

# Convolutional Neural Network

박석희

부산대학교 기계공학부

\*E-mail: selome815@pusan.ac.kr



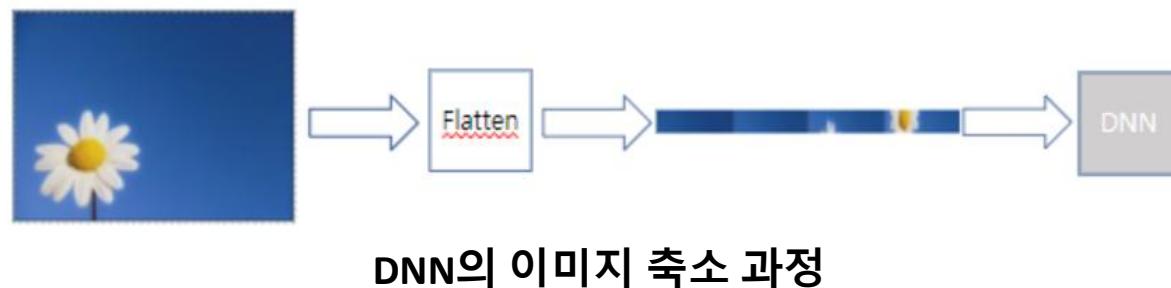
PUSAN  
NATIONAL UNIVERSITY

2015 NATIONAL UNIVERSITY  
SCHOOL OF MECHANICAL ENGINEERING  
**School of  
Mechanical Engineering**

# CNN (Convolutional Neural Network)

## ❖ CNN의 도입

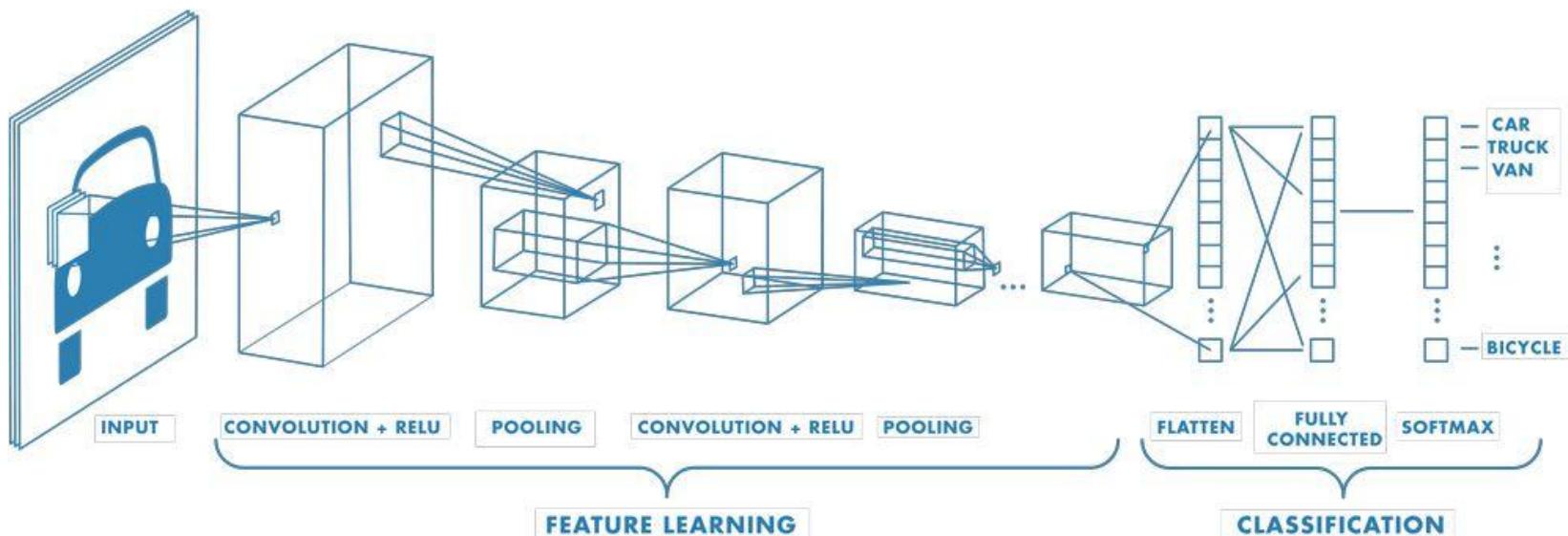
- 일반 DNN은 1차원 형태의 데이터를 사용하기 때문에, 2차원의 이미지 데이터를 1차원의 데이터로 변환하는 과정에서 공간 정보가 손실됨
- CNN은 이러한 문제를 해결하기 위해 공간적 정보를 유지하면서 데이터를 축소하여 neural network에서 연산을 수행함



# CNN (Convolutional Neural Network)

## ❖ CNN의 구조

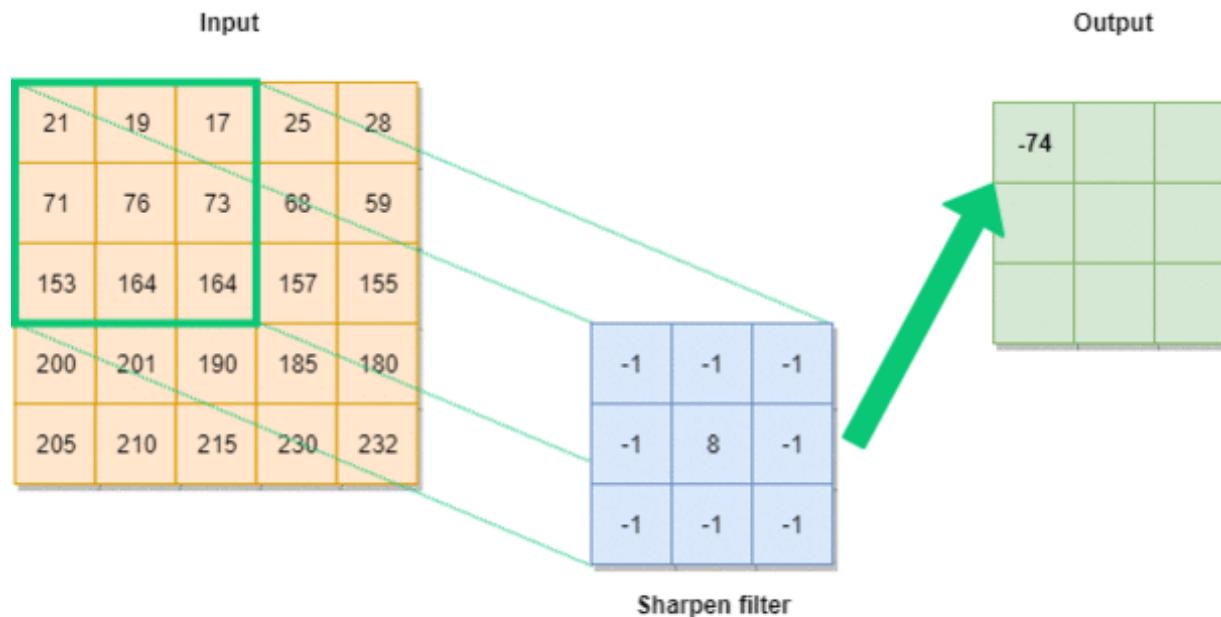
- Convolution layer : 입력 데이터에 필터를 적용해 특징을 추출
- Pooling layer : 다운 샘플링을 통해 파라미터의 개수를 줄여 출력 단순화
- Convolution 및 pooling이 완료된 후 fully connected layer 및 softmax 함수를 이용해 분류 수행



# CNN (Convolutional Neural Network)

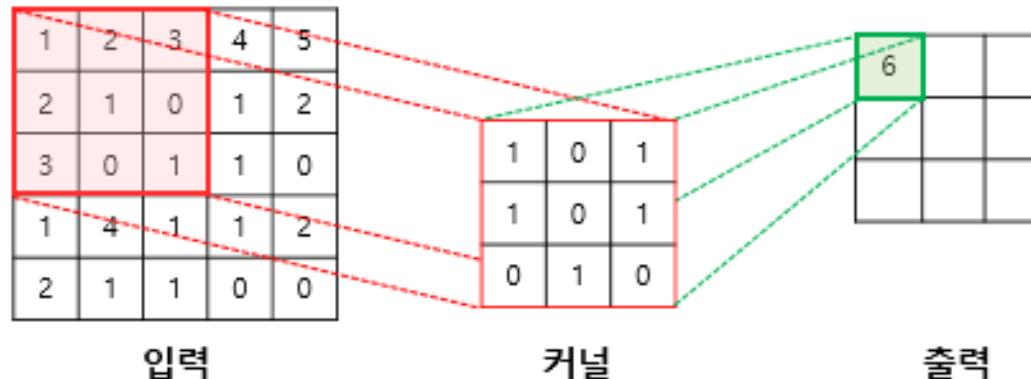
## ❖ Convolution layer

- 2D 형태의 필터를 이용하여 이미지 전체를 이동하면서 합성곱 연산을 수행
- 이를 이용하여 이미지 데이터의 공간적 상관관계를 추출

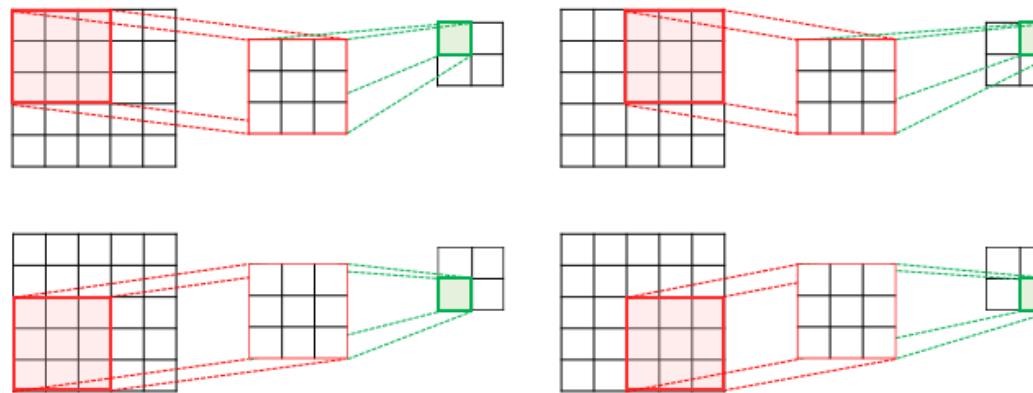


### ❖ Convolution layer 관련 개념

- Filter : Kernel이라고도 하며 입력 데이터에 곱해서 feature map을 추출



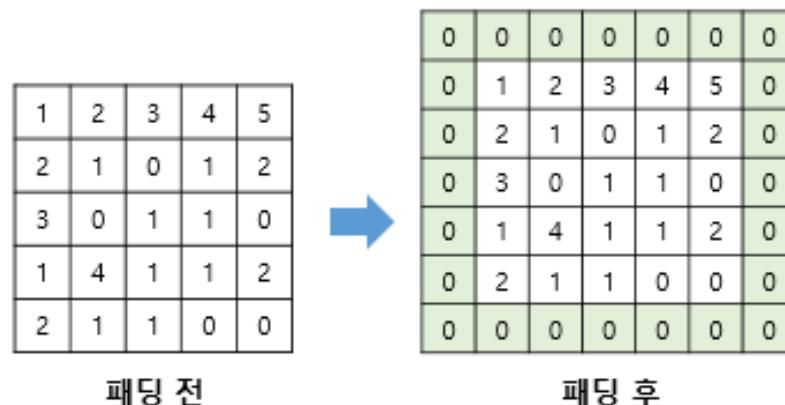
- Stride : 필터가 이동하는 간격의 단위



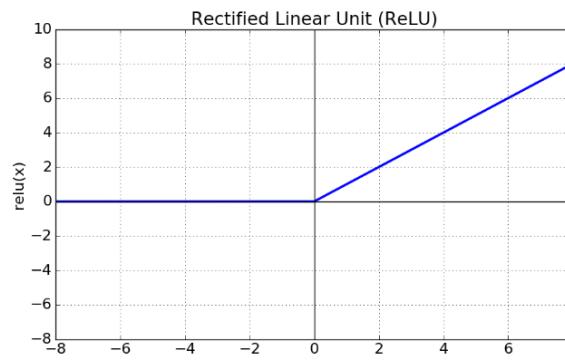
# CNN (Convolutional Neural Network)

## ❖ Convolution layer 관련 개념

- Padding : 합성곱 후 출력된 feature map의 크기를 입력데이터와 같은 크기로 만들어주는 작업  
패딩을 통해 과도한 정보의 축소를 막을 수 있음. 주로 0으로 채우는 zero padding 사용



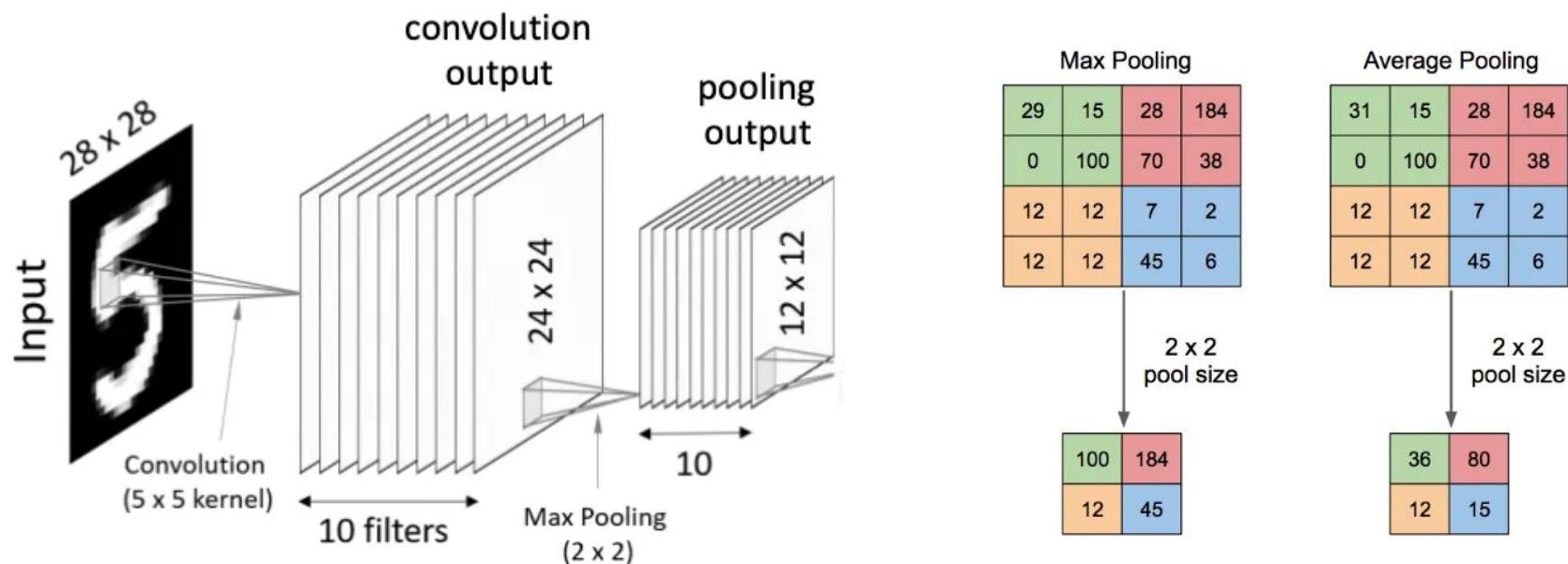
- Activation function : 선형 함수인 convolution에 비선형성을 추가하기 위해 사용  
주로 ReLU를 사용하여 음수 값은 0으로 매픅하여 training을 향상시킴



# CNN (Convolutional Neural Network)

## ❖ Pooling layer

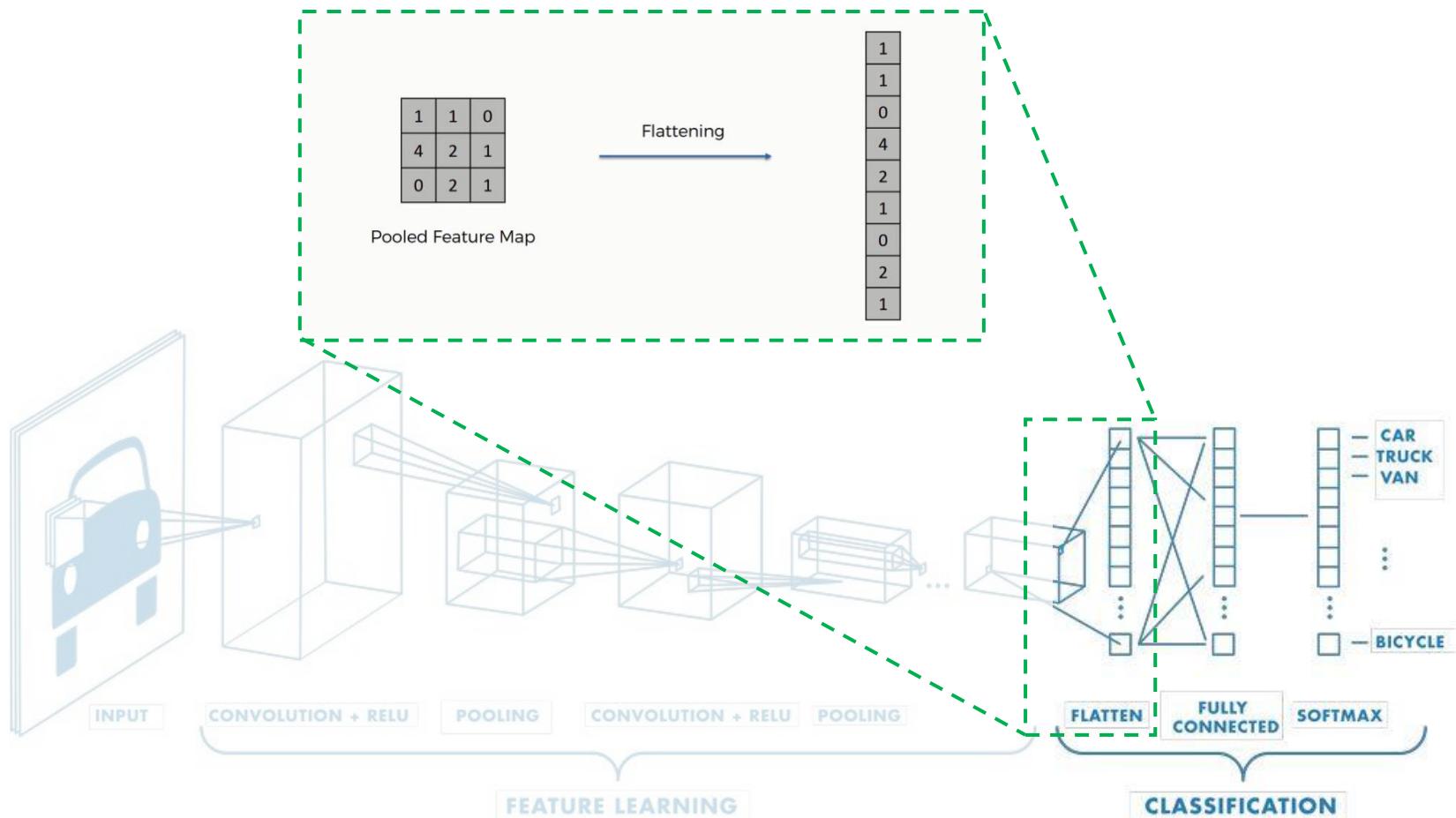
- Pooling : convolution 과정을 통해 생성된 feature map에서 특정 부분만 추출해 차원을 축소 필요한 부분 추출하고, parameter를 줄여 과적합을 방지하고 연산 속도를 향상시킴
- 구역에서 최대값을 추출하는 max pooling과, 평균값을 추출하는 average pooling이 대표적



# CNN (Convolutional Neural Network)

## ❖ Flattening

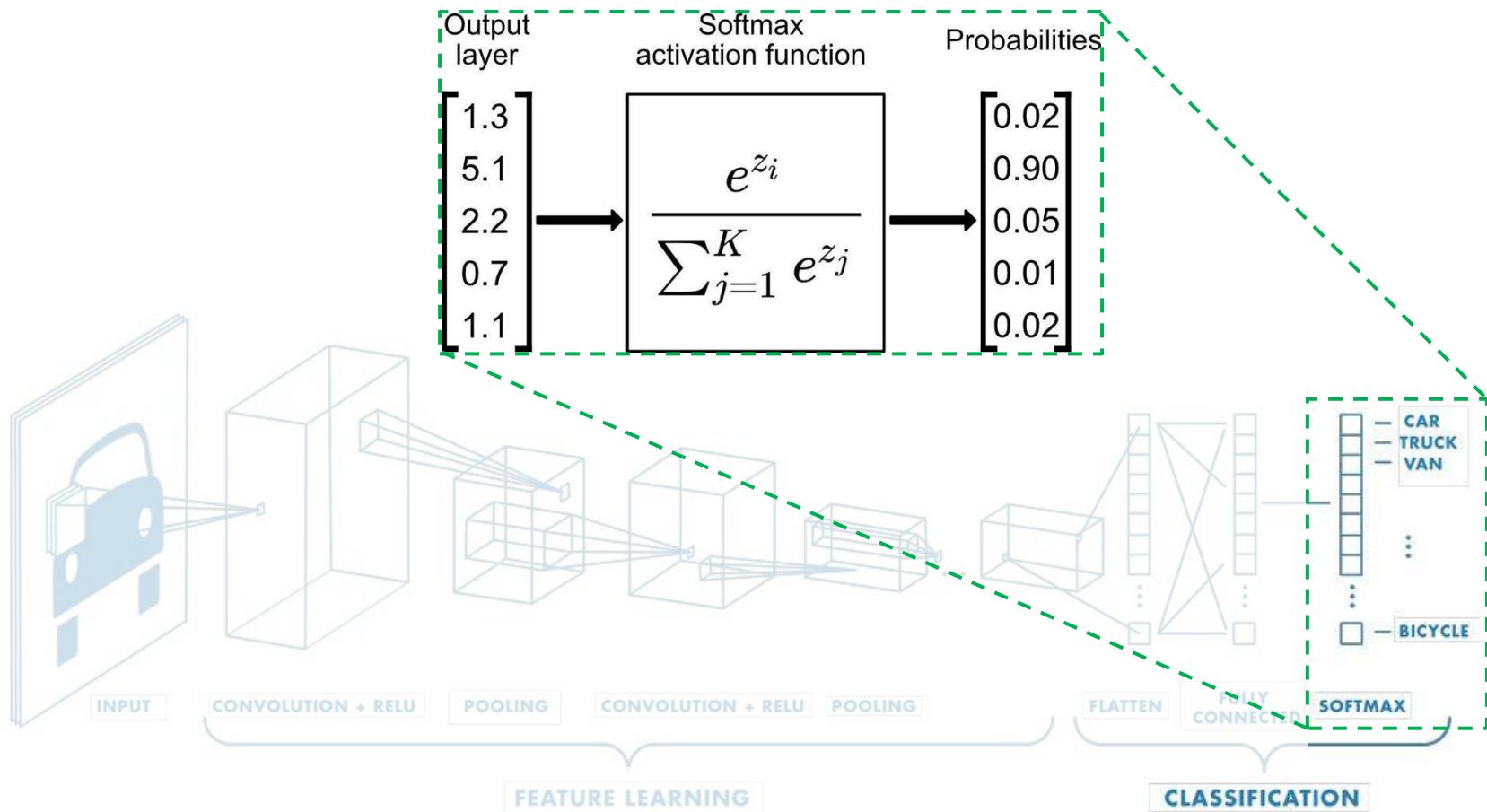
- Convolution 및 pooling layer를 거쳐 출력된 feature map을 fully connected layer에 연결하기 위해 flattening을 통해 1차원의 데이터로 축소 시킴



# CNN (Convolutional Neural Network)

## ❖ 결과 추출

- Fully connected layer를 적용시킨 뒤 최종적으로 softmax activation function을 적용하여 가장 확률이 높은 class를 추출함



## ❖ 1-D CNN

- CNN은 주로 사용되는 이미지 분류 뿐만 아니라 시계열 및 텍스트 분석에도 사용
- DNN도 수치 데이터에 이용할 수 있지만 고정된 입력 크기가 필요하고, 데이터의 로컬 구조를 고려하지 않음
- 1D CNN은 가변 길이 입력을 처리하고 로컬 구조를 고려할 수 있기 때문에 시계열 또는 오디오 신호 등의 데이터 처리에 적합함

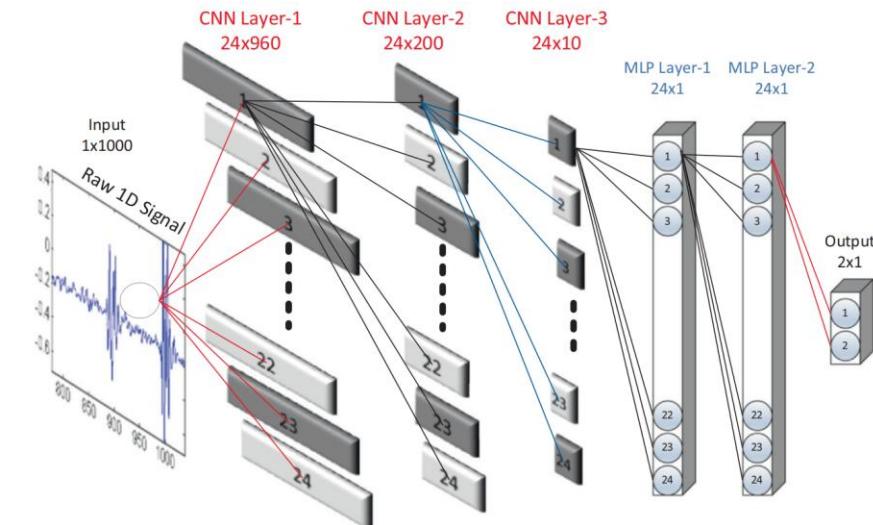
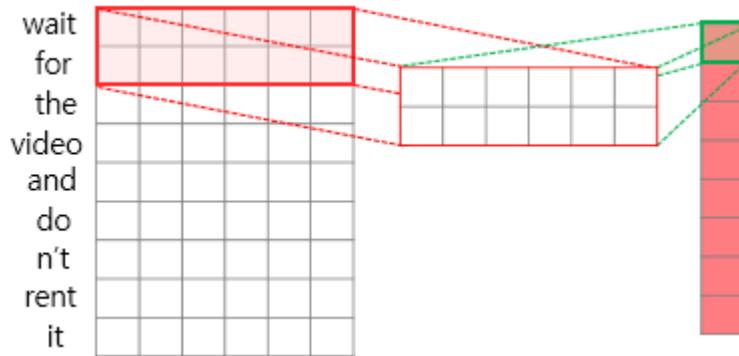


Fig. 5. A sample 1D CNN configuration with 3 CNN and 2 MLP layers.

## ❖ Object detection

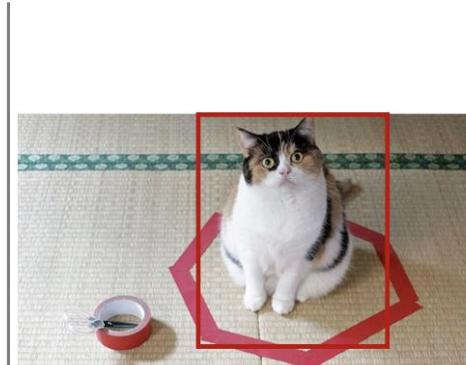
- CNN 기반으로 이미지를 단순히 분류하는 것에서 이미지 내에서 해당 객체가 어디에 위치하는지 표현하는 localization으로 변화
- 또한 한 이미지에서 하나의 객체만을 탐지하던 모델에서 다양한 object를 탐지할 수 있는 object detection으로 변화

1



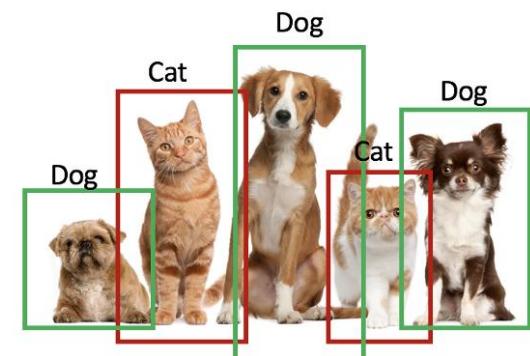
Is this image of Cat or not?

2



Where is Cat?

3



Which animals are there in image and where?

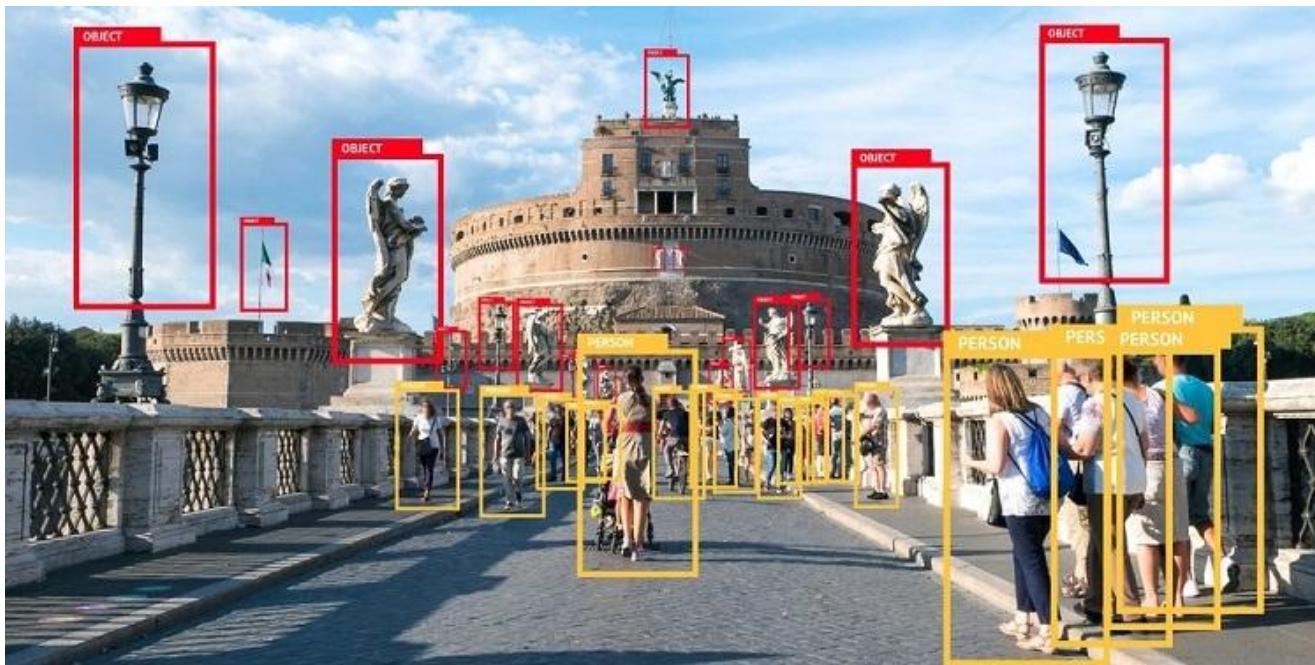
Image classification problem

Classification with localization problem

Object detection problem

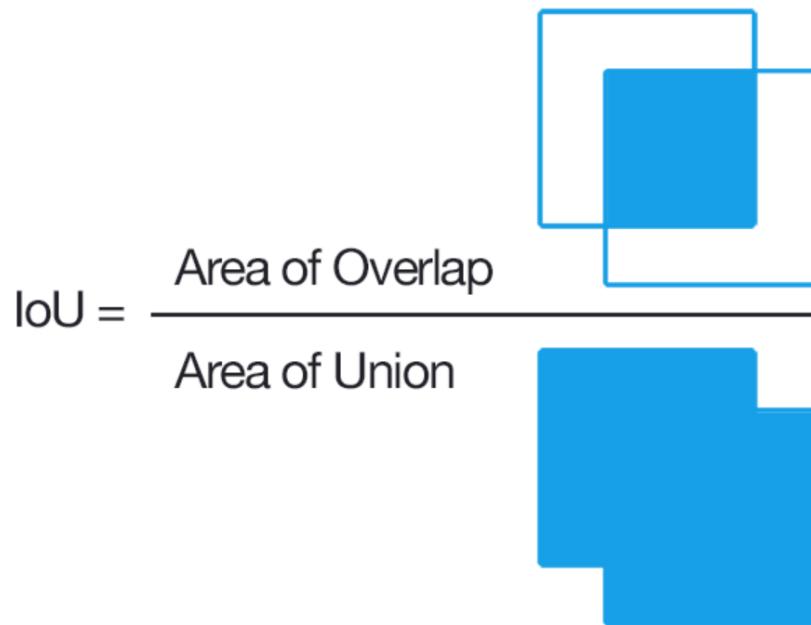
## ❖ Object detection의 특징

- Object detection은 Multipled label classification과 bounding box regression으로 구성
- 즉, 이미지 또는 영상에서 객체에 label을 부착하고 그 위치를 box 형태로 표시하는 과정으로 구성됨



## ❖ 성능 평가 지표

- IoU (Intersection over Union) : Object detection에서 정확도를 측정하기 위한 지표로, 탐지 결과와 labeling된 Ground truth와의 겹치는 면적의 비율을 나타내는 값  
Object detection 결과와 ground truth의 겹치는 면적을 두 영역을 합친 면적으로 나누어 준 값을 사용하며, 일반적으로 0.3 이하이면 poor, 0.7 이상이면 good으로 판단함



# Object detection

## ❖ 성능 평가 지표

- F1 Score : Precision과 Recall 성능을 동시에 고려하기 위해서 F1-Score를 사용  
precision과 recall의 조화평균으로, 1에 가까울수록 분류성능이 좋음

Precision : 모델이 positive로 판정한 것 중 실제로 positive인 샘플의 비율

positive로 검출된 결과가 얼마나 정확한지 판단하는 기준

Recall : 실제 positive 샘플 중 분류 모델이 positive로 판정한 비율

분류 모델이 실제 positive 클래스를 얼마나 빠지지 않고 잘 잡아내는지 판단

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$(Precision) = \frac{TP}{TP + FP}$$

$$(Recall) = \frac{TP}{TP + FN}$$

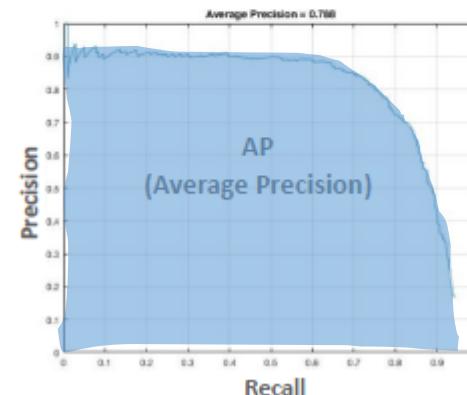
## ❖ 성능 평가 지표

- mAP (mean Average Precision) : 객체가 탐지된 재현율(Recall)의 변화에 따른 정밀도(Precision)의 값을 평균한 성능 수치  
P-R Curve의 면적(AUC)을 계산하여 얻어냄. 여러 개의 클래스를 검출하는 경우, 모든 클래스의 AUC를 계산한 뒤 평균하여 획득

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

$$(Precision) = \frac{TP}{TP + FP}$$

$$(Recall) = \frac{TP}{TP + FN}$$

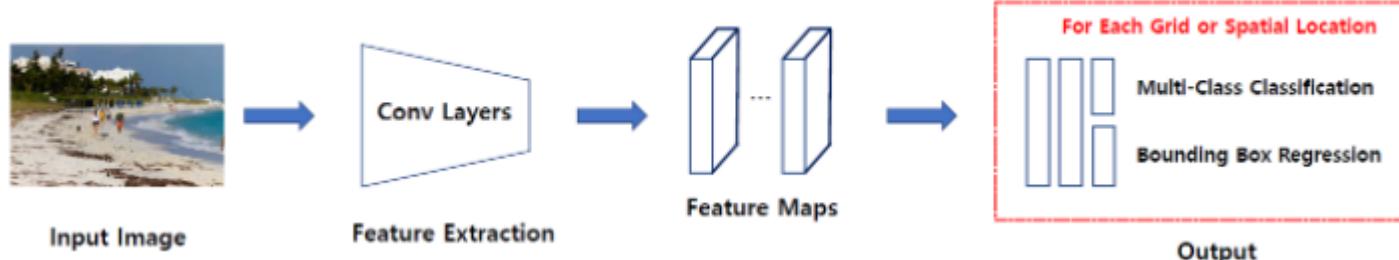


# Object detection

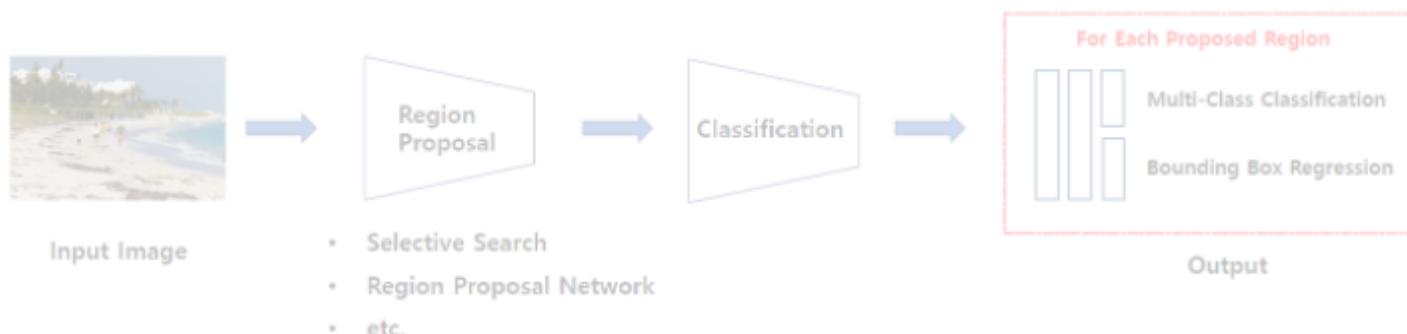
## ❖ Object detection의 구조 – 1-stage detector

- Object detection model은 구조에 따라 1-stage detector와 2-stage detector로 구분 가능
- 1-stage detector는 region proposal과 detection을 동시에 수행하여 빠른 속도를 가지나, 정확도가 2-stage detector 대비 낮음. 대표적인 모델로 YOLO, SSD 계열이 있음

### 1-stage detector



### 2-stage detector

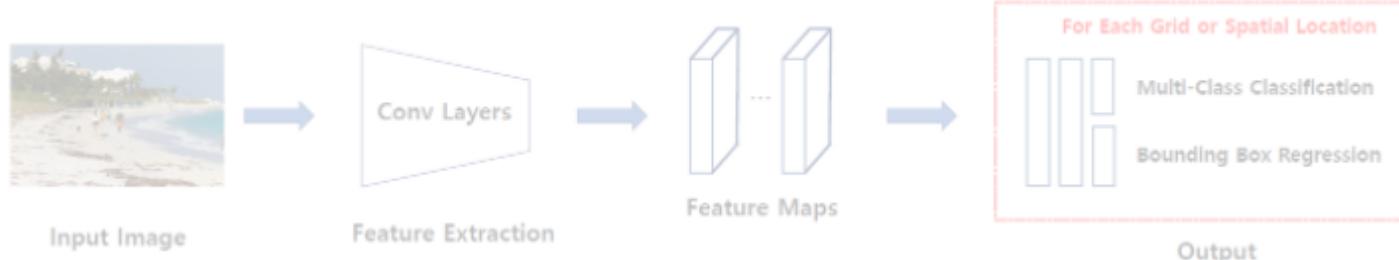


# Object detection

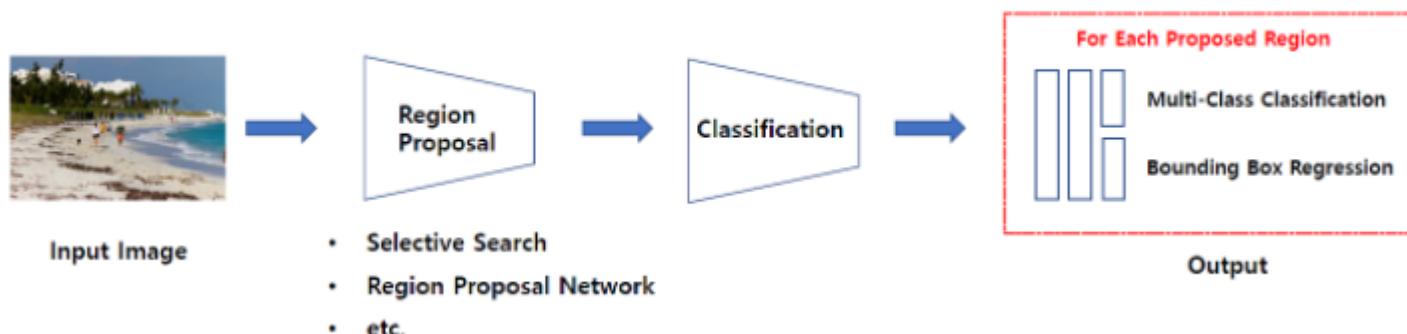
## ❖ Object detection의 구조 – 2-stage detector

- 2-stage detector의 경우 region proposal과 detection을 순차적으로 수행
- 연산량이 많아 1-stage detector 대비 속도가 느리지만 높은 정확도를 가짐.  
대표적인 모델로 R-CNN 계열이 있음

1-stage detector



2-stage detector

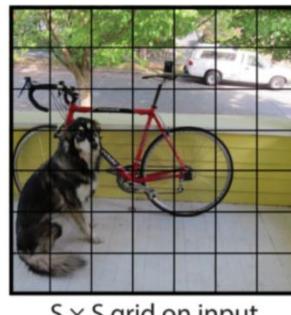


## ❖ 1-stage detector – YOLO (You Only Look Once)

- YOLO는 1-stage detector의 대표적인 모델로 빠른 객체 검출 속도로 실시간 객체 검출에 주로 사용됨.
- 입력된 사진을 grid 영역으로 나눠주고 각 그리드에 대해 미리 정의된 anchor box 기반 confidence score를 계산하여 높은 confidence score를 가진 위치를 선택하여 객체 카테고리를 파악함

(3)

(1)



Bounding boxes + confidence

Class probability map

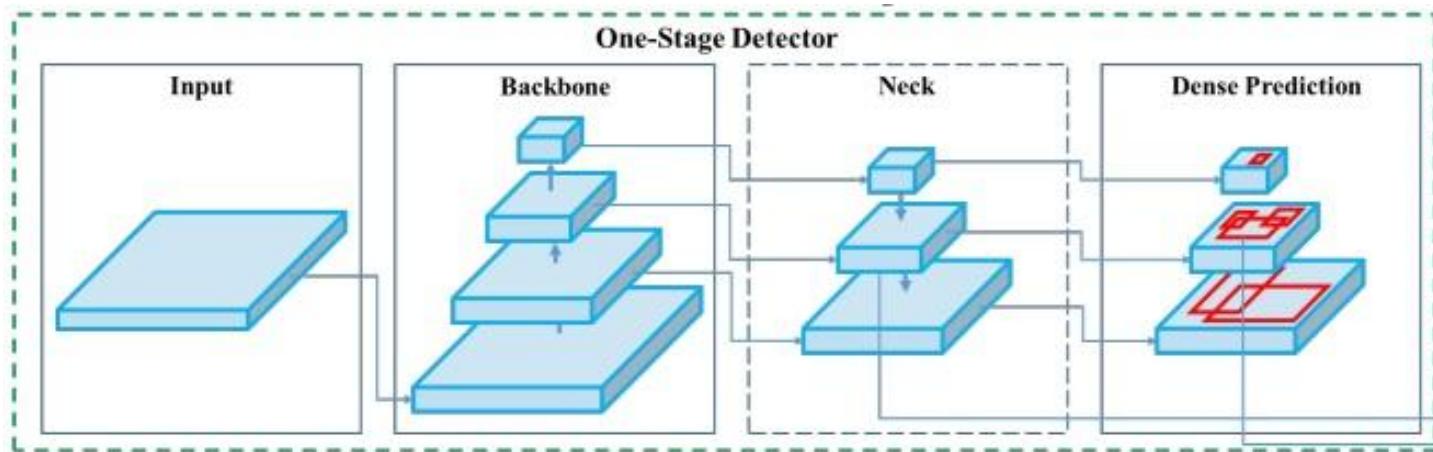
(2)

(4)



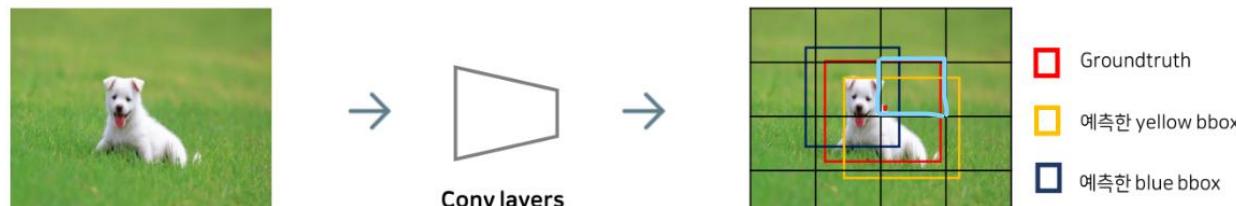
## ❖ YOLO 구조

- YOLO는 주로 backbone, neck, head로 구성됨
- Backbone은 convolution을 통해 input 이미지로 부터 feature map을 생성하는 파트
- Neck은 backbone과 head를 연결해주는 역할로, 다양한 scale에 대해 feature pyramid를 구성하여 head에서 서로 다른 사이즈의 object를 검출할 수 있게 해 줌
- Head는 검출이 이루어지는 부분으로 anchor box를 이용하여 물체의 bounding box를 생성



## ❖ YOLO 학습 과정 및 loss 함수

- YOLO가 학습할 때 탐지 과정에서 여러 개의 예측 bounding box를 생성하고, 그 중 IoU값이 높은 하나의 bounding box를 선택하여 학습
- Training loss는 SSE (Sum of squared Error)를 사용하며 bounding 박스의 좌표와 ground truth의 오차, class 분류 확률 값의 오차, cell내의 class별 확률 값의 오차를 사용함.



Input Image

Regression  
loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$

Confidence  
loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

Classification  
loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$