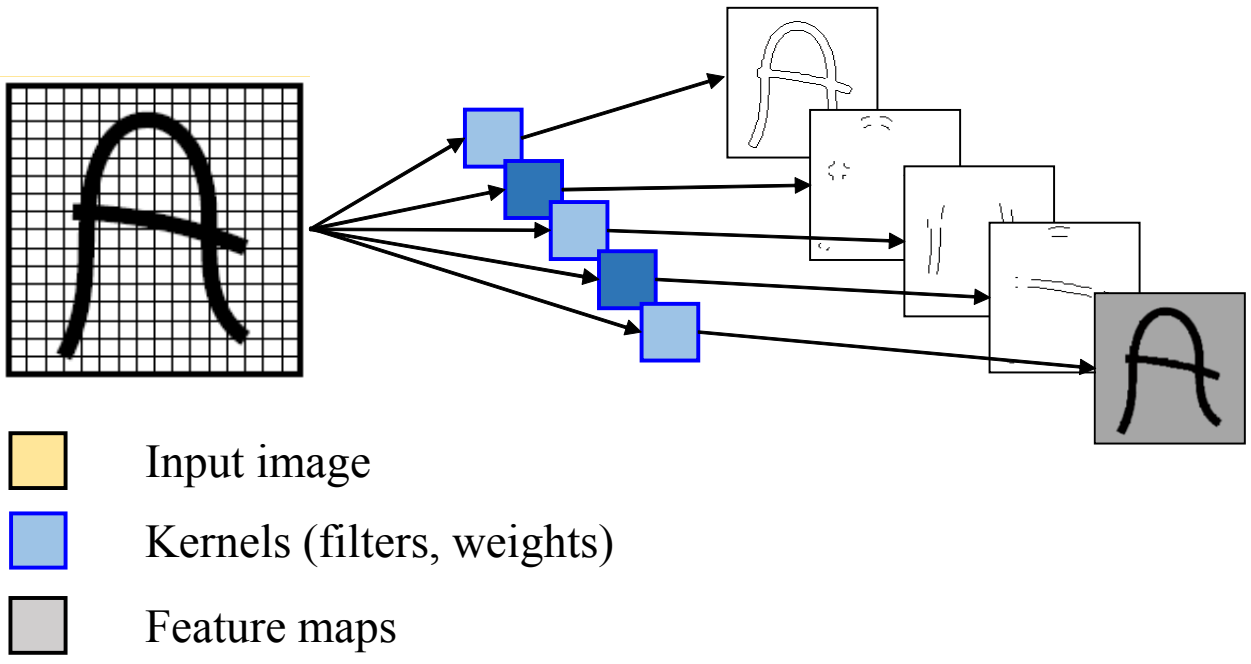


Figure 14-5. Applying two different filters to get two feature maps

Convolution: filter effect

Feature extraction by different kernels



- 컨볼루션은 입력 이미지의 다른 부분들에 존재하는 동일한 특징들을 찾아냄
- Feature map에 있는 뉴런은 template or kernel과 일치되는 경우에 활성화

CNN: convolution layers with multiple feature maps

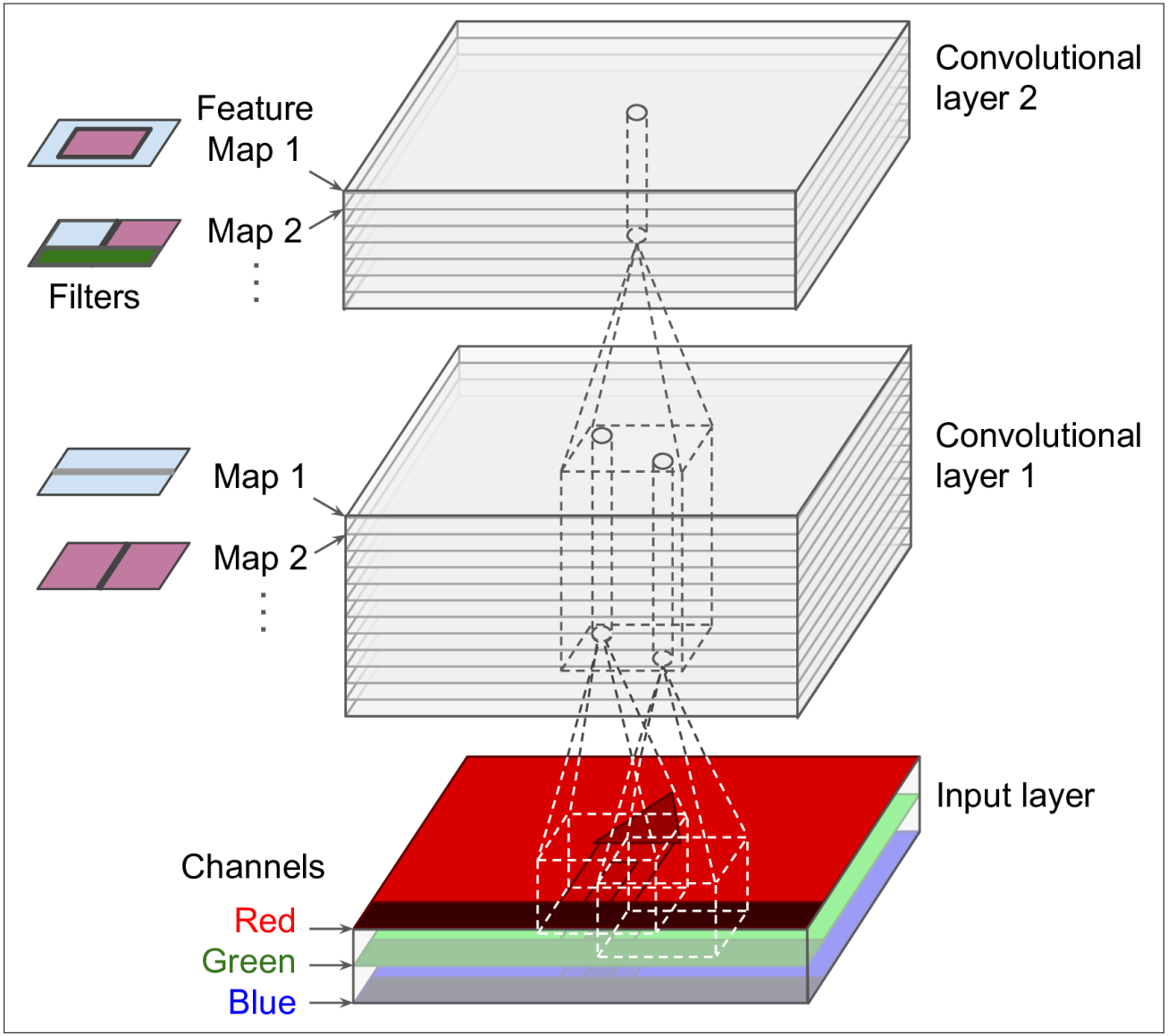
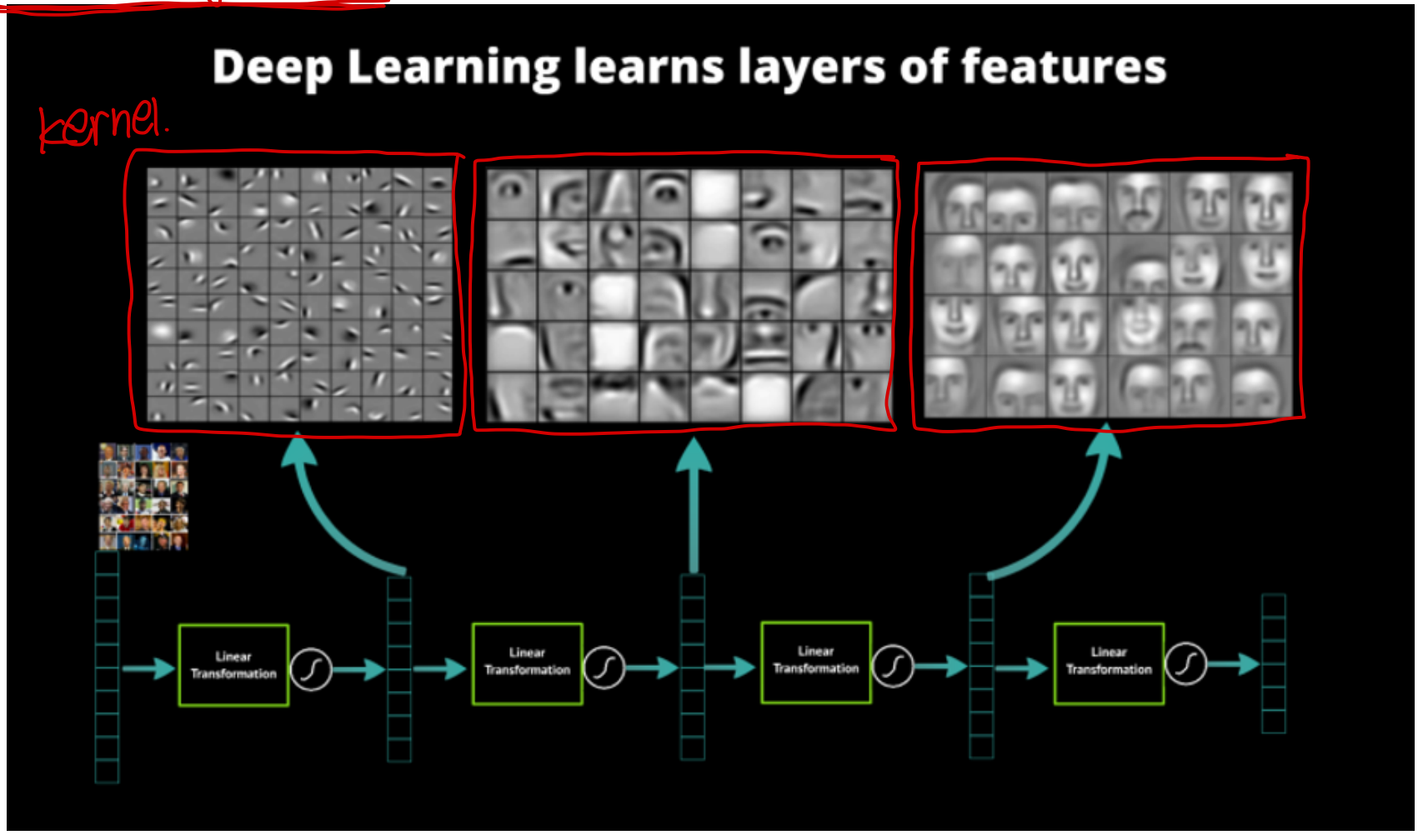


Figure 14-6. Convolution layers with multiple feature maps, and images with three color channels

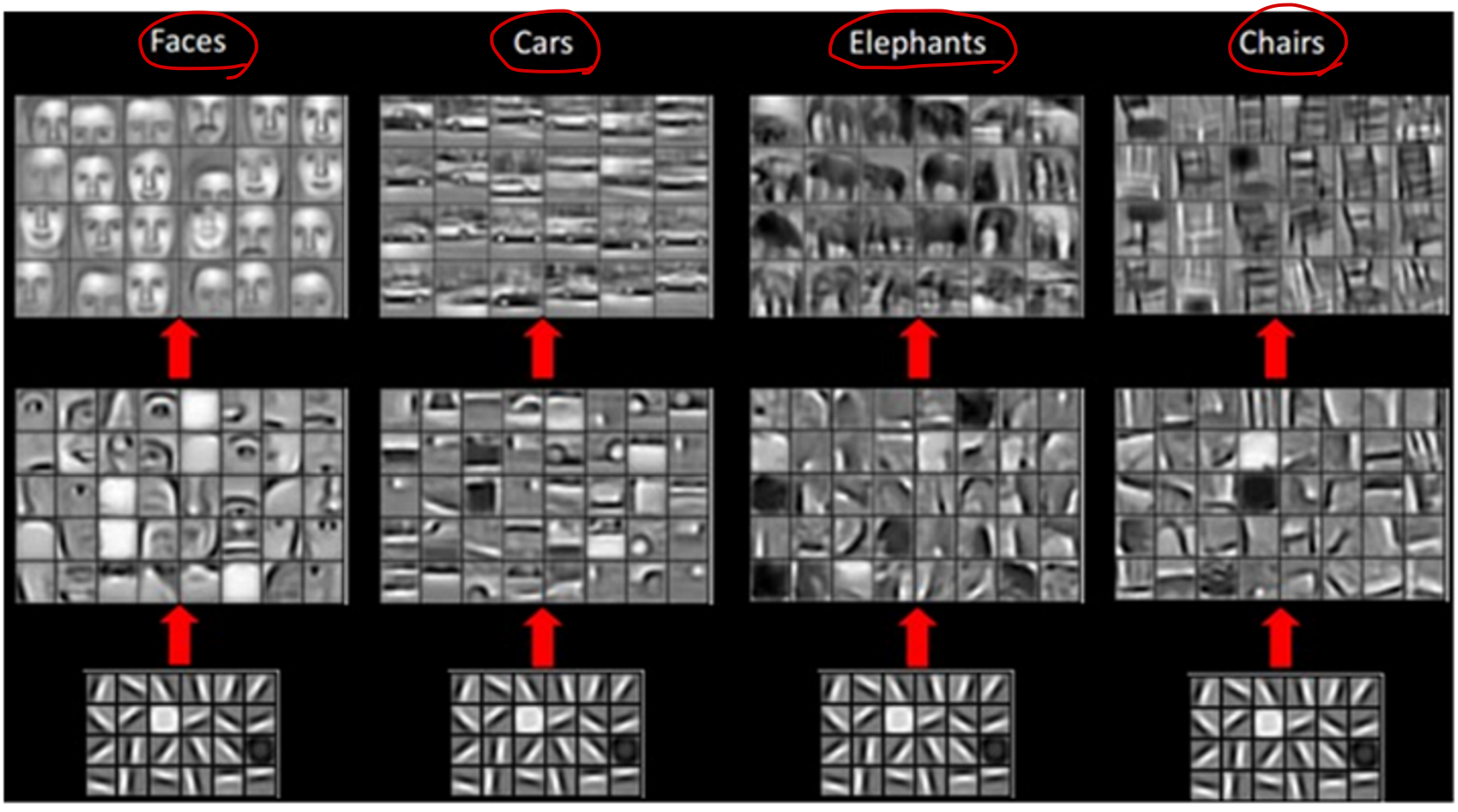
CNN: hierarchical learning of features

face recognition

conv. layer 3개의 예.



CNN: hierarchical learning of features



CNN: dimensional reduction by pooling layer

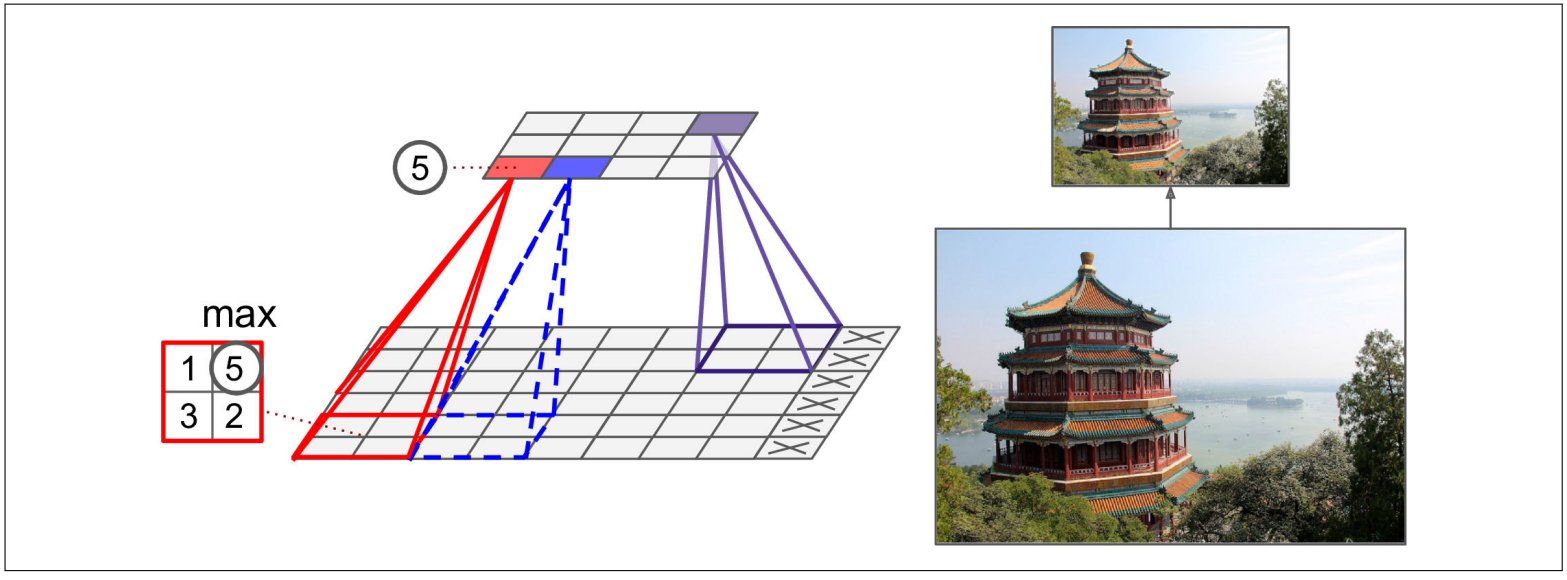


Figure 14-8. Max pooling layer (2 × 2 pooling kernel, stride 2, no padding)

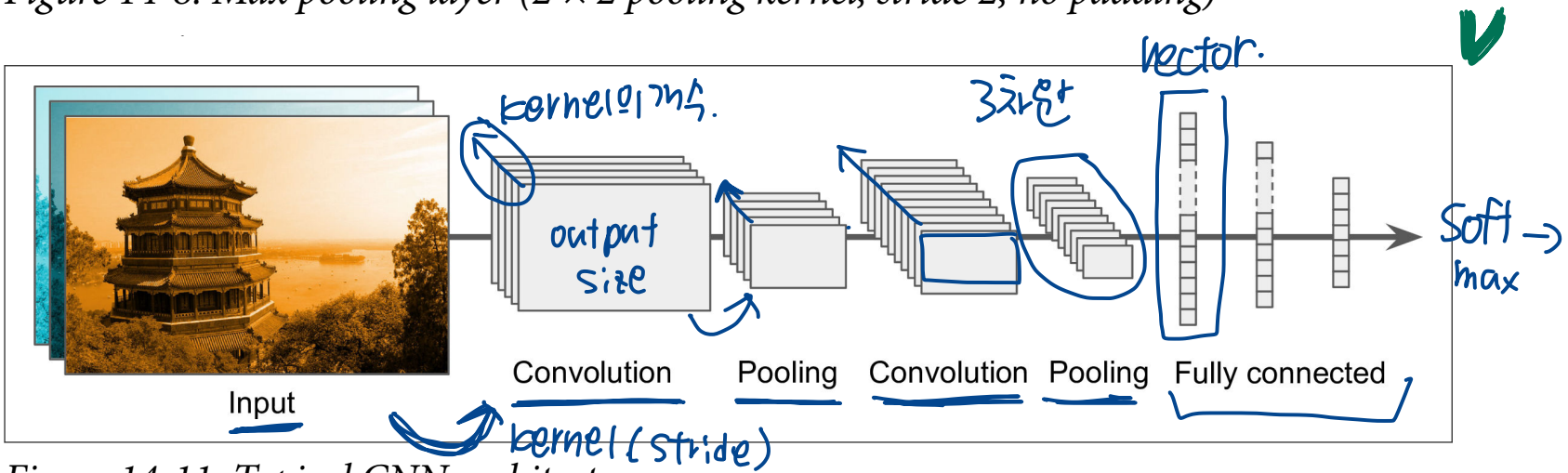


Figure 14-11. Typical CNN architecture

Appendix

Reference and further reading

- “Chap 14 | Deep Computer Vision Using Convolutional Neural Network” of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- “Lecture 10 | Convolutional Neural Networks” of Kwang Il Kim, Machine Learning (2019)