

• 04/13

* CNN (Convolution Neural Network)

* "Computer Vision"
↓ computer가 (이미지) 보고 사람 혹은 여러 객체를 **식별하고**
구역을 찾도록

자연학은 computer science
분야!

* Deep Learning
(DNN) → [Vision → **物体 찾고리즘 (CNN)**
NLP (Natural Language Process) → (RNN)
자연어 처리 (LSTM)]

Computer Vision의 목적

↳ pixel을 이해하는 방법을 (**고안해요!**)
알아내요!)

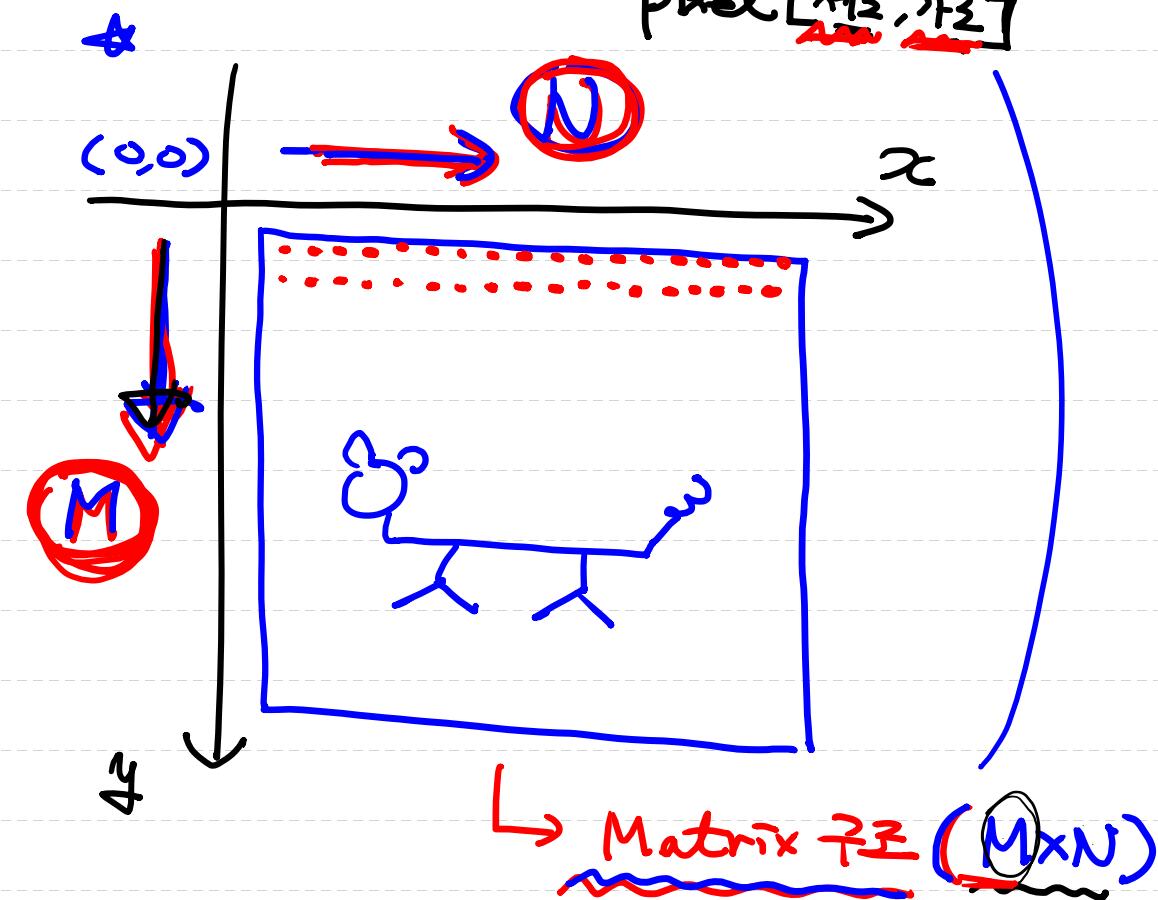
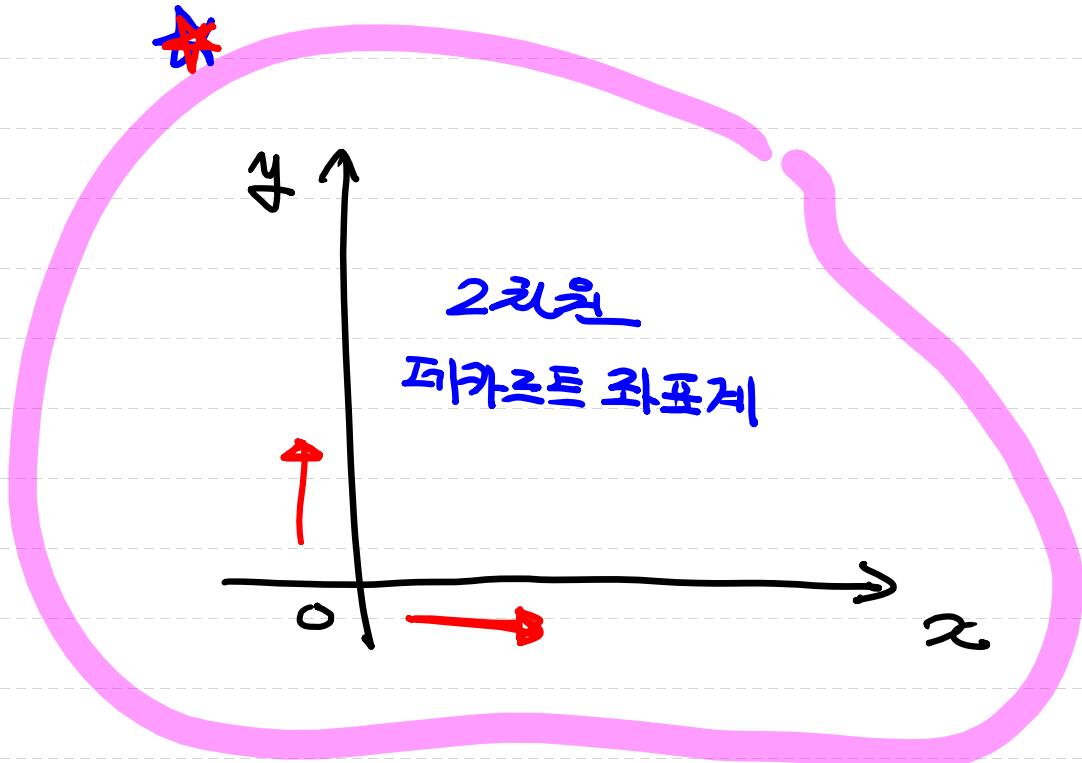


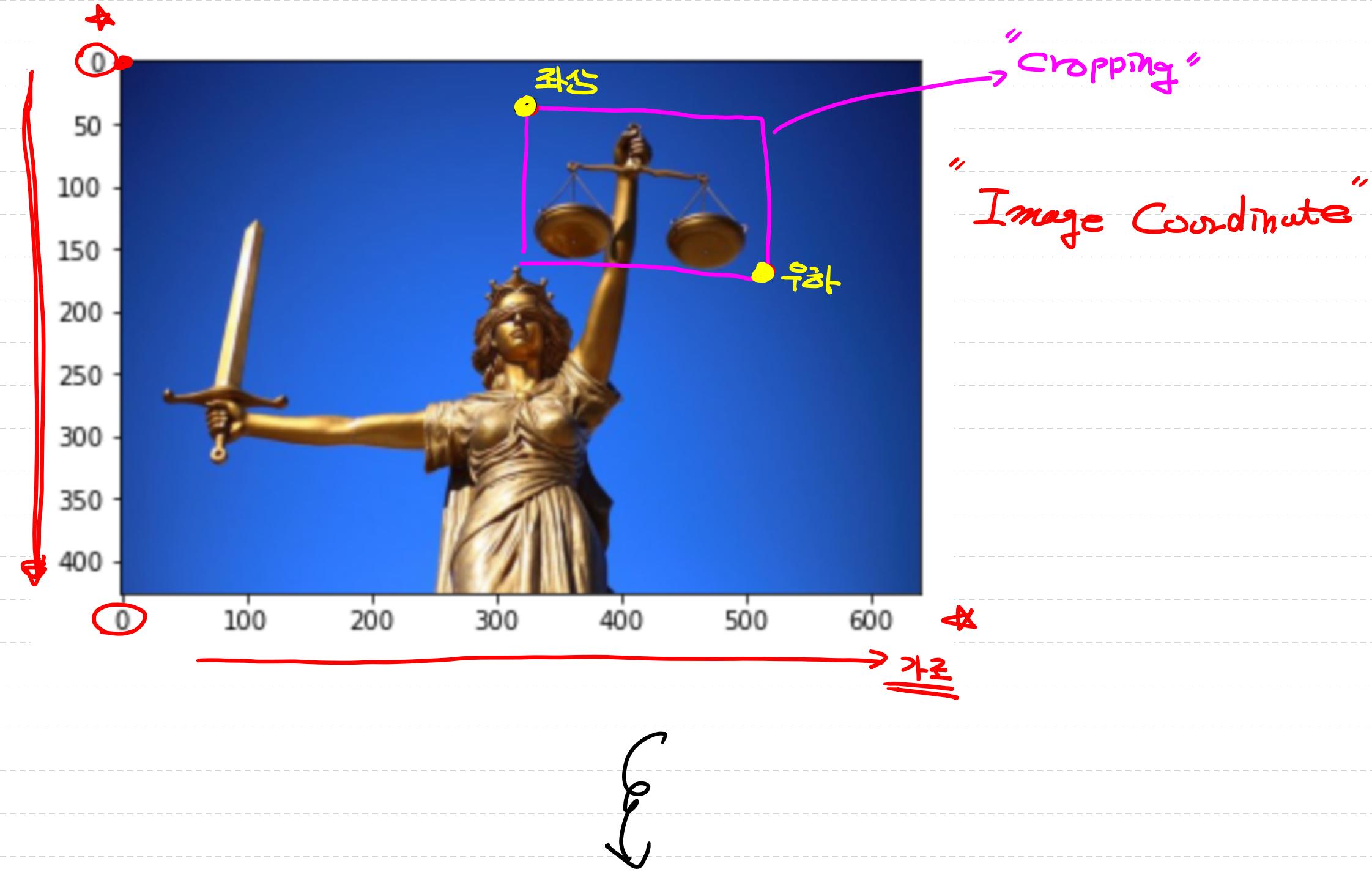
- 이미지를 이루는 가장 작은 단위 \rightarrow pixel

모니터, 랜드폰 해상도 (1024×768) \rightarrow 가로 pixel 1024
 (1600×900) 세로 pixel 768
 (2048×1024)

* 이미지 좌표계 (Image coordinate)

그림의 이미지를
2차원 ndarray로 표현
Pixel [세로, 가로]





① Digital Image의 종류

① 이진 이미지 (binary image) → 가장 간단한 형태

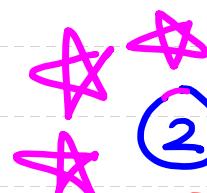
각 pixel의 값이 0 or 1
(0=검은색) (1=흰색)

앗!! 그러면 이미지를 표현할 때 이미지 1bit 가 쓰겠네요??

각 pixel당 1 bit만 있으면 되니까

구현하는데 8 bits로 표현해도 되어 좋네요

실제로는 아니예요 ㅎ



② 흑백 이미지 (gray-scaled image)

* (3차원)
* (2차원)

↳ 각 pixel의 값을 0 ~ 255 사이의 값으로 표현한 이미지
↳ 8 bits (1 byte)가 필요

→ Deep Learning 할 때 color image를 gray-scaled image로
변환하는 처리.

* 3차원

③ 컬러 이미지 (color image)

↳ 각 pixel의 값을

R (0~255) 8bit
G (0~255) 8bit
B (0~255) 8bit

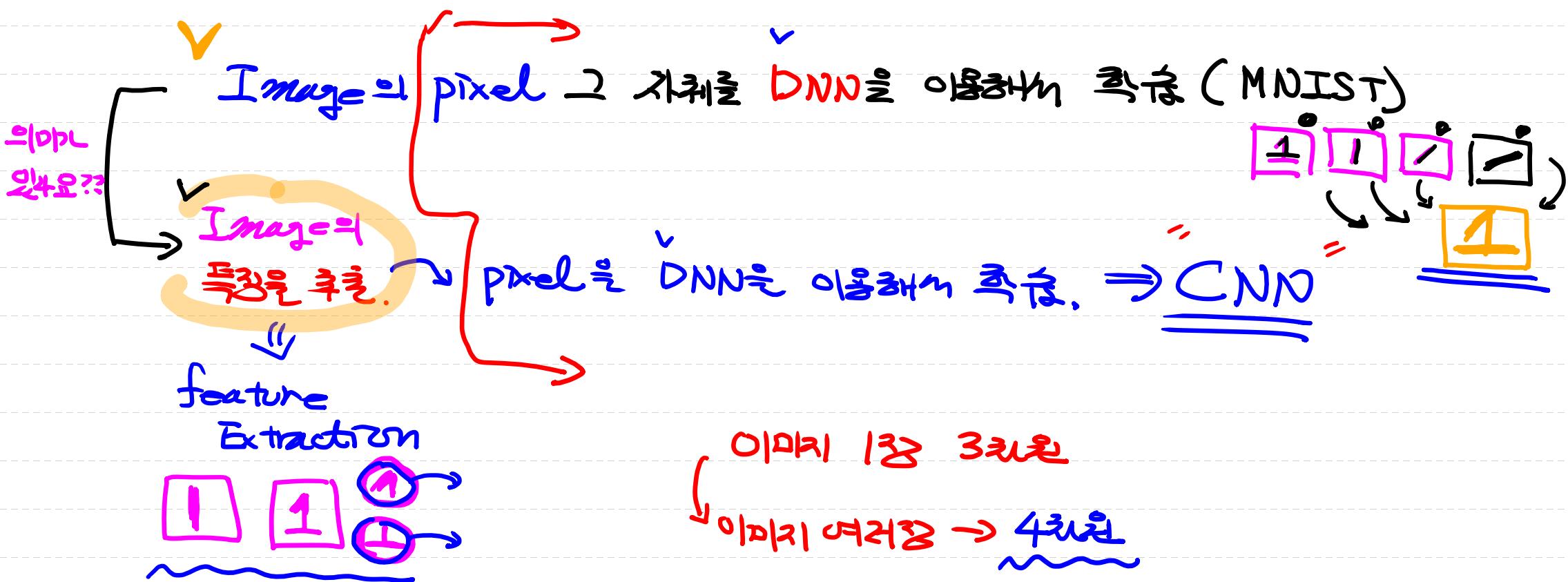
24bit

True
color

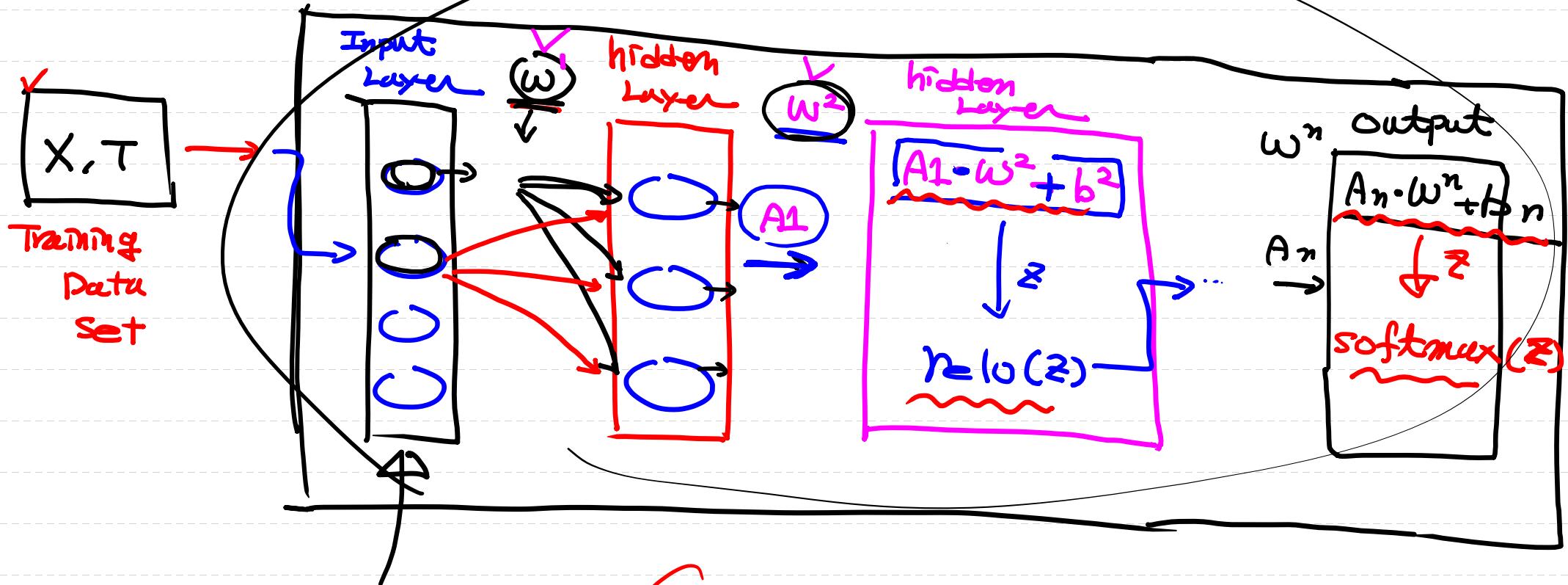
→ * CNN (Convolutional Neural Network)

↙ convnet 이란지도 불러요! (한국어 신경망)

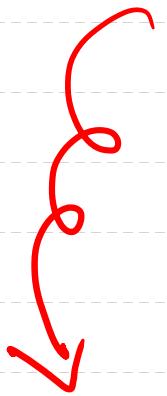
* DNN (Deep Neural Network) 우리가 알고 있는 일반적인 Deep Learning 구조
FC Layer (Fully Connected Layer): Dense Layer



DNN (Deep Neural Network)



FLATTEN

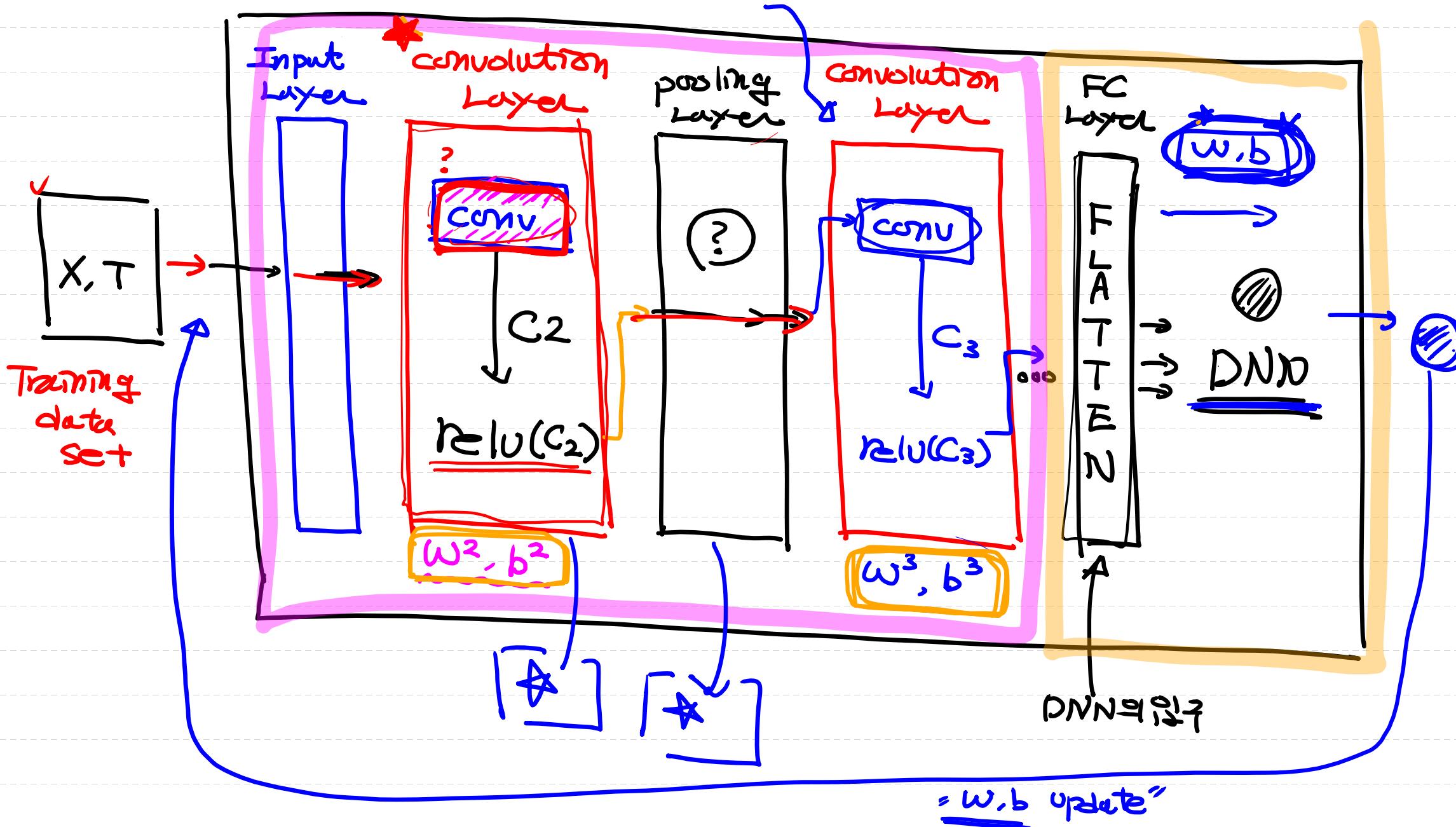


CNN으로 표현해 보아요 ~~

① CNN (convolutional Neural Network)

$$\text{Filter } \underset{20}{\approx} \xrightarrow{\star} (x, x, 20)$$

$(10, 10, 3)$



① Convolution (컨볼루션)

→ 주제적인 퀴즈는 어려워요!



컨볼루션은 두 흡수 f, g 가 있을 때

하나의 흡수를 뒤집(reverse), 그다음 shift 시키면

두 흡수를 곱한 후 적용하는 연산

그림을 통해 이해

IMAGE DATA

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



FILTER

1	0	1
0	1	0
1	0	1

(3x3)

필드는 여러개를 사용

이미지의 특징을 다양하게 추출

같은 위치에 있는 것끼리 곱한 다음 합

$$1 + 0 + 1 + 0 + 1 + 0 + 0 + 0 + 1 = 4$$

Stride → scalar(filter)가 이동하는 거리) → 1

"온라인 3x3"

"Stride = 2" 사용할 3 줄이요!!

image의 크기, Filter의 크기야

파라미터 stride는 제작을

별이요.

일반적으로 1을 많이 사용

IMAGE DATA

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

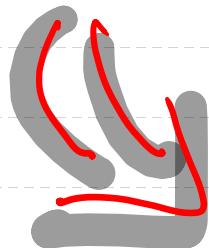
FILTER

1	0	1
0	1	0
1	0	1

결과

Activation Map

if stride = 1 이라면



4	3	4
..
..

3x3

결과

Feature

MAP

결과

② 3 channel인

Image DATA

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

H Stride = 1

Channel

R

G

B

Filter

1	0	1
1	1	1
0	0	1

1	0	1
1	1	1
0	0	1

1	0	1
1	1	1
0	0	1

각 channel

v	v
v	v

+

v	v
v	v

+

v	v
v	v

- 입력 이미지의 channel 수는 1개일 때 Filter 1개당
(이미지)
1개의 Feature Map 생성

⇒ filter가 여러개면

여러개의 Feature Map이 생성 (3개일 때) → 여러개의 생성

↳ channel

“결과”

Convolution Layer의 결과는

Activation Map 이 되요.



Filter의 Stride에 의해 Feature Map의 크기는 작아져요!

이제 이미지의 차원을 유지하기 위해 패딩을 넣어야

→ “Padding” (zero padding)

