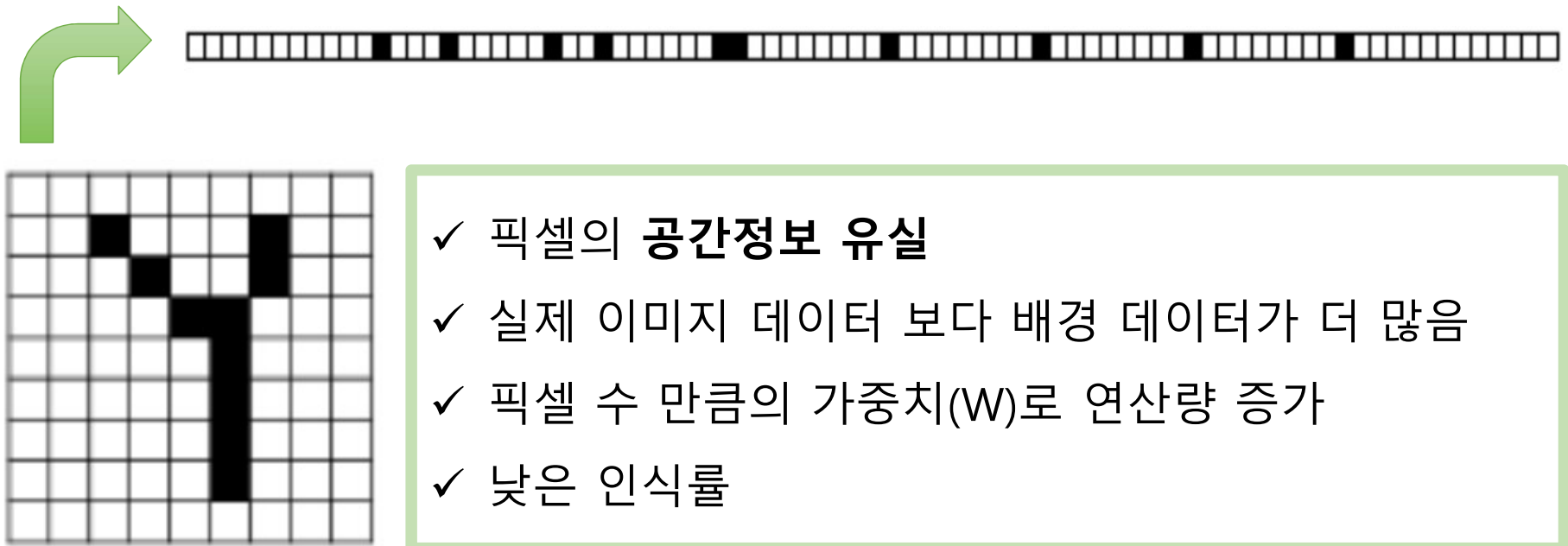


CNN

(CONVOLUTION NEURAL NETWORK)

CNN(Convolution Neural Network)

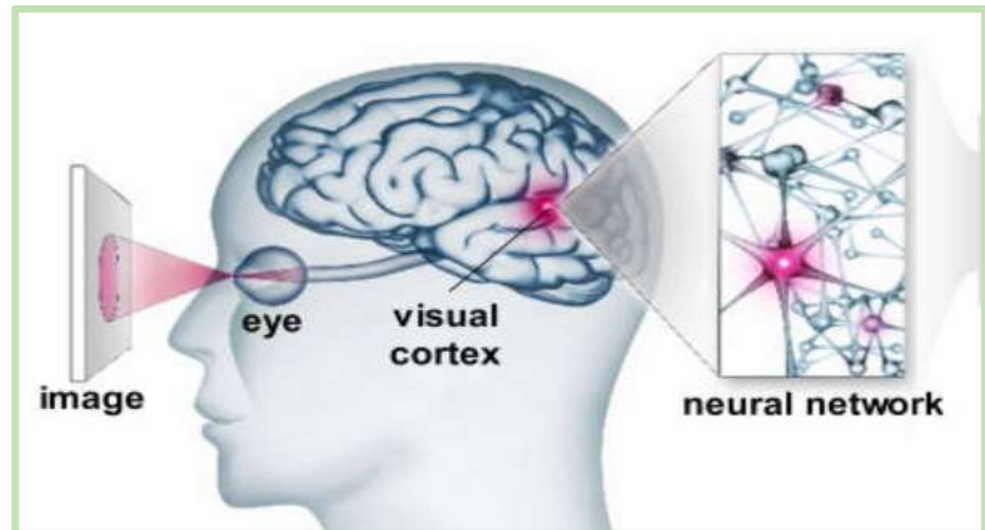
◆ DNN (MLP) 이미지 인식



CNN(Convolution Neural Network)

◆ 사람/동물 시각 인식

- ✓ 동물의 **시각피질**(visual cortex, 視覺皮質) 구조에서 영감
- ✓ 시각 자극이 1차 시각피질을 통해서 처리된 다음, 2차 시각피질 경유하여, 3차 시각피질 등 여러 영역 통과하며 모여진 정보가 계층적으로 처리되며 추상적인 **특징이 추출**되어 **시각 인식**



CNN(Convolution Neural Network)

◆ CNN 합성곱신경망

- DNN의 한 종류로 컴퓨터 **비전, 시각적 이미지 인식**에 주로 사용
- **텍스트 처리** 등 **여러 다른 분야에도 다양하게 활용**
- **LeNet-5은 1998년** Yann LeCun 교수가 발표한 CNN 알고리즘으로 지속적인 연구와 발전 진행, 특히 2010년 초중반에 많은 발전

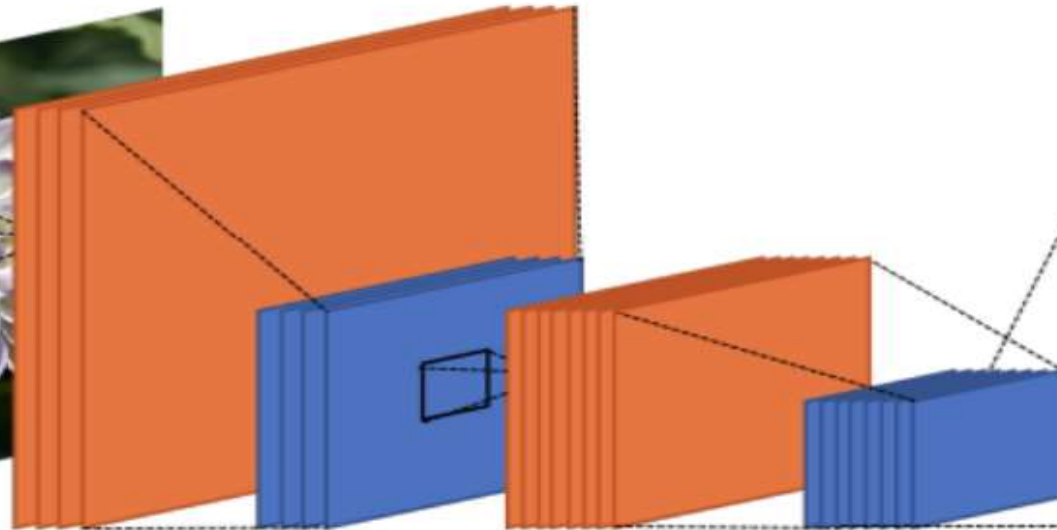
LeNet-5 -> Alexnet -> GoogLeNet -> VGGNet -> Resnet
(1998) (2012) (2014) (2014) (2015)

CNN(Convolution Neural Network)

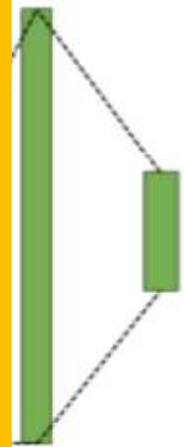
◆ CNN 합성곱신경망

- 입력 데이터의 형상을 유지하며 특징 추출 후 분류
 - 전반부 : 3차원 이미지 입력 받아 특징 추출
 - 후반부 : 특징 입력 받아 분류

전반부



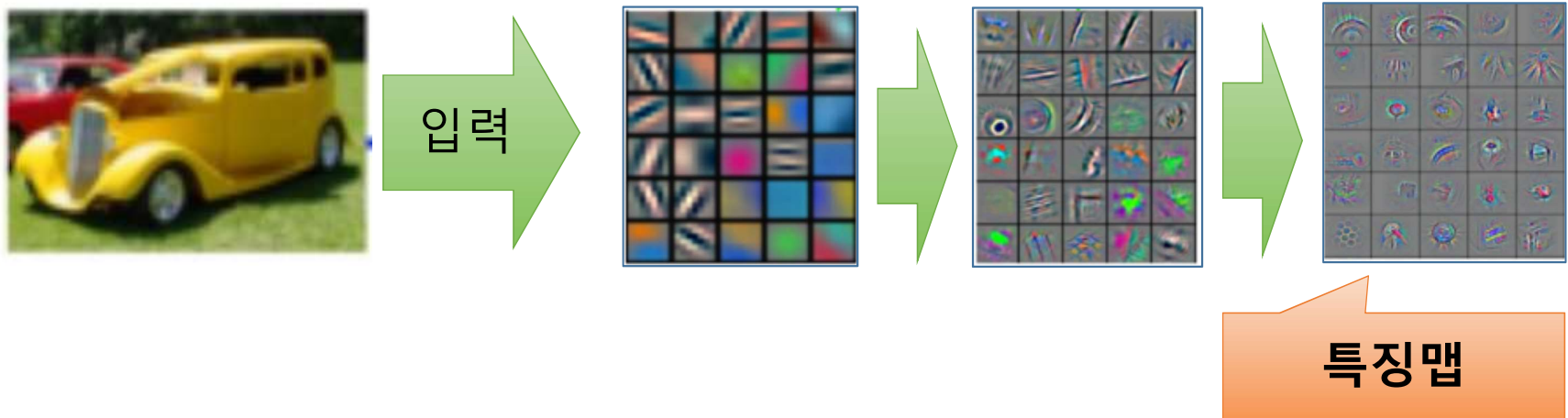
후반부



CNN(Convolution Neural Network)

◆ Convolution Layer

- 이미지 위를 일정 간격으로 이동하며 특징(정보)를 하나씩 추출
- 위에서 아래로 전체 이동으로 특징(정보)를 모은 특징맵을 출력하는 기능의 Layer



CNN(Convolution Neural Network)

◆ Convolution Layer

■ 커널/필터/마스크

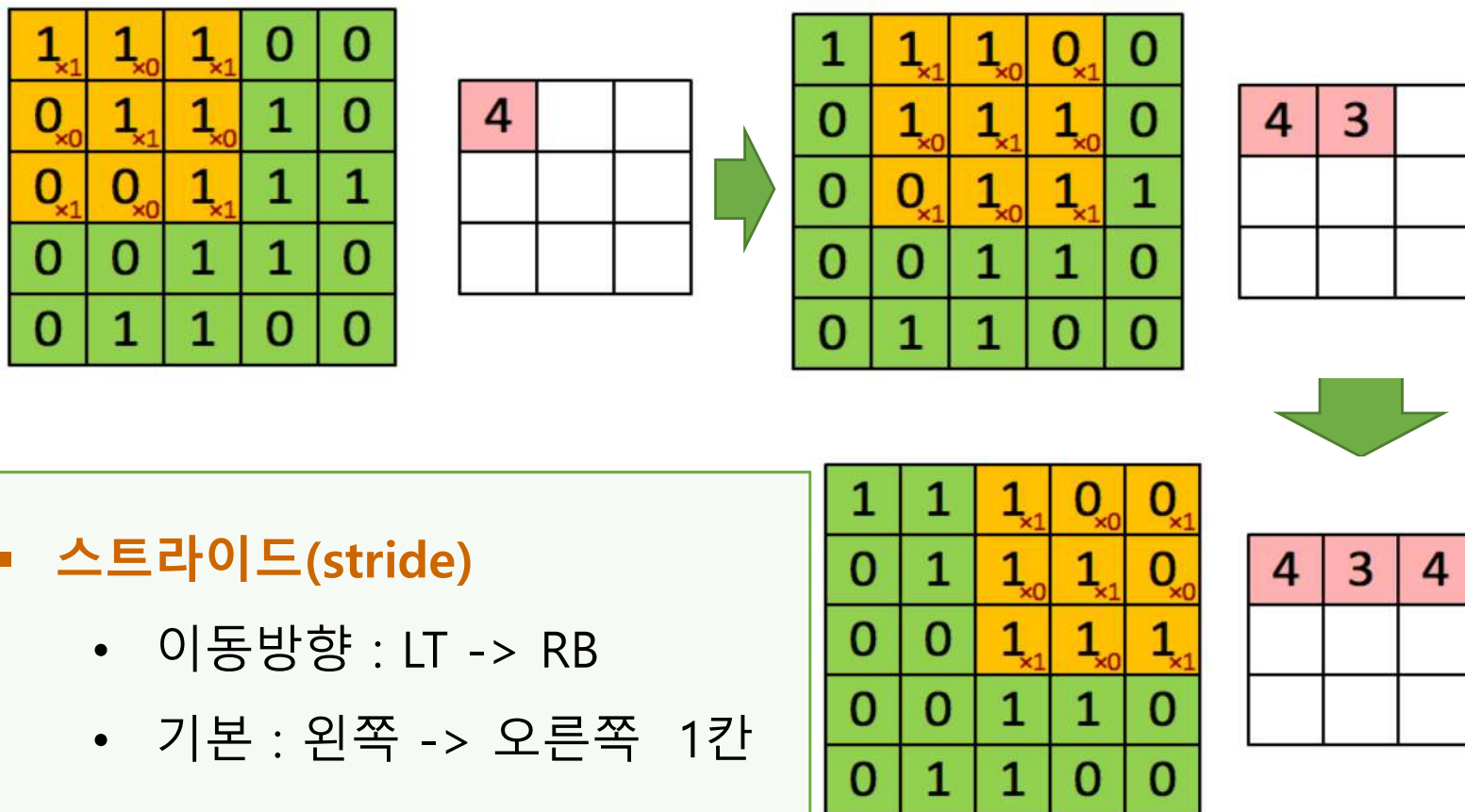
- 가중치로 구성되며 일반적으로 3x3, 5x5 크기
- 너무 큰 커널은 특징 추출 부족
- 이미지 위를 일정 간격 이동

■ 스트라이드(stride)

- 커널의 이동 방향 및 크기
- 기본 : 왼쪽 -> 오른쪽 1칸

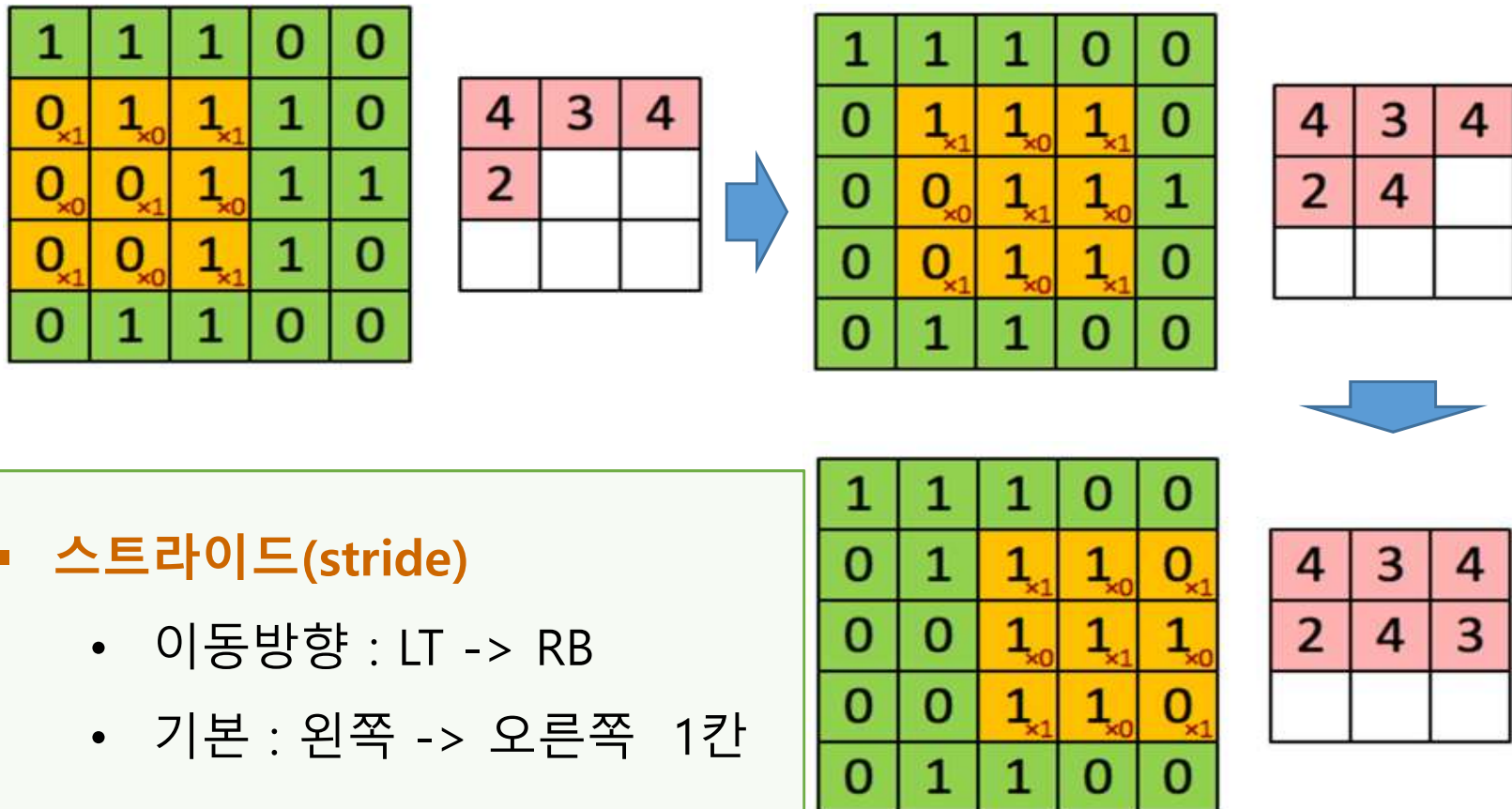
CNN(Convolution Neural Network)

◆ Convolution Layer



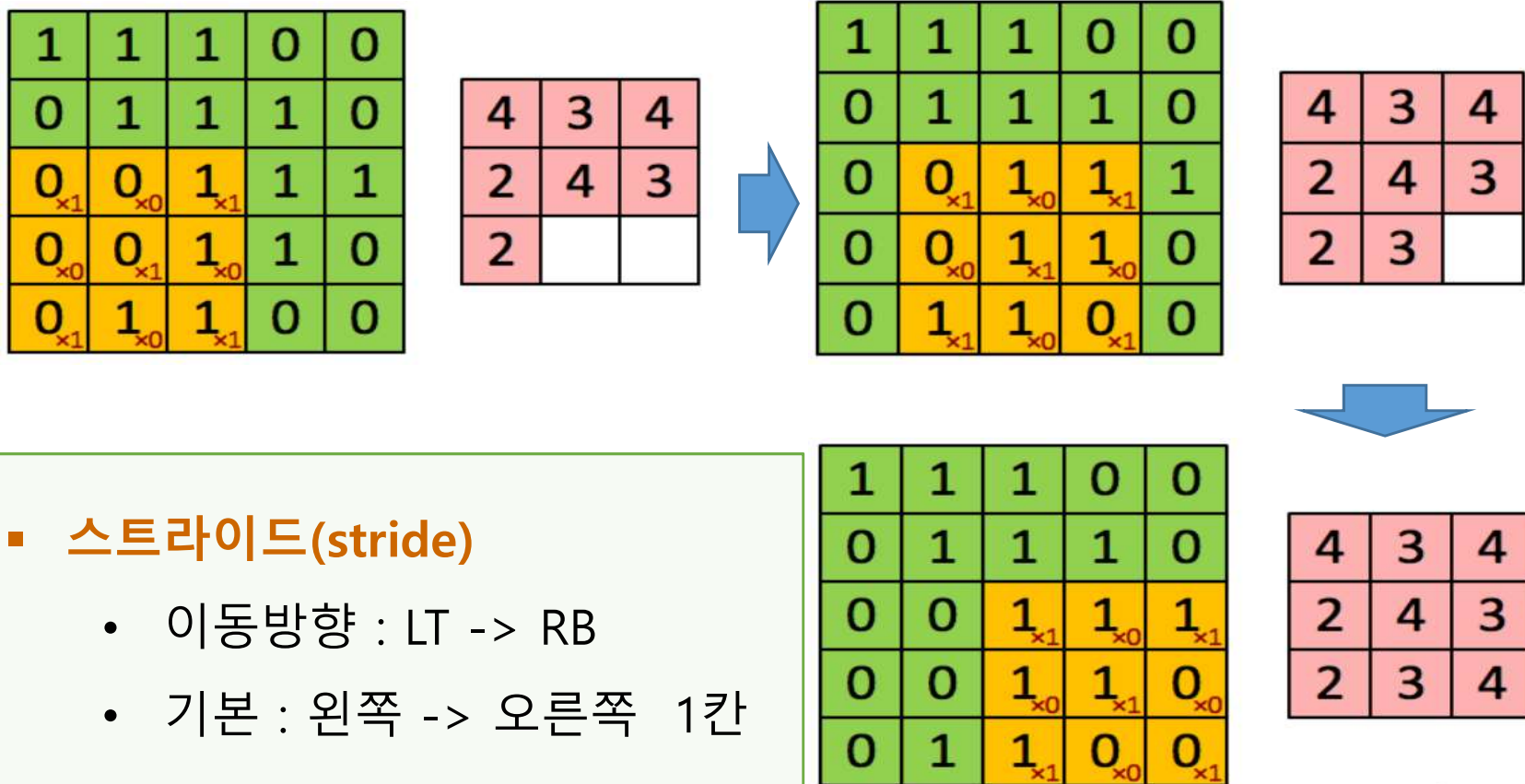
CNN(Convolution Neural Network)

◆ Convolution Layer



CNN(Convolution Neural Network)

◆ Convolution Layer



■ 스트라이드(stride)

- 이동방향 : LT -> RB
- 기본 : 왼쪽 -> 오른쪽 1칸

CNN(Convolution Neural Network)

◆ Convolution Layer

■ 패딩(Padding)

- 커널(필터) 이동 시 좌우상하 모서리 부분 특징 추출 안됨
- 입력 데이터 사면을 특정값(0)으로 채운 후 합성곱층 진행

➤ Valid Padding

- 입력 데이터와 출력 데이터 크기 다름

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

4	3	4
2	4	3
2	3	4

CNN(Convolution Neural Network)

◆ Convolution Layer

■ 패딩(Padding)

➤ Same Padding

- 입력 크기와 출력 크기 동일
- 입력 데이터 사면을 특정값(0)으로 채운 후 진행

x1	x0	x1
x0	x1	x0
x1	x0	x1

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	1	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0



Same-padding

2	2	3	1	1
1	4	3	4	1
2	2	5	3	3
1	2	3	4	1
1	2	3	1	1

stride = 1

◆ Convolution Layer

■ 패딩(Padding)

- 입력 이미지와 특징맵 크기

커널/필터

w	w	w
w	w	w
w	w	w

No Padding

[illegible]

Padding

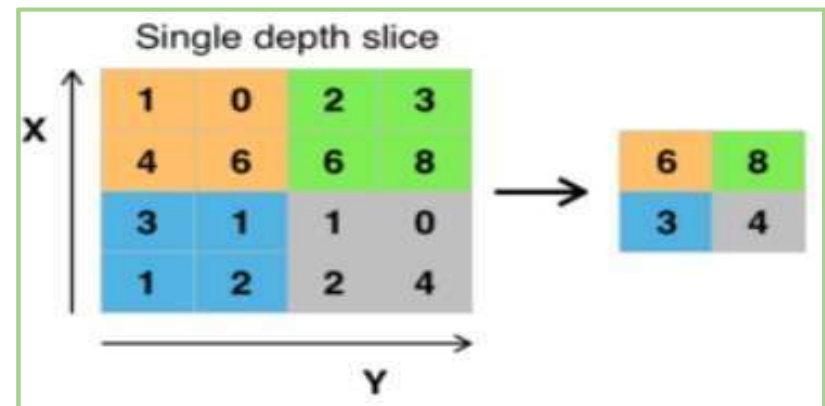
[illegible]

CNN(Convolution Neural Network)

◆ Polling Layer

- 합성곱 층(합성곱 연산 + 활성화 함수) 다음에 풀링 층 추가
- 특성 맵을 다운샘플링하여 **특성 맵의 크기 줄**이는 **풀링 연산** 진행
- 합성곱층과 달리 **커널이 중첩되지 않음**
- 커널 크기 : 2x2
- 종류 → 최대 풀링(max pooling), 평균 풀링(average pooling)

[최대 풀링 연산]

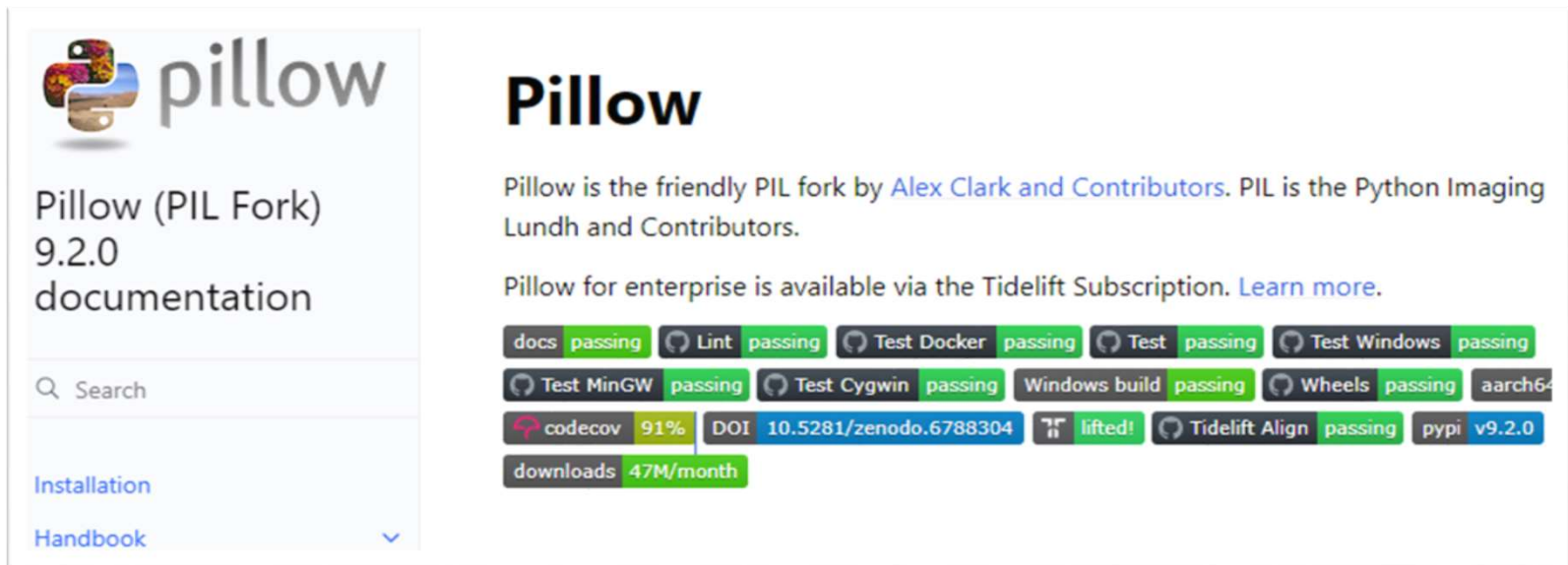


CNN(Convolution Neural Network)

◆ Image 처리

➤ Pillow 패키지 활용

- <https://pillow.readthedocs.io/en/stable/>



The screenshot shows the official Pillow documentation page. On the left, there's a sidebar with the Pillow logo, the text 'Pillow (PIL Fork) 9.2.0 documentation', a search bar, and links for 'Installation' and 'Handbook'. The main content area has the title 'Pillow' and a description: 'Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Lundh and Contributors.' Below this, it mentions 'Pillow for enterprise is available via the Tidelift Subscription. Learn more.' A large section of the page is dedicated to build status badges, showing various tests passing (docs, lint, test docker, test, test windows, test mingw, test cygwin, windows build, wheels, aarch64) and other metrics like codecov (91%), DOI (10.5281/zenodo.6788304), lifted!, Tidelift Align (passing), pypi v9.2.0, and downloads (47M/month).

Pillow

Pillow is the friendly PIL fork by [Alex Clark and Contributors](#). PIL is the Python Imaging Lundh and Contributors.

Pillow for enterprise is available via the Tidelift Subscription. [Learn more](#).

docs **passing** Lint **passing** Test Docker **passing** Test **passing** Test Windows **passing**
Test MinGW **passing** Test Cygwin **passing** Windows build **passing** Wheels **passing** aarch64
codecov **91%** DOI **10.5281/zenodo.6788304** lifted! Tidelift Align **passing** pypi **v9.2.0**
downloads **47M/month**

CNN(Convolution Neural Network)

◆ Image 처리

➤ Pillow 패키지 활용

```
!pip install pillow
```

```
import numpy as np  
from PIL import Image
```

```
FILE = './new_data_2.png'
```

```
with Image.open(FILE) as img:  
    img_resize = img.resize((32, 32))  
    img_resize.save('./new_data_2_32.png')
```

```
x = np.array(img_resize)  
print(x.shape)
```

```
(32, 32, 3)
```