

# OBJECT DETECTION WITH PYTORCH



# **ABOUT OBJECT DETECTION**

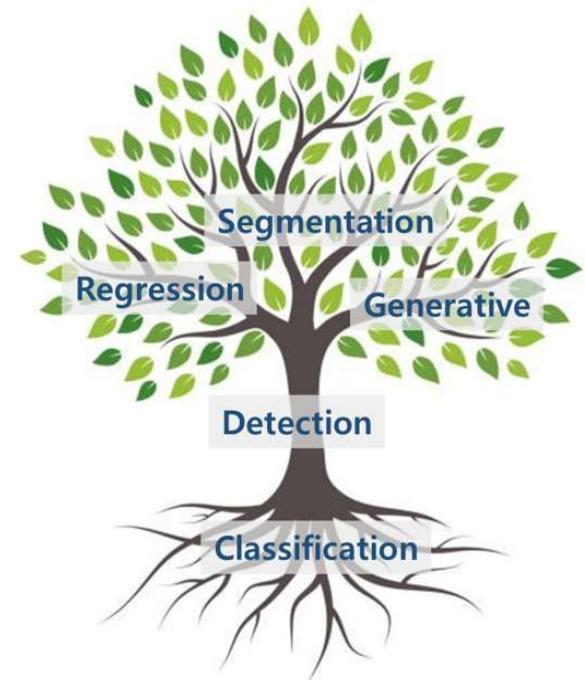
# ABOUT OBJECT DETECTION

3

## ◆ 비전 기술

### ■ 딥러닝 기반 컴퓨터 비전 → 대부분 CNN기반으로 개발

- 입력 : 이미지 또는 영상
- 출력 : 원하는 대로 생성
  - Image Classification (영상 분류)
  - Object Detection (객체 검출)
  - Image Segmentation (영상 분리)
  - Regression 응용 : 객체 추적, 랜드마크 검출, 동작분석
  - Generative



# ABOUT OBJECT DETECTION

4

## ◆ 비전 기술

### ❖ Image Classification

- 이미지 내 대표 객체 종류 분류
- 분류 할 객체 종류 → 사전 정의 되어야 함
- 이미지 데이터 셋 → 정의된 분류에 따라 annotation이 라벨링
- CNN Convolution 연산 적용 → Feature Map 특성 기반 대상 분류
- 출력 : K개 클래스에 대한 확률 (  $P_0, P_1, \dots, P_{k-1}$  )



CAT

# ABOUT OBJECT DETECTION

5

## ◆ 비전 기술

### ❖ Localization

- 이미지 내 **대표 객체 위치/영역 인식**
- 분류 할 객체 종류 → 사전 정의 되어야 함
- 이미지 데이터 셋 → 정의된 분류에 따라 annotation + 위치/영역 정보 라벨링
- 출력 : 위치/영역 경계 상자(bounding box)
  - 형식1 : 상자 왼쪽 위( $X_1, Y_1$ ), 오른쪽 아래( $X_2, Y_2$ ) 좌표
  - 형식2 : 상자 중심 좌표( $C_x, C_y$ )와 너비( $w$ ), 높이( $h$ )
  - 형식3 : 상자 왼쪽 위( $X, Y$ ) 좌표와 너비( $w$ ), 높이( $h$ )



CAT

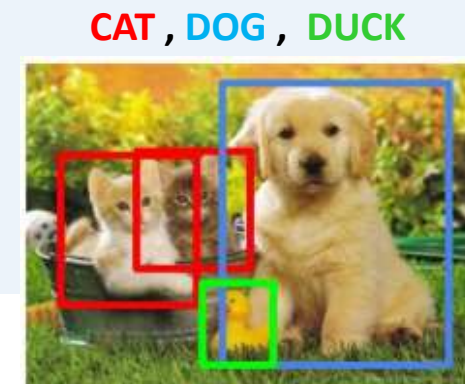
# ABOUT OBJECT DETECTION

6

## ◆ 비전 기술

### ❖ Object Detection

- 이미지 내 **다수 객체 인식**
- 각 객체의 **위치인식 및** 그 영역에 대해 **객체 종류 분류**
- 출력 : i번째 발견된 객체의 클래스에 대한 확률 (  $P_i^0, P_i^1, \dots, P_i^{k-1}$  )  
i번째 발견된 객체의 경계 상자 (  $x_1^i, y_1^i, x_2^i, y_2^i$  )
  - 형식1 : 상자 왼쪽 위( $x_1, y_1$ ), 오른쪽 아래( $x_2, y_2$ ) 좌표
  - 형식2 : 상자 중심 좌표( $C_x, C_y$ )와 너비( $w$ ), 높이( $h$ )
  - 형식3 : 상자 왼쪽 위( $x, y$ ) 좌표와 너비( $w$ ), 높이( $h$ )



# ABOUT OBJECT DETECTION

7

## ◆ 비전 기술

### ❖ Image Segmentation

- 이미지 내 **다수 객체 분리**
- 각 객체 **픽셀별로 객체 종류 분류**하고 **객체(instance) 구분**
- 종류 : semantic segmentation, instance segmentation, panoptic segmentation
- 출력 : i번째 발견된 객체의 클래스에 대한 확률 (  $P^i_0, P^i_1, \dots, P^i_{k-1}$  )  
i번째 발견된 객체의 경계 상자 (  $x^i_1, y^i_1, x^i_2, y^i_2$  )  
i번째 경계 상자 내부의 j번째 픽셀이 전경색일 확률 (  $F^i_j$  )

CAT , DOG , DUCK



# ABOUT OBJECT DETECTION

8

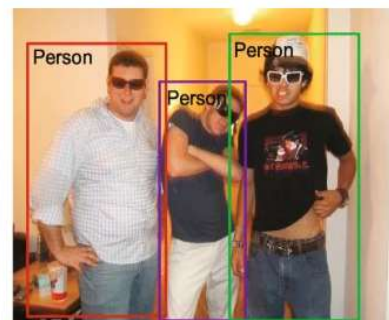
## ◆ 비전 기술

객체 범주 식별



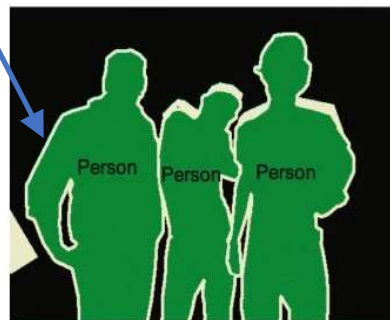
(a) Object classification

객체 카테고리 식별  
객체 위치 인식



(b) Object detection

각 픽셀 카테고리  
인식



(c) Semantic segmentation

각 픽셀 카테고리 및  
객체 카테고리 인식



(d) Instance segmentation



# ABOUT OBJECT DETECTION

9

## ◆ 객체 감지

- 컴퓨터 비전과 이미지 처리와 관련된 컴퓨터 기술
- **이미지 및 비디오 내에서 유의미한 특징 객체를 감지**하는 작업

### 【 객체 탐지 분야 】

- 얼굴 인식(Face Detection)
- Video Tracking(비디오 추적)
- 사람 수 세기(People Counting)
- 자율 주행 자동차, 의료 데이터

# ABOUT OBJECT DETECTION

10

## ◆ 객체 감지 발전 단계

❖ 딥러닝 기반 컴퓨터 비전 → 대부분 CNN기반으로 개발

- **LeNet-5**은 1998년 Yann LeCun 교수가 발표한 CNN 알고리즘으로 지속적인
- 연구와 발전 진행, 특히 2010년 초중반에 많은 발전

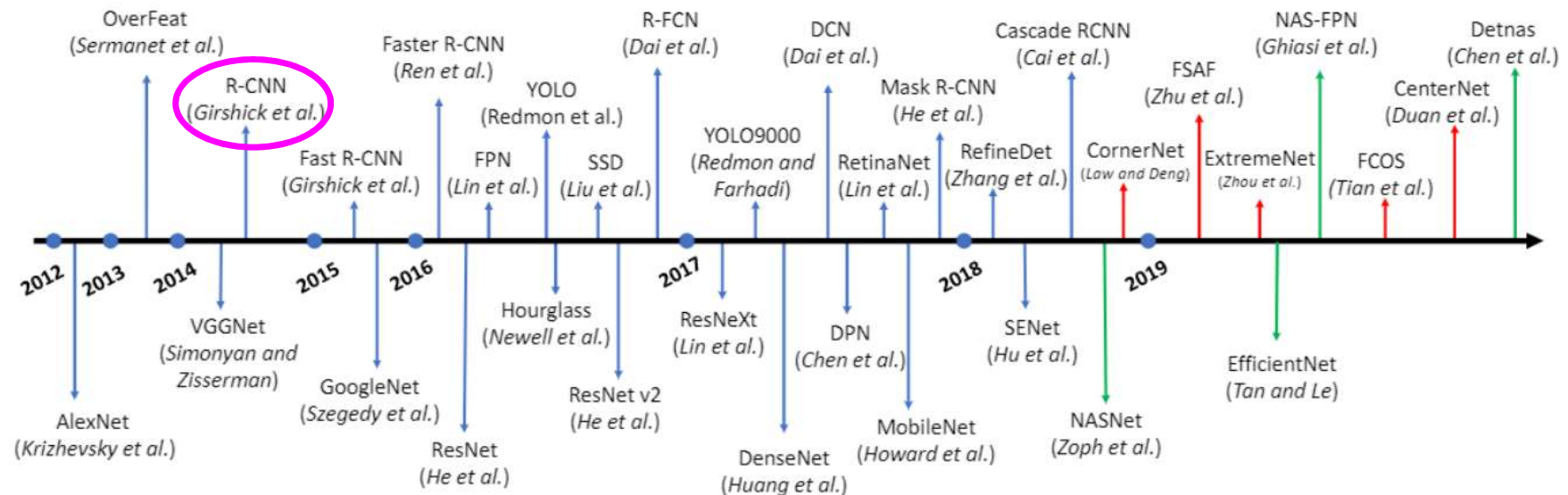


# ABOUT OBJECT DETECTION

11

## ◆ 객체 감지 발전 단계

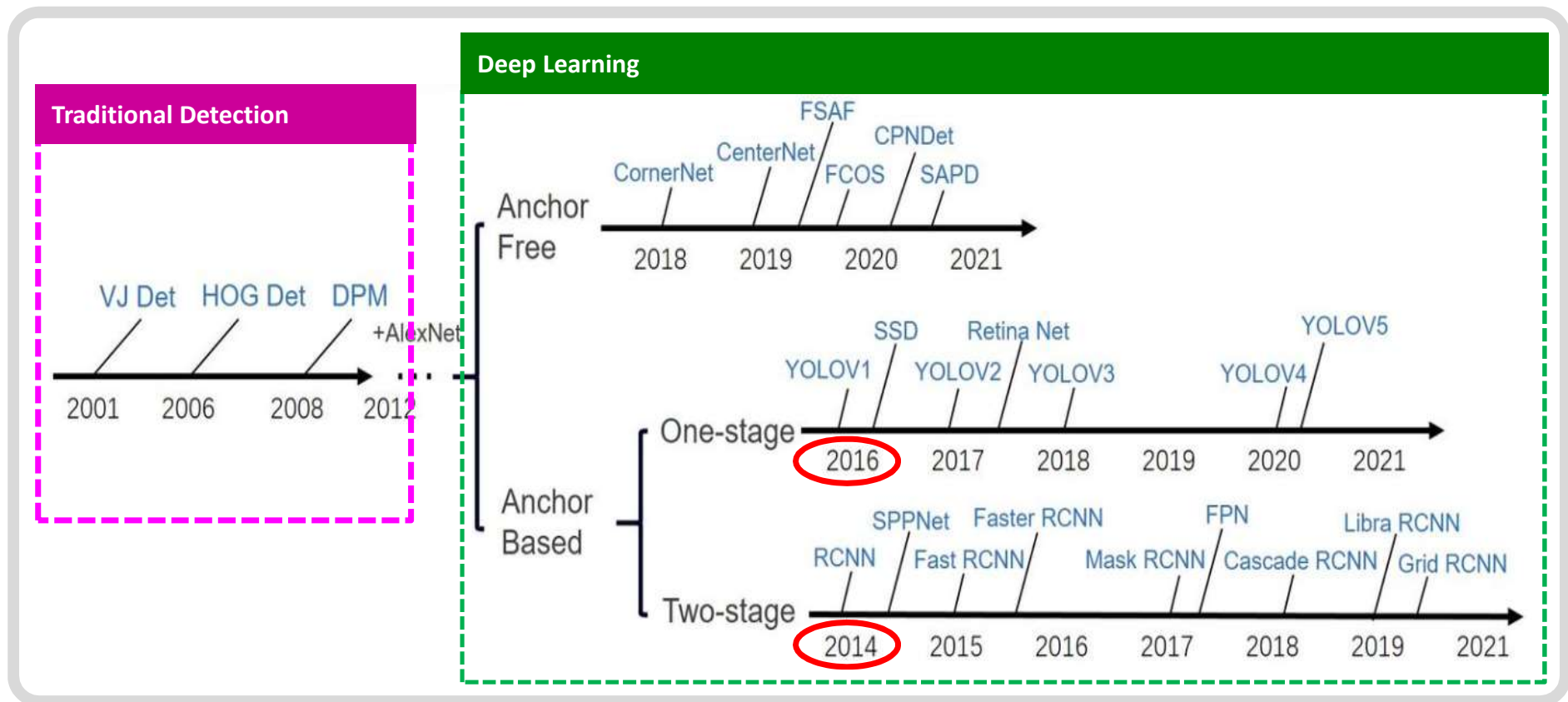
❖ 딥러닝 기반 컴퓨터 비전 → 대부분 CNN기반으로 개발



# ABOUT OBJECT DETECTION

12

## ◆ 객체 감지 발전 단계



# ABOUT OBJECT DETECTION

13

## ◆ 객체 감지 방식

### ❖ 전통적 방식 (~ 2012)

- **Sliding Window 방식** : Window를 좌측 상단 → 우측 하단 이동하며 Object Detection
  - Window : 연속된 데이터로 구성된 일정 크기 데이터, 배열, 문자열, 데이터 스트림 등에 효과적
  - 데이터를 패킷으로 분할하여 네트워크 전송 시, 모든 패킷이 순서대로 손상 없이 도착하지 않는 문제 해결 위해 탄생
- 활용
  - 연속된 데이터에서 [최대값/최소값 찾기](#)
  - [평균, 중간값 계산](#)
  - **패턴 탐색** (예: 문자열 내 서브스트링 찾기)
  - 네트워크 통신에서의 [데이터 전송 관리](#)

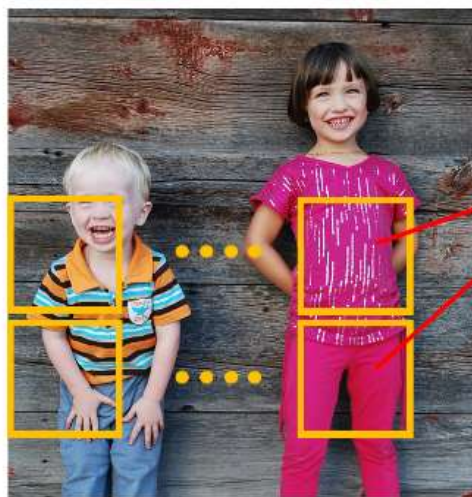
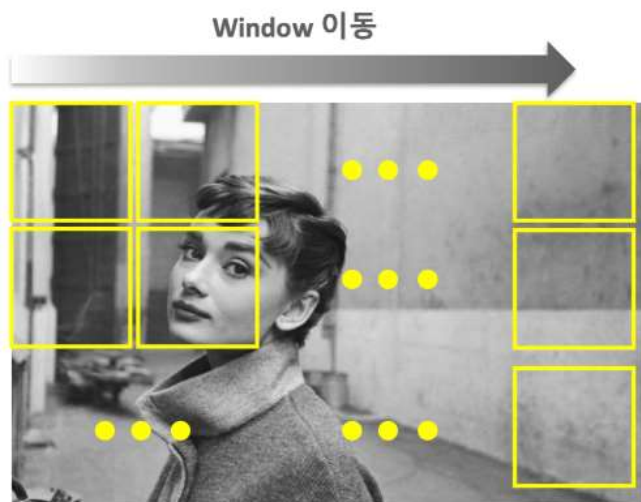
# ABOUT OBJECT DETECTION

14

## ◆ 객체 감지 방식

### ❖ 전통적 방식 (~ 2012)

- 동작원리 : CNN에서 커널이 입력 이미지와 연산하며 움직이는 것처럼 왼쪽 → 오른쪽 하단 이동



→ 여러 형태 Window + 고정 이미지

→ 고정 Window + 크기 변경 이미지

?

인식 X

# ABOUT OBJECT DETECTION

15

## ◆ 객체 감지 방식

### ❖ 전통적 방식 (~ 2012)

- 장점
  - 한 번에 여러 패킷 전송하여 네트워크 대역폭 **효율적** 사용
  - 잘못된 패킷이나 손실된 패킷 재전송 가능해 **정확한 데이터 전송 보장**
- 단점
  - Object가 없는 영역도 **무조건 Sliding (Exhaustive Search) → 계산량 많아짐!**
  - 여러 형태의 Window와 여러 Scale을 가진 이미지를 SCAN해서 검출
  - **수행 시간이 오래 걸리고 검출 성능이 상대적으로 낮음**
  - 많은 이미지에서 Object 없는 영역, 즉 배경이 대부분인 경우 많음

# ABOUT OBJECT DETECTION

16

## ◆ 객체 감지 방식

### ❖ 딥러닝 방식 (2012 ~)

- 컴퓨팅파워 발전과 대량 이미지 데이터(ImageNet) 기반 Classification 경진대회 개최
- 이미지 데이터를 기반으로 다양한 모델들 개발
- Sliding Window 방식의 단점을 해결하는 **Region Proposal(영역추정) 방식** 사용
- 기술 발전 분류
  - Anchor box 기반 : Two-shot-detection >>> One-shot-detection
  - Anchor free 기반 (2018~)



# ABOUT OBJECT DETECTION

17

## ◆ 객체 감지 방식

### ❖ 딥러닝 방식 (2012 ~)

▪ **Region proposal 방식** : Object가 있을 만한 후보 영역 Detection하는 방식

- sliding window 방법의 단점을 극복 위해 ROI (Range of Interest) 제시
  - **Exhaustive Search와 Segmentation, Greedy Algorithm 적용**한 알고리즘
  - 이미지의 모든 부분이 아닌 특정 영역에서 Detection 하기 위한 방식
- 
- 주변 pixel간 유사도 평가/비슷한 영역 합쳐 Segmented area 생성 → 반복 → 여러 개 ROI
  - 빠른 Detection과 높은 Recall 예측 성능을 동시에 만족하는 기법
  - Color, Texture, Size, Shape에 따라 유사한 Region 계층적 그룹핑 방법으로 계산

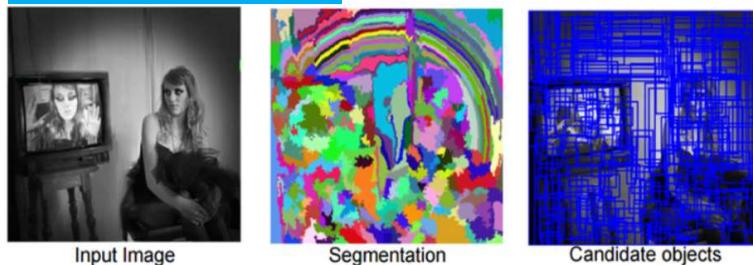
# ABOUT OBJECT DETECTION

18

## ◆ 객체 감지 방식

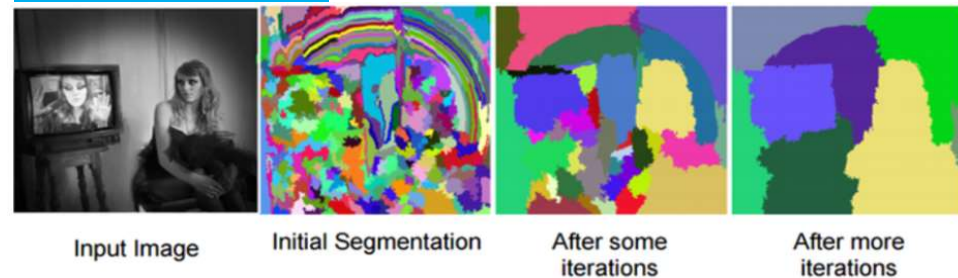
- **Selective Search 알고리즘 동작원리** : bounding box 위치 찾기 위해 적용된 대표적인 방법  
수많은 작은 영역 분할한 뒤 명암 차이 등 그룹화 기준으로 **영역들을 합치는 Bottom-up 방식**

### Selective Search 1단계



### Selective Search 2단계

비슷한 영역 1개 남을때까지 반복하며 **통합 By 유사도**



# ABOUT OBJECT DETECTION

19

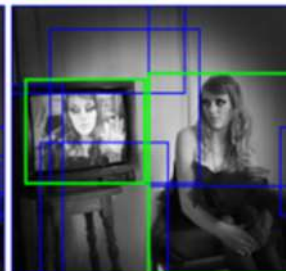
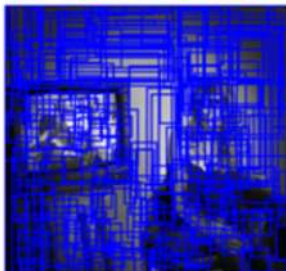
## ◆ 객체 감지 방식

### ▪ Selective Search 알고리즘 동작원리

Selective Search N단계 통합된 영역들을 바탕으로 후보 영역 생성 ← Box



Input Image



Greedy  
Algorithm

1. 가장 유사한 2가지 선택
2. 하나의 더 큰 영역으로 통합
3. 추가 결합할 영역 없을 때까지 반복

후보영역  
(candidate)

# ABOUT OBJECT DETECTION

20

## ◆ 객체 감지 용어 및 개념

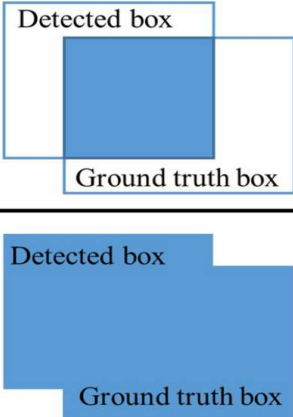
- **ROI : region of interest 관심영역** : object 존재 할 만한 위치
- **Region proposal 영역추정** : object 속할 가능성 있는 영역들의 ROI 추출
- **Ground Truth** : 학습하고자 하는 데이터의 **원본 혹은 실제 값**을 표현
- **IoU(Intersection over Union)** : 두 bounding box가 겹치는 비율 의미  
정답 bounding box와 예측한 bounding box의 IOU 비율이  
높을수록 정확히 예측한 것으로 판단
- **NMS(Non-maximum Suppression)** : 비최대 억제 알고리즘  
최대가 아닌 박스들(=Bounding Box)을 삭제하는 알고리즘  
IoU 값이 특정 임계점(threshold) 이상인 중복 bounding box 제거

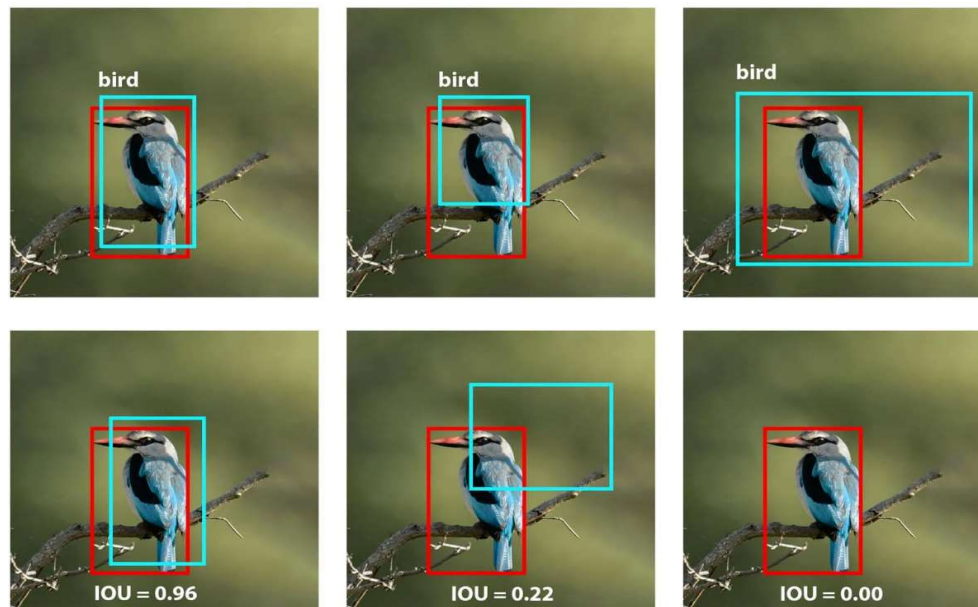
# ABOUT OBJECT DETECTION

21

## ◆ 객체 감지 용어 및 개념

- ROI : region of interest 관심영역 : object 존재 할 만한 위치

$$\text{IOU} = \frac{\text{Intersection}}{\text{Union}}$$




# ABOUT OBJECT DETECTION

22

## ◆ 객체 감지 용어 및 개념

▪ **Segmentation** : 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Semantic  
Segmentation

- 사진에 있는 **모든 픽셀을 해당하는 (미리 지정된 개수의) class로 분류**하는 것
- 이미지에 있는 모든 픽셀에 대한 예측이기 때문에 **dense prediction** 이라고도 함
- 같은 class의 instance 구별하지 않음 즉, 사람이 여러명 있다면 사람 class로만 분류
- 중첩은 분할하지 않고 인식
- 하늘이나 길과 같은 물체로 **셀 수 없는 클래스의 영역 분할로 가능**
- 주요 모델 : FCN, U-net, ParseNet, DeepLab, PSPNet
- 평가지표 : MIoU(Mean IoU) <--- Jaccard Index 자카드 지수라고도 함

# ABOUT OBJECT DETECTION

23

## ◆ 객체 감지 용어 및 개념

### ▪ Segmentation

: 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Semantic  
Segmentation



0: Background/Unknown

1: Person

2: Purse

3: Plants/Grass

4: Sidewalk

5: Building/Structures

# ABOUT OBJECT DETECTION

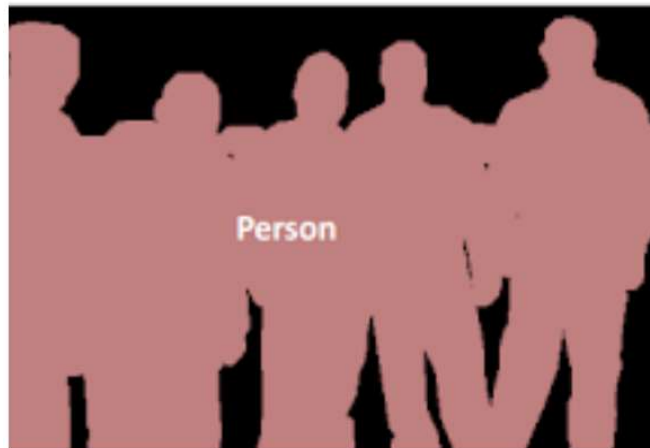
24

## ◆ 객체 감지 용어 및 개념

### ▪ Segmentation

: 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Semantic  
Segmentation





# ABOUT OBJECT DETECTION

25

## ◆ 객체 감지 용어 및 개념

▪ **Segmentation** : 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Instance  
Segmentation

- 기존 Semantic Segmentation과 달리 각각의 개체를 구분하는 방식
- background 같이 구분하기 애매한 것들 제외시키고 object 대상으로 하는 task
- **Object detection과 같은 물체의 인식을 픽셀 레벨에서 수행**
- 중첩의 경우 분할하여 인식
- 주요 모델 : Mask R-CNN
- 평가지표 : MAP (Mean Average Precision)

# ABOUT OBJECT DETECTION

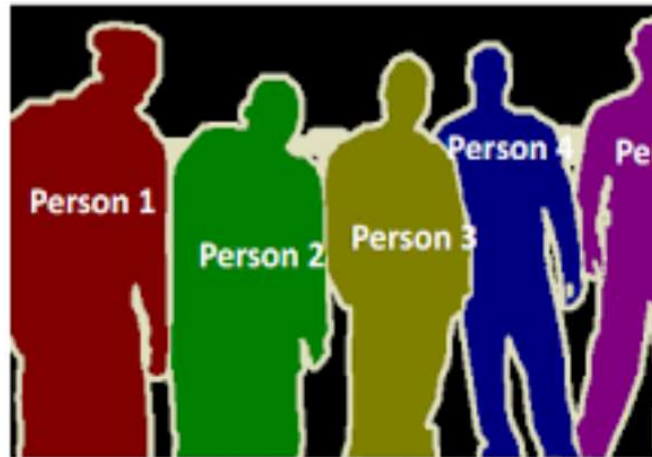
26

## ◆ 객체 감지 용어 및 개념

### ▪ Segmentation

: 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Instance  
Segmentation



# ABOUT OBJECT DETECTION

27

## ◆ 객체 감지 용어 및 개념

▪ **Segmentation** : 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Panoptic  
Segmentation

- **Semantic + Instance를 합친 형태**
- 이미지 안의 모든 화소에 대해 클래스 라벨을 예측하고 임의의 ID를 부여
- **이미지의 개별 객체를 식별하고 세분화 + 장면의 의미적 내용 식별**
- 셀수 있는 클래스 Thing에 대해서는 Instance Segmentation
- 셀수 없는 클래스 Stuff에 대해서는 Semantic Segmentation
- 평가지표 :  $PQ \text{ (Panoptic Quality)} = SQ \text{ (Semantation Quality)} * RQ \text{ (Recognition Quality)}$

# ABOUT OBJECT DETECTION

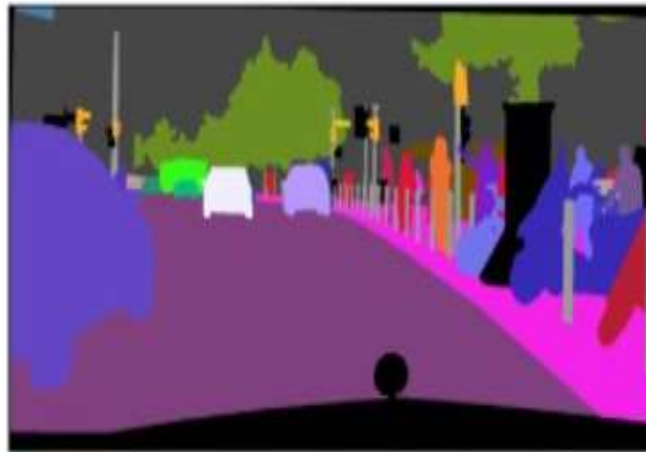
28

## ◆ 객체 감지 용어 및 개념

### ▪ Segmentation

: 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할

Panoptic  
Segmentation



# ABOUT OBJECT DETECTION

29

## ◆ 객체 감지 용어 및 개념

### ▪ Segmentation

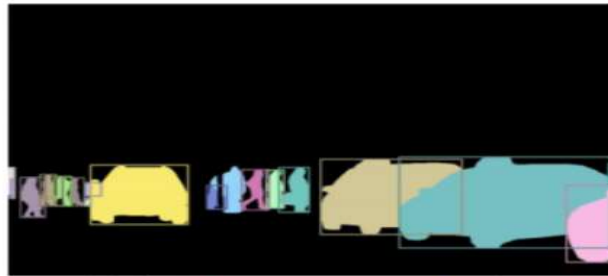
: 이미지 내 모든 픽셀별의 레이블 예측하는 분야 / 이미지 분할



(a) image



(b) semantic segmentation



(c) instance segmentation



(d) panoptic segmentation

# ABOUT OBJECT DETECTION

30

## ◆ 객체 감지 용어 및 개념

- **RPN (Region Proposal Network)** : 원본 이미지에서 region proposals 추출

- Selective Search 기반 Region Proposal 시 **많은 시간 소요에 대한 해결책**
- **Faster R-CNN 모델에 적용**
- **Anchor box (중심 좌표를 기준으로 여러 크기와 비율을 가지고 생성된 box)로 후보군 선정**
- 추후 학습 통해 Bounding box가 아닐 경우 : NMS (Non-maximum suppression) 통해 탈락
- 가능성있는 anchor box : Regression 하여 최종 bounding box를 찾아내는 기법

# ABOUT OBJECT DETECTION

31

## ◆ 객체 감지 용어 및 개념

▪ **Anchor box** : 특정 높이와 너비를 갖는 미리 정의된 경계 상자 세트

- 특정 aspect ratio 가진 object 탐지 위해 다양한 크기와 비율로 "미리 정의해놓은" bounding box
- 검출하려는 특정 객체 클래스의 스케일과 종횡비를 캡처하도록 정의
- 훈련 데이터셋의 객체 크기를 기반으로 선택
- 미리 정의된 앵커 상자는 검출 과정에서 영상 전체에 걸쳐 타일 형식으로 배치
- 신경망은 타일 형식 배치된 모든 앵커 상자에 대한 확률과 배경, IoU, 오프셋 등 기타 특성 예측
- 각기 다른 객체 크기에 여러 앵커 상자를 정의 가능

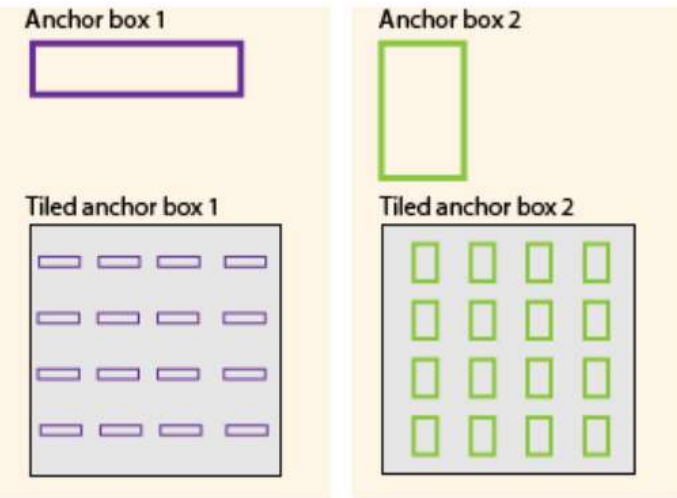
# ABOUT OBJECT DETECTION

32

## ◆ 객체 감지 용어 및 개념

▪ **Anchor box** : 특정 높이와 너비를 갖는 미리 정의된 경계 상자 세트

- 객체의 크기, 비율 등 객체의 특징에 해당하는 N개의 anchor box 디자인
- 이미지 전체를 SxS grid로 나눔
- 각 grid에서 디자인한 N개의 anchor box에 해당하는 feature 추출하여 network에 input으로 넣어주는 방식
- tiling : 각기 다른 가로, 세로, 비율을 가진 anchor box들이 이미지의 각 위치마다 놓는 것.



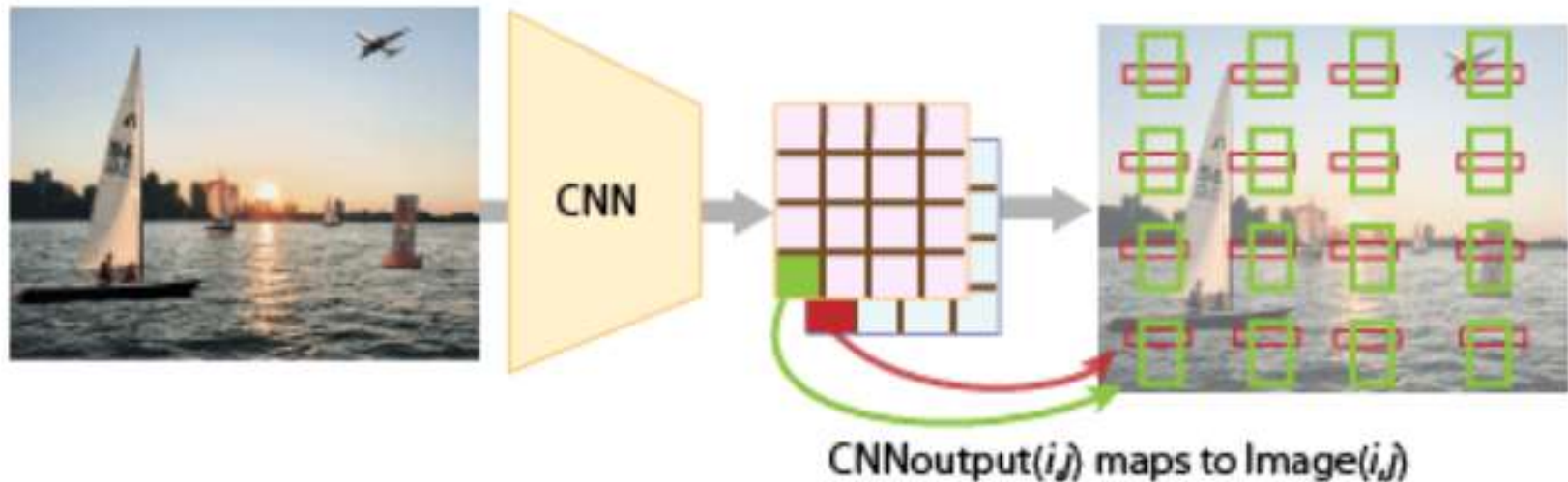


# ABOUT OBJECT DETECTION

33

## ◆ 객체 감지 용어 및 개념

- **Anchor box** : 특정 높이와 너비를 갖는 미리 정의된 경계 상자 세트



# ABOUT OBJECT DETECTION

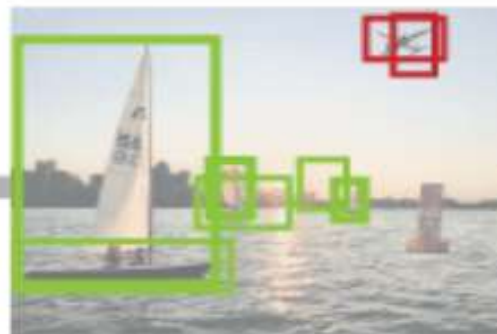
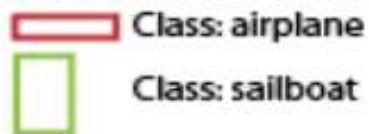
34

## ◆ 객체 감지 용어 및 개념

- **Anchor box** : 특정 높이와 너비를 갖는 미리 정의된 경계 상자 세트



Two anchor boxes



Filter by class scores,  
perform non-max suppression  
and intersection over union



# ABOUT OBJECT DETECTION

35

## ◆ 객체 감지 용어 및 개념

▪ **Anchor box** : 특정 높이와 너비를 갖는 미리 정의된 경계 상자 세트

- **장점**

- 학습 데이터에 높은 정확도

- **단점**

- anchor box 디자인 : 객체 특징 파악하여 사람이 직접 디자인, 데이터 변경 시 마다 작업
  - 높은 computation cost : 다양함 크기와 비율 가진 객체 탐지로 2개 이상 anchor 사용  
각 anchor별 feature 많아져 계산 비용 커짐
  - 낮음 inductive bias : 학습데이터 안에 미포함 특징을 가진 새로운 데이터 예측 시, 미리 설계된 anchor box로는 예측 성능 떨어짐

# ABOUT OBJECT DETECTION

36

## ◆ 객체 감지 용어 및 개념

- **FPN (Feature Pyramid Network)** : 이미지의 여러 가지 스케일의 feature map 사용 네트워크

- **Feature Pyramid란?**

객체 인식에 필요한 feature 들을 담고 있는 다양한 스케일의 feature map들

- FPN에서는 합성곱 신경망에서 자연스럽게 생성되는 feature map들을 활용하여 연산량을 적게 유지하면서도, 예측을 위한 적절한 정보들을 포함한 feature pyramid를 만드는 방법

# ABOUT OBJECT DETECTION

37

## ◆ 객체 인식 주요 구성 요소

### ❖ 영역 추정

Object가 있을만한 위치를 미리 추정 → Bounding Box Regression

### ❖ Deep Learning Network

Feature Extraction(Back-Born) + FPN(NECK) + Newtork Prediction(Head)

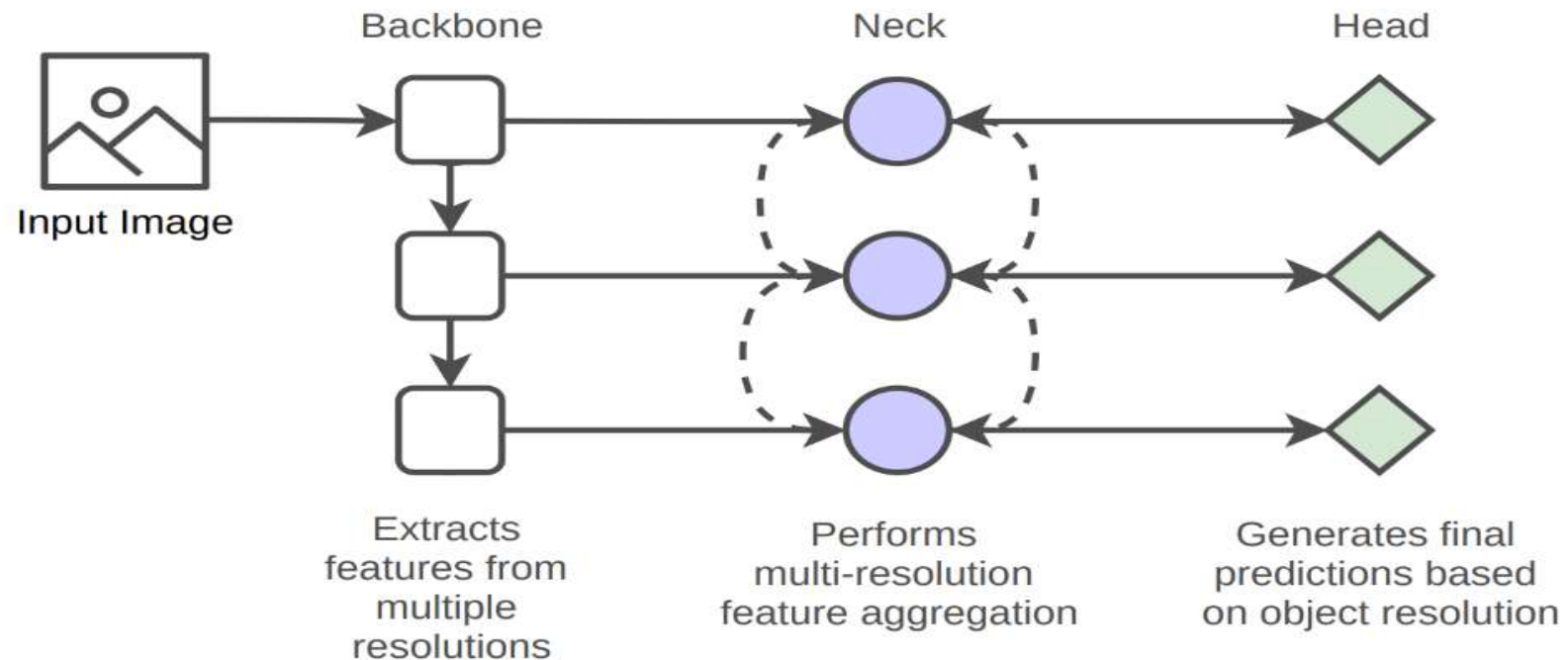
### ❖ Dectection 구성 요소

IoU, NMS, mAP, Anchor Box

# ABOUT OBJECT DETECTION

38

## ◆ 객체 인식 네트워크 구조



# ABOUT OBJECT DETECTION

39

## ◆ 객체 인식 네트워크 역할

### ❖ Backborn Network

- 입력 **이미지의 feature map**을 추출시켜주는 **부분**
- 사전학습된 모델로 **base model**이라고도 함
- 대표 : VGG16, ResNet50, Xception, InceptionV3, MobileNet

### ❖ Nect Network

- Backbone과 Head를 연결, Backbone에서 추출된 feature 적절하게 조화 시킴
- **feature map**을 **refinement(정제)**, **reconfiguration(재구성)**
- map을 upsampling하여 키우고, backbone의 feature map을 concat 등의 방식으로 같이 반영
- 대표 : FPN, PAN, BiFPN, NAS-FPN

# ABOUT OBJECT DETECTION

40

## ◆ 객체 인식 네트워크 역할

### ❖ Head Network

- Backbone에서 추출한 feature map의 **location** 및 **classification** 작업 수행
- 하나의 **Image**에서 여러 객체를 효과적으로 **detect** 하는 부분
- 대표 : [1-Stage] YOLO, SSD, [2-Stage] Faster R-CNN, R-FCN





# OBJECT DETECTION DATASET

# OBJECT DETECTION DATASET

42

## ◆ 객체 감지 데이터셋

### ❖ PASCAL VOC DATASET 크기

- 이미지 개수:
  - object detection : 주석 달린(annotated) 이미지 총 9,963개, 이 중 5,011개 학습 데이터
  - segmentation : 422개의 학습 데이터
- 이미지당 평균 object 수 : 2.4개
- 이미지당 평균 class 수 : 1.4개
- class 개수 : 20개

## ◆ 객체 감지 데이터셋

### ❖ PASCAL VOC DATASET 구조

- XML 포맷 형식
  - Annotation - Class
    - Bounding box (x,y,w,h),
    - Pose : 각각 오브젝트 방향성 정보
    - Truncated : 객체가 해당 이미지에 온전히 표현되지 못하고 잘려나갔는지 여부
    - Difficult : 인식하기 어려운지 정도
  - Image sets : 특정 클래스 어떤 이미지 인지 정보
  - JPEG Images : jpg 확장자 이미지 파일
  - Segmentation class: semantic segmentation 학습 위한 labeled images
  - Segmentation object: instance segmentation 학습 위한 labeled images

# OBJECT DETECTION DATASET

44

## ◆ 객체 감지 데이터셋

### ❖ PASCAL VOC DATASET

```
<filename>000001.jpg</filename>

<size> /*xml파일과 대응되는 이미지의 width, height, channels 정보에 대한 tag입니다.*/
  <width>353</width> /*xml파일에 대응되는 이미지의 width값입니다.*/
  <height>500</height> /*xml파일에 대응되는 이미지의 height값입니다.*/
  <depth>3</depth> /*xml파일에 대응되는 이미지의 channels값입니다.*/
</size>

<object> /*xml파일과 대응되는 이미지속에 object의 정보에 대한 tag입니다.*/
  <name>dog</name> /*오브젝트의 클래스가 무엇인지*/
  <pose>Left</pose> /*각각의 오브젝트의 방향성 정보*/
  <truncated>1</truncated> /*오브젝트가 해당 이미지에 온전히 표현되지 못하고 잘려나갔는지*/
  <difficult>0</difficult> /*인식하기 어려운지*/
  <bndbox> /*해당 object의 바운딩상자의 정보에 대한 tag입니다.*/
    <xmin>48</xmin> /*object 바운딩상자의 왼쪽상단의 x축 좌표값입니다.*/
    <ymin>240</ymin> /*object 바운딩상자의 왼쪽상단의 y축 좌표값입니다.*/
    <xmax>195</xmax> /*object 바운딩상자의 우측하단의 x축 좌표값입니다.*/
    <ymax>371</ymax> /*object 바운딩상자의 우측하단의 y축 좌표값입니다.*/
  </bndbox>
</object>
```

# OBJECT DETECTION DATASET

45

## ◆ 객체 감지 데이터셋

### ❖ ImageNET DATASET

- Large Scale Visual Recognition Challenge (ILSVRC) 대회에서 Image classification, Object detection 성능평가 데이터셋으로 사용
- ILSVRC 2012 데이터셋이 자주 사용
- <https://www.image-net.org/>

IMAGENET

14,197,122 images, 21841 synsets indexed

[Home](#) [Download](#) [Challenges](#) [About](#)

Not logged in. [Login](#) | [Signup](#)

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. The project has been instrumental in advancing computer vision and deep learning research. The data is available for free to researchers for non-commercial use.

Mar 11 2021. ImageNet website update.

# OBJECT DETECTION DATASET

46

## ◆ 객체 감지 데이터셋

### ❖ ImageNET DATASET 크기 및 특징

- 데이터셋 크기
  - 이미지 개수: 1000k
  - class 개수 1000개
- 데이터셋 특징
  - 이미지 내 object가 큰 편임
  - object가 중앙에 위치
  - 이미지당 object 수가 적음

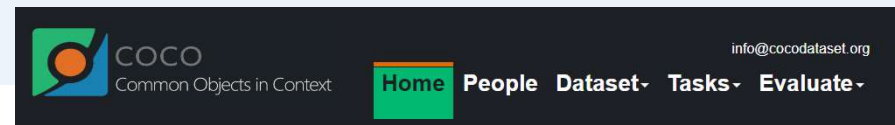
# OBJECT DETECTION DATASET

47

## ◆ 객체 감지 데이터셋

### ❖ COCO DATASET

- 2014년도 등장한 object detection, segmentation, keypoint detection 위한 데이터셋
- POSCAL, ImageNet 학습 후 현실세계 이미지 객체 인식 잘 못하는 문제의 대안 데이터셋
- 매년 다른 데이터셋으로 전 세계의 여러 대학/기업이 참가하는 대회에 사용
- 많은 그림 파일 가지고 있으므로, 용량이 GB 단위로 큼
- <https://cocodataset.org/#home>



#### News

- We are pleased to announce the [LVIS 2021 Challenge and Workshop](#) to be held at ICCV.
- Please note that there will not be a COCO 2021 Challenge, instead, we encourage people to participate in the LVIS 2021 Challenge.
- We have partnered with the team behind the open-source tool [FiftyOne](#) to make it easier to download, visualize, and evaluate COCO
- [FiftyOne](#) is an open-source tool facilitating visualization and access to COCO data resources and serves as an evaluation tool for model analysis on COCO.

# OBJECT DETECTION DATASET

48

## ◆ 객체 감지 데이터셋

### ❖ COCO DATASET 특징 및 장점

- 다양한 크기의 물체 존재
- 높은 비율로 작은 물체들이 존재
- Object들이 혼잡하게 존재하고, occlusion(폐색)이 많이 존재
- 어떤 카테고리에 속하는지 분류하기가 모호한 사진 많음 ← 현실 세계 반영



# OBJECT DETECTION DATASET

49

## ◆ 객체 감지 데이터셋

### ❖ COCO DATASET 크기

- 이미지 개수:
  - 학습(training) 데이터셋: 118,000장의 이미지
  - 검증(validation) 데이터셋: 5,000장의 이미지
  - 테스트(test) 데이터셋: 41,000장의 이미지
- 이미지당 평균 object 수: 8개      이미지당 평균 class수: 3.5개      class 개수 : 91개
- 수행하는 task
  - classification, object detection      - semantic segmentation, instance segmentation
  - pose estimation etc

# OBJECT DETECTION DATASET

50

## ◆ 객체 감지 데이터셋

### ❖ COCO DATASET 구조

- JSON format
- Annotation:
  - captions : 이미지에 대한 설명 텍스트
  - instances : 이미지에 있는 인스턴스에 대한 카테고리/class와 영역 Bounding box (x,y,w,h)
  - person\_keypoint : 자세 관련 데이터
- JPEG Images

# OBJECT DETECTION DATASET

51

## ◆ 객체 감지 데이터셋

### ❖ COCO DATASET 구조

```
"images": [  
  ...  
  {  
    "license": 1,  
    "file_name": "000000324158.jpg",  
    "coco_url": "http://images.cocodataset.org/val2017/000000324158.jpg",  
    "height": 334,  
    "width": 500,  
    "date_captured": "2013-11-19 23:54:06",  
    "flickr_url": "http://farm1.staticflickr.com/169/417836491_5bf8762150_z.jpg",  
    "id": 324158  
  },  
  ...  
],
```

# OBJECT DETECTION DATASET

52

## ◆ 객체 감지 데이터셋

### ❖ COCO DATASET 구조

```
"annotations": [  
  ...  
  {  
    "segmentation": [  
      [  
        216.7,  
        211.89,  
        216.16,  
        217.81,  
        215.89,  
        220.77,  
        ...  
        212.16  
      ]  
    ]  
  },  
]
```

```
"area": 759.3375500000002,  
"iscrowd": 0,  
"image_id": 324158,  
"bbox": [  
  196.51,  
  183.36,  
  23.95,  
  53.02  
],  
"category_id": 18,  
"id": 10673  
},
```



# **OBJECT DETECTION MODELS**

# R-CNN

Region-based Convolutional Neural Networks

## ◆ 특징

- CNN은 이미지 분류 분야 엄청난 성능 → Object Detection 분야 바로 적용 못함
- 2013년 11월 로스 기르쉬크 제안
- **2014년 R-CNN 등장으로 CNN을 Object Detection 분야에 최초로 적용**
- CNN기반 Object Detection 분야도 높은 수준 성능 이끌어 낼 수 있다
- 향후 **Fast R-CNN, Faster R-CNN, Mask R-CNN** 등의 기본이 됨
- **Regions with Convolutional Neural Networks features** 약자

R-CNN Family



## ◆ 특징

- 전체 task 두 단계 분리 → 2-Stage
  - 1 단계 → 물체 위치 찾는 Region Proposal
  - 2 단계 → 물체 분류 Region Classification
- 구조
  - Region Proposal
  - (pre-trained) CNN(AlexNet) + ( SVM + Bounding Box Regression )



## ◆ 특징

### ▪ 학습 Fine-tuning

- 미리 학습된 AlexNet CNN 모델 사용
- 해당 모델의 가중치를 고정하고 Bounding Box 후보 영역 학습  
: IoU 0.5이상 - 클래스, 0.5 미만 - 배경
- SVM과 같은 전통적인 머신 러닝 알고리즘 활용하여 Object Detection 수행  
: IoU 0.3이상이며 GT - 클래스, 0.3 미만 - 배경

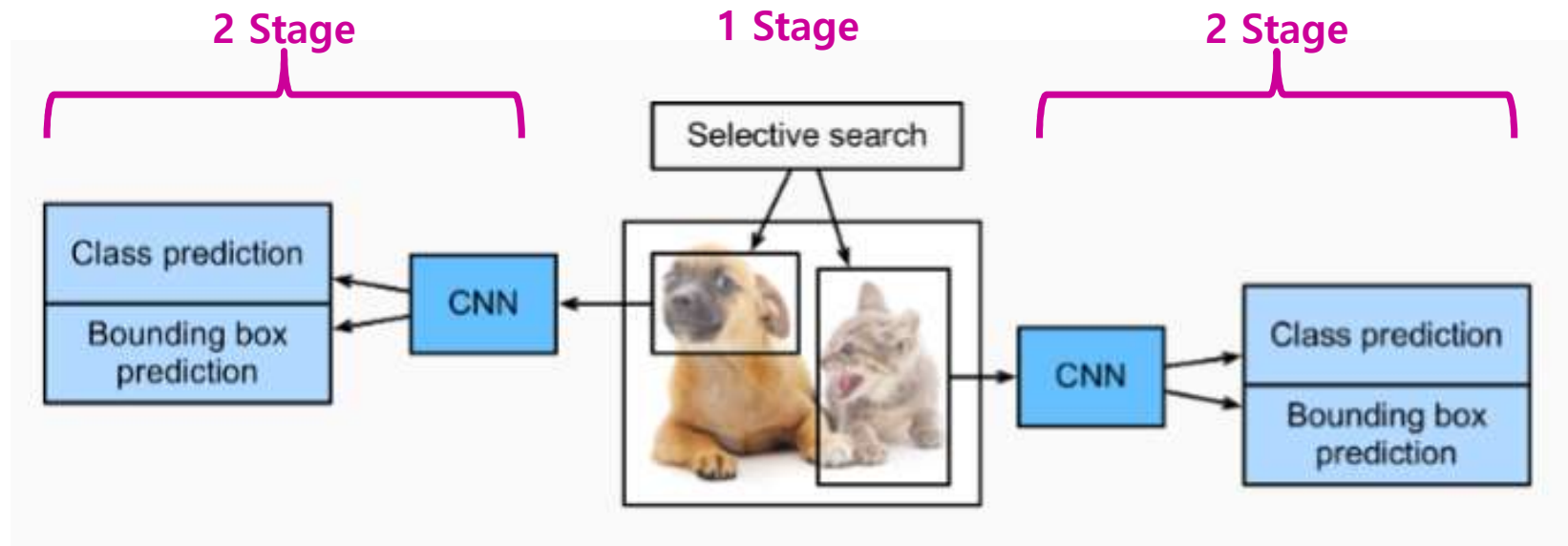
# R-CNN

Region-based Convolutional Neural Networks

58

## ◆ 특징

### ❖ 2-stage Detector



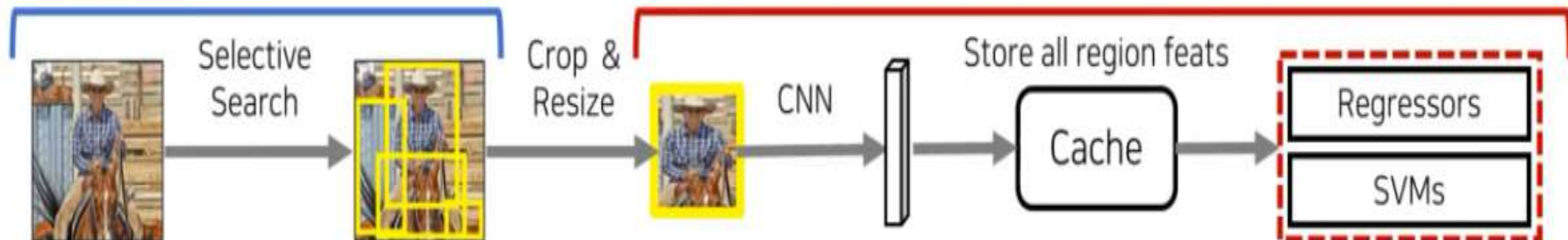
# R-CNN

Region-based Convolutional Neural Networks

59

## ◆ 특징

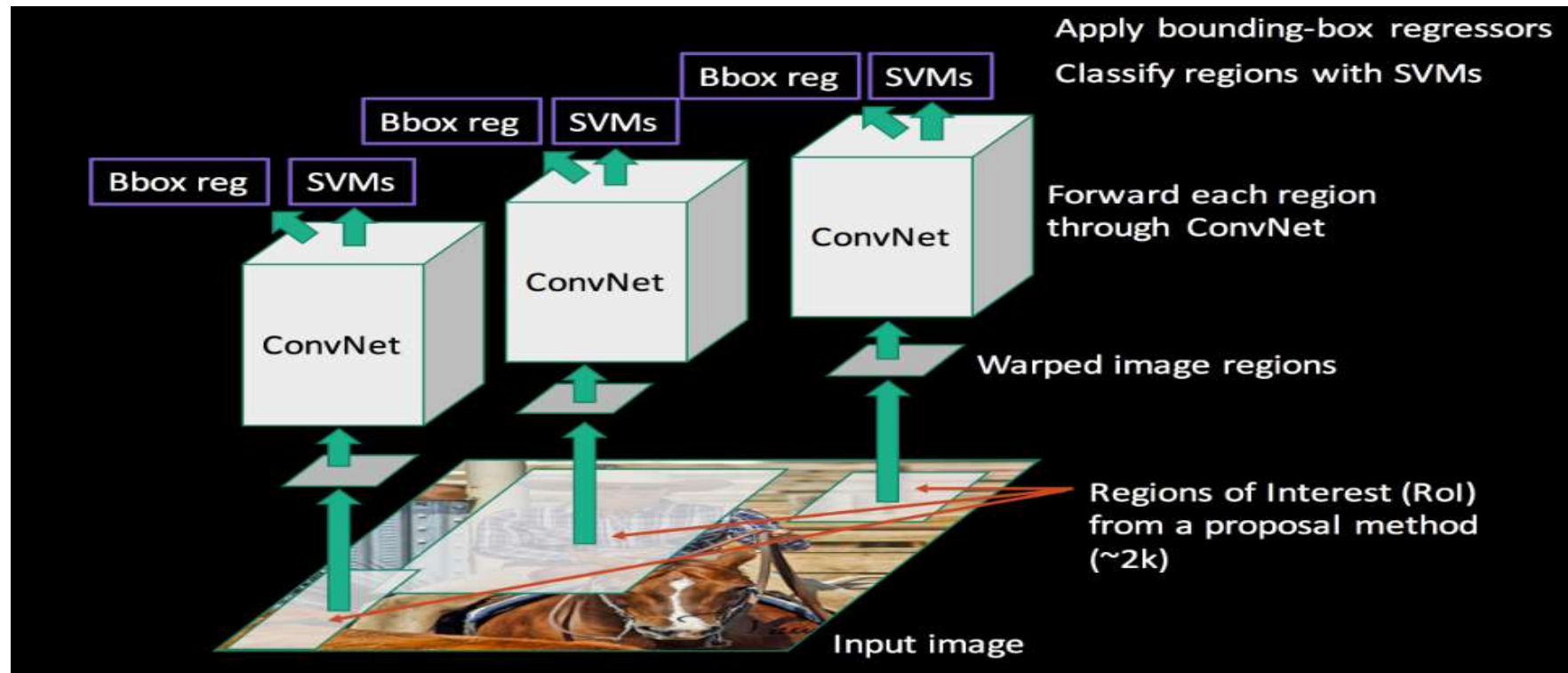
### ❖ 2-stage Detector



# R-CNN Region Based Convolutional Neural Networks

60

## ◆ 구조



## ◆ 구조

### ❖ 1- STAGE : Region proposal

- Selective search 알고리즘을 통해 ROI를 2000여 개 정도의 Region 추출
- 추출된 ROI에 bounding box 생성

### ❖ Pre-Processing

- CNN에 넣기 전에 같은 사이즈(227 x 227)로 warping (Crop & Resizing)
- 일률적 같은 사이즈로 만들기 때문에 input image가 왜곡되고 정보가 소실되는 현상 발생

## ◆ 구조

### ❖ 2- STAGE : ① CNN

- 이미지넷 데이터(ILSVRC2012 classification)로 미리 학습된 CNN 모델 사용
- Object Detection용 데이터 셋으로 fine tuning하는 방식
- 각 클래스별로 IoU가 0.5가 넘으면 positive sample, 그렇지 않으면 "background"라고 labeled

### ❖ 2- STAGE : ② SVM(Support Vector Machine) + bounding box regression

- CNN을 통해 나온 feature map을 활용
- Linear SVM(Support Vector Machine) 통한 분류
- Regressor를 통한 bounding box regression

## ◆ 분류 방식

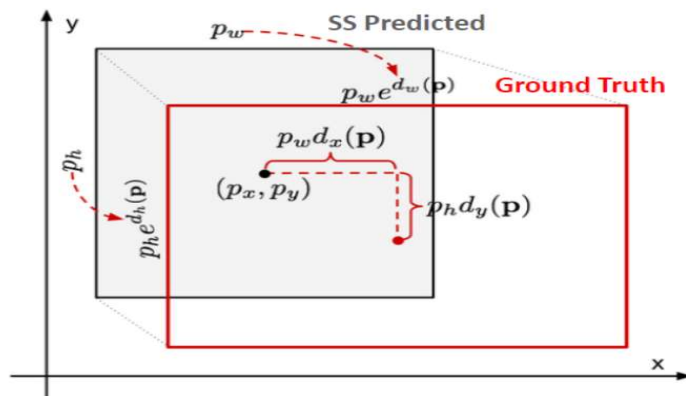
### ❖ Classification → Linear SVM(Support Vector Machine)

- IOU 기준이 0.3
- 0.3 이상 >> ground-truth boxes(정답 박스)만 positive
- 0.3 미만 >> 모두 negative

## ◆ 회귀 방식

### ❖ Regressor → BBR( Bounding Box Regression )

- 레이블  $y$ 와 CNN 후 output의 Bounding Box → 차이
- 차이를 줄이도록 조정하는 선형회귀 모델 절차
- Bounding Box Regression 나온 값을 CNN 단계 전으로 전달 → Region Proposal 잘 되도록!



- regression 통해 나온 SS predicted의 중앙값( $p_x, p_y$ )값과 Ground Truth의 중앙 값의 차 통해 판단
- 중앙값 거리의 차(loss)가 최소 되도록 optimization



## ◆ 단점

- AlexNet을 그대로 사용하기 위해 image를 강제로 변형 시켜야 함
- warping 하는 과정에서 input **이미지가 왜곡되고 정보 손실 발생**
- Selective Search(CPU사용)를 통해 뽑힌 2000개의 Region proposal 후보를 모두 CNN에 집어넣기 때문에 **training / testing 시간이 오래 걸림**
- Selective Search나 SVM이 GPU에 적합한 구조가 아님
- CNN, SVM, Bounding Box Regression 세 가지 모델 결합된 형태로 **한 번에 학습 불가능**
- **Back propagation이 안되므로 CNN은 업데이트 되지 않음**
- **Real-Time 분석 안됨!**

# SPP-Net

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

## ◆ 특징

- 2014년 R-CNN의 입력 이미지 왜곡과 CNN 여러 번 통과 문제점 해결
  - CNN Network에 입력 시 고정 크기로 변형된 이미지에 대한 feature 추출 / 정보 손실 & 왜곡
    - ➔ crop 적용 : crop된 구역만 CNN을 통과시키기 때문에, 전체 이미지 정보가 손실이 발생
    - ➔ warp 적용 : 이미지에 변형



## ◆ 특징

### ▪ Spatial Pyramid Pooling layer 추가

- 원인] RNN의 AlexNet 224x224 입력 크기 고정, CNN+FC 구성 때문
- 해결] CNN과 FC Layer 분리
  - 이미지 전체 → CNN에 입력
  - 결과 Feature Map → Resion proposal 제안된 영역부분만 crop
  - 잘라낸 Feature map → Spatial Pyramid Pooling 과정 거쳐 고정된 크기 벡터로 변환
  - 변환된 벡터 → FC layer에 입력한 뒤 최종적인 output feature 추출
- Feature 추출 위해 → **R-CNN은 CNN 2000번 연산 →: SPP-Net CNN 1번 연산**

## ◆ SPP (Spatial Pyramid Pooling)

### ▪ 피라미드 Pyramid

다양한 사이즈를 동일한 feature map에 적용해서 전부 Concat하는 구조

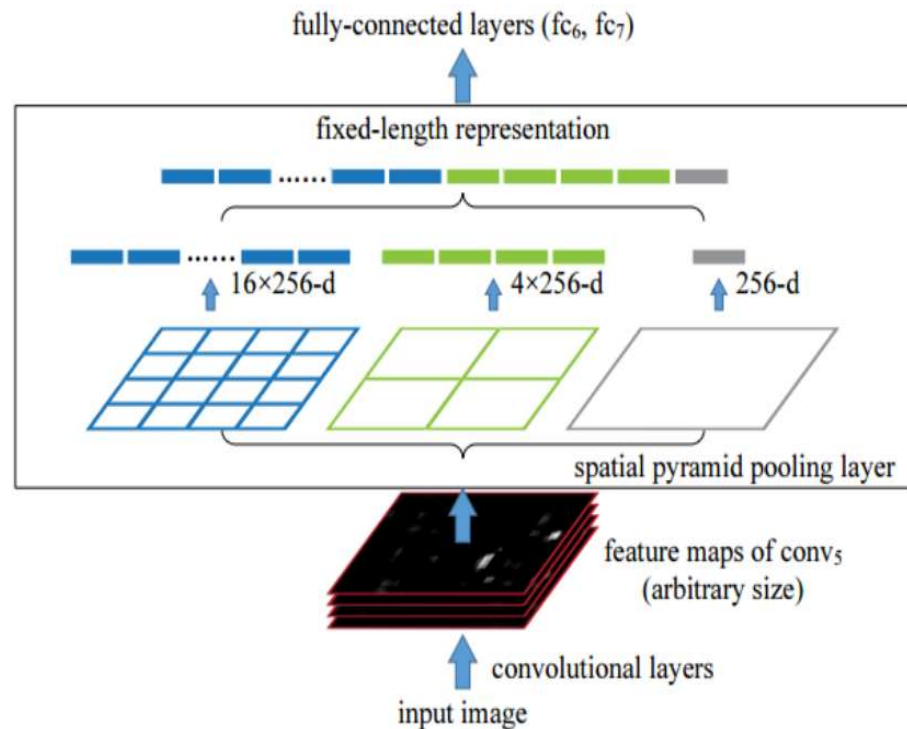
### ▪ SPP Layers

5개 conv layer와 3개 fc layer

### ▪ spatial bins

fc layer에 입력될 feature 수  $\rightarrow 16+4+1 = 21$

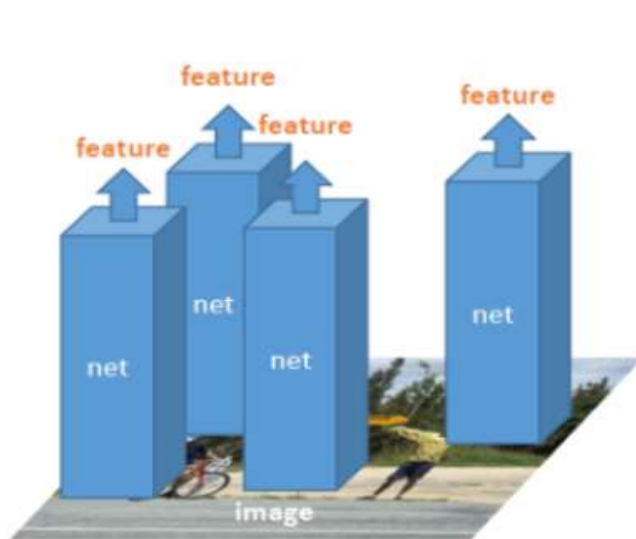
4x4, 2x2, 1x1 영역 피라미드로 feature map 분리



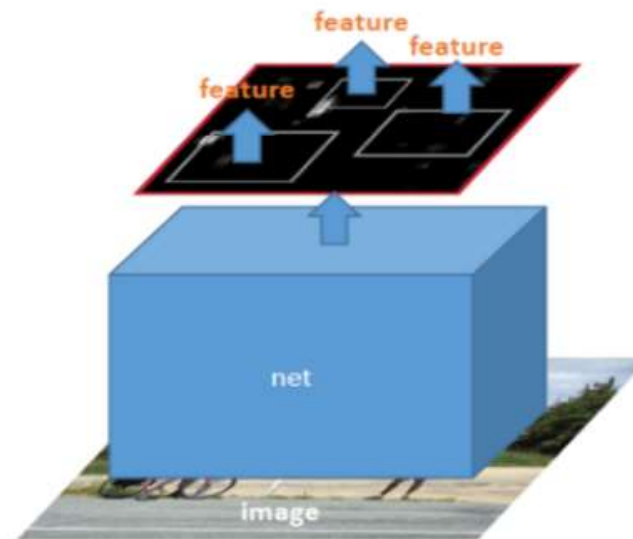
# SPP-Net

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition 70

## ◆ SPP (Spatial Pyramid Pooling)



**R-CNN**  
2000 nets on image regions



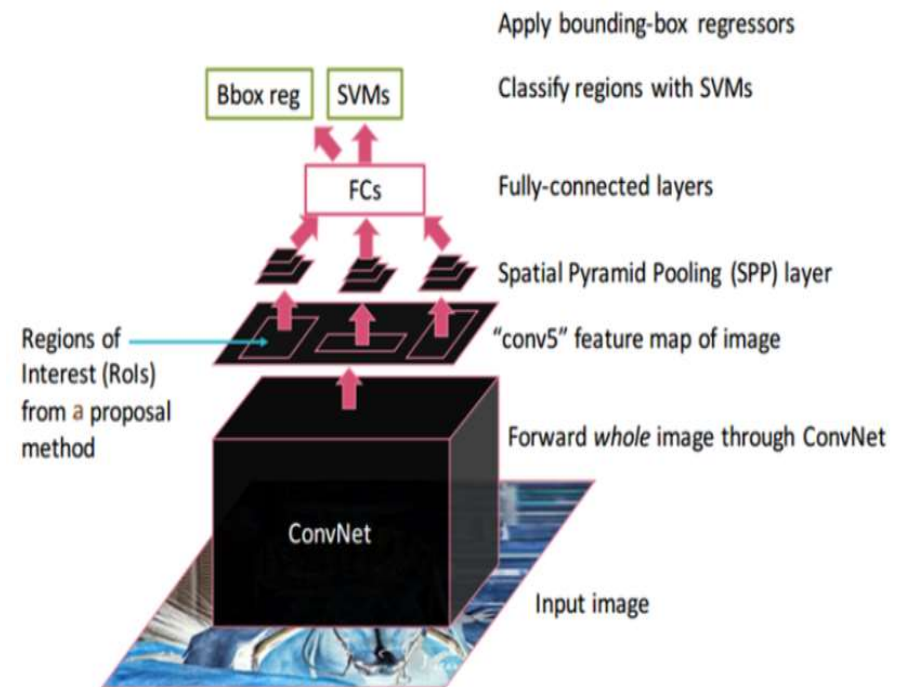
**SPP-net**  
**1 net on full image**

# SPP-Net

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition 71

## ◆ 구조 및 동작

- (1) 이미지를 CNN에 통과시켜 **feature map 추출**
- (2) Selective Search 통해 2000개 region proposals 생성
- (3) 원본 이미지에서 추출된 Feature Map에  
2000개의 Region Proposal 이미지를 매핑  
2000개의 feature map 추출
- (4) SPP layer를 적용하여 얻은 고정된 크기 벡터 추출  
FC layer에 전달
- (5) SVM으로 카테고리를 분류
- (6) Bounding box regression 예측 수행



# Fast R-CNN

Fast Region-based Convolutional Network Method



## ◆ 특징

- R-CNN의 한계점을 극복하고자 2015년 로스 기르쉬크가 발표
  - RoI (Region of Interest) 마다 CNN연산으로 속도저하  
→ RoI pooling 해결
  - CNN Network에 입력 시 고정 크기로 변형된 이미지에 대한 feature 추출 / 정보 손실 & 왜곡  
→ 입력으로부터 feature 추출 + 분류/회귀를 하나의 모델로 수행
  - multi-stage pipelines으로 모델 한번에 학습 불가  
→ CNN 특징 추출부터 하나의 모델에서 학습

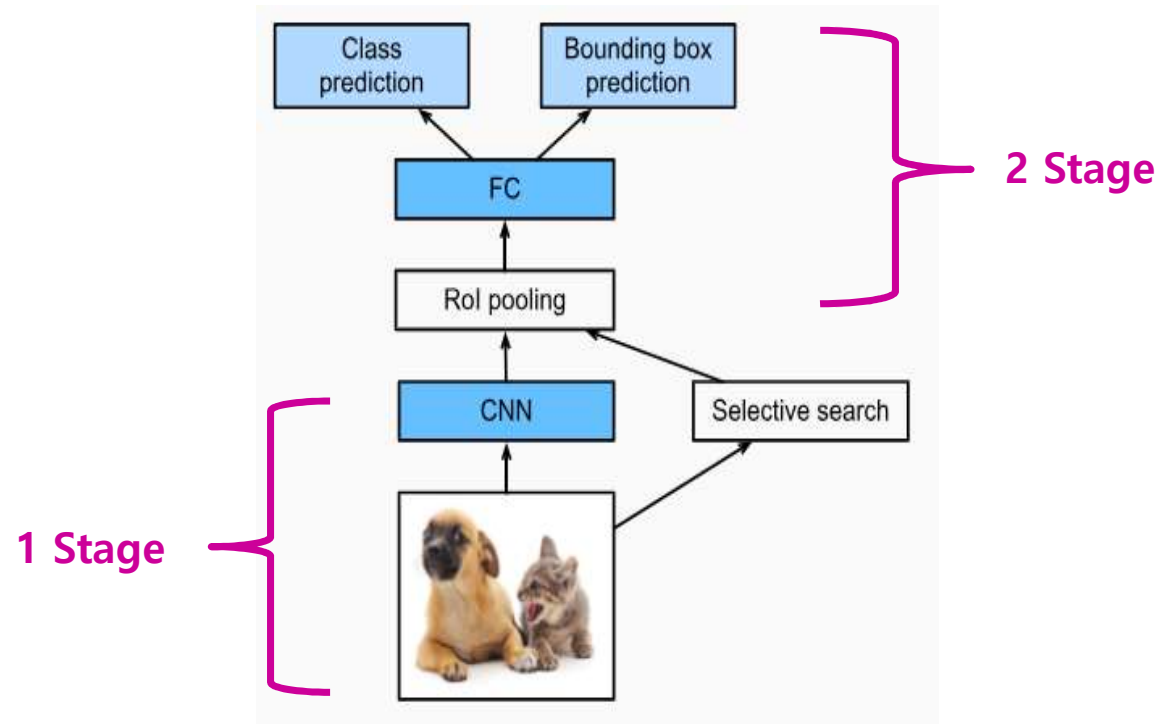
# Fast R-CNN

Fast Region-based Convolutional Network Method

74

## ◆ 특징

### ❖ 2-stage Detector



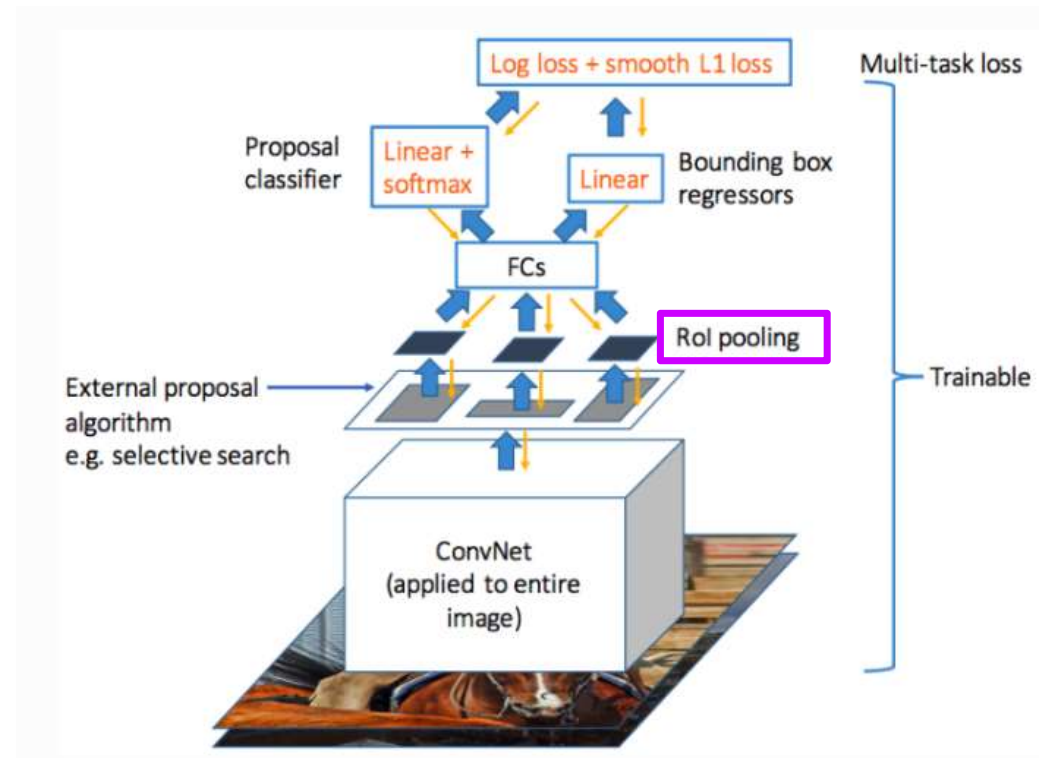
# Fast R-CNN

Fast Region-based Convolutional Network Method

75

## ◆ 구조

Feature extraction (CNN)  
+ Classification (Softmax)  
+ Bounding box regression 과정  
통합



# Fast R-CNN

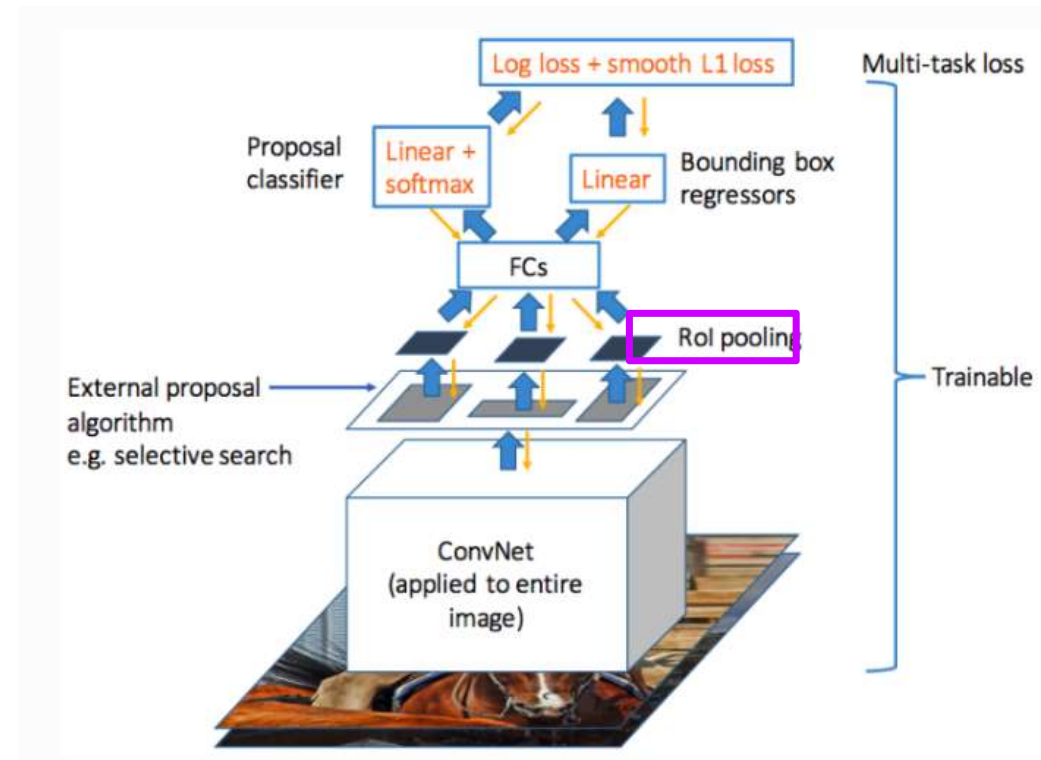
Fast Region-based Convolutional Network Method

76

## ◆ 구조

### ❖ Feature extraction (CNN)

Input image를 미리 학습된 VGG16  
CNN통과시켜 feature map을 추출



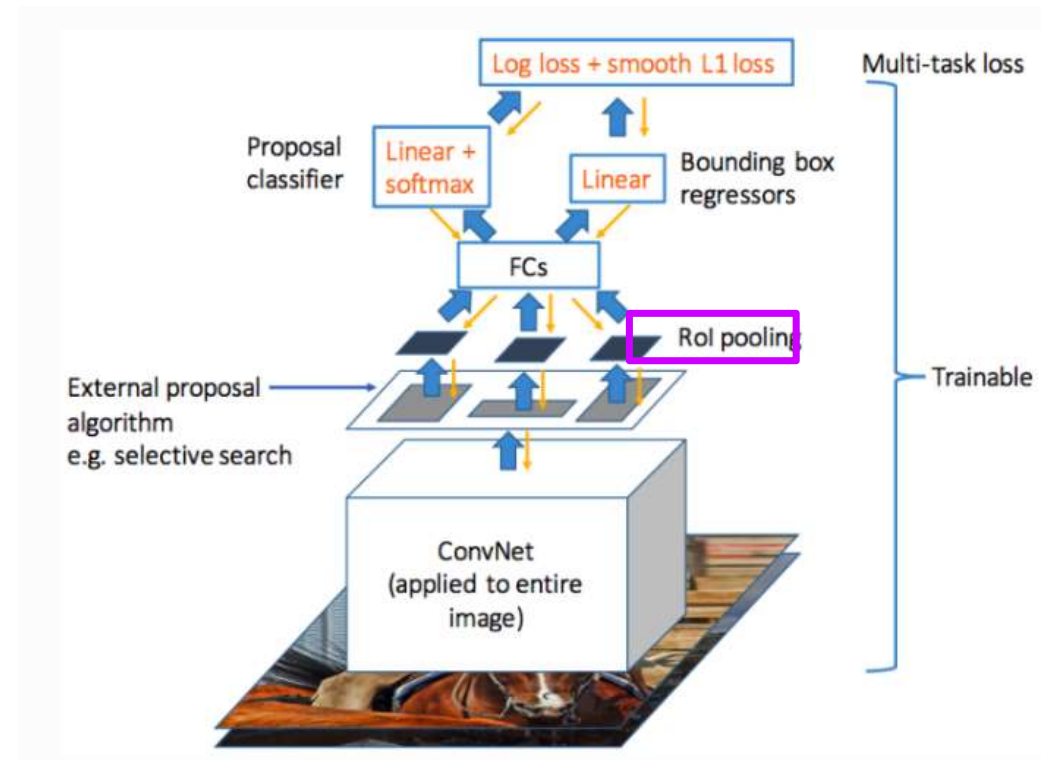
# Fast R-CNN Fast Region-based Convolutional Network Method

77

## ◆ 구조

### ❖ Region proposal

- ① Selective Search 통해 각각의 ROI에 대한 B-Box 그림
- ② 추출된 feature map에서 ROI의 해당 영역 찾기
  - FM은 CNN과정으로 크기 작아짐
  - B-Box 크기와 중심좌표를 FM에 맞게 변경시켜서 적용
  - **ROI Feature Map** 추출됨



# Fast R-CNN

Fast Region-based Convolutional Network Method

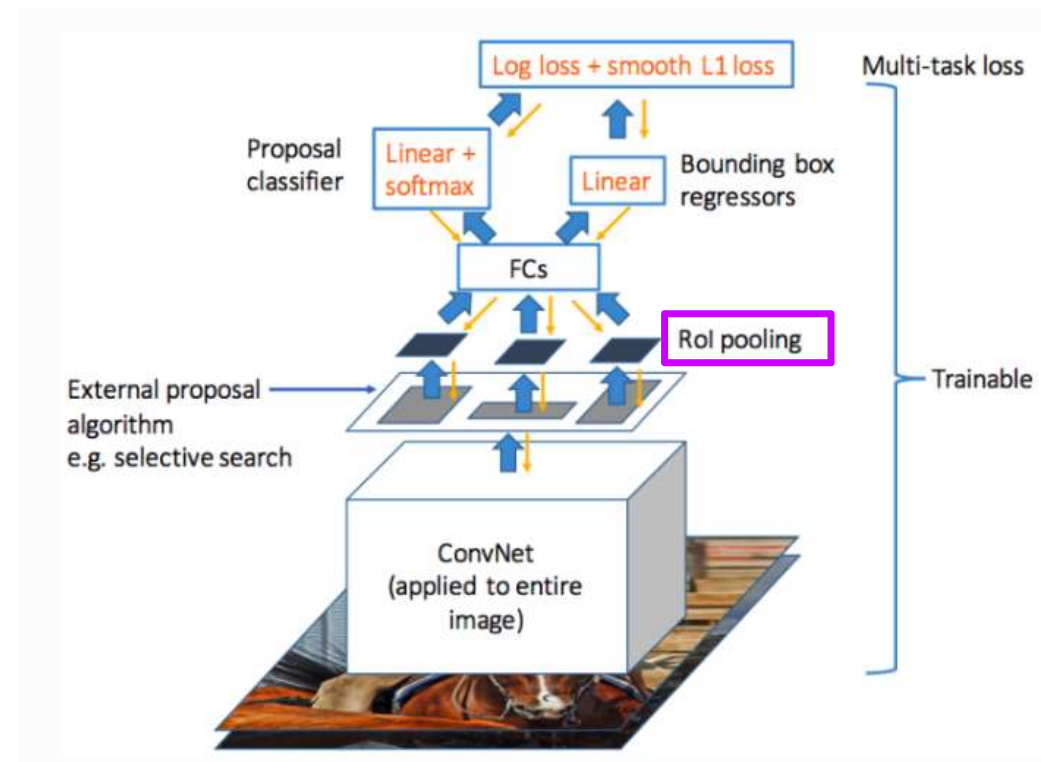
78

## ◆ 구조

### ❖ RoI Pooling

추출한 ROI Feature Map에  
**Max-pooling** 진행하여  
고정된 크기 **pooling map** 생성

랜덤한 크기 가지는 ROI들이  
FC layer에 들어갈 수 있도록  
**고정된 크기를 갖게 됨!**



# Fast R-CNN

Fast Region-based Convolutional Network Method

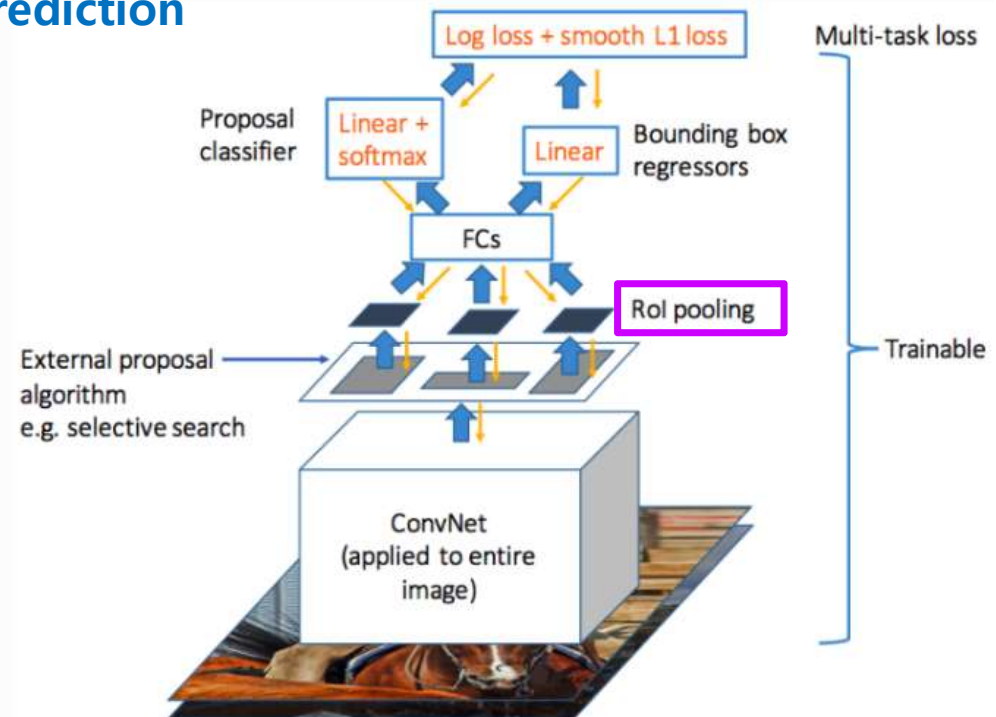
79

## ◆ 구조

### ❖ Classification & Bounding Box Prediction

Feature Map을 FC layer에 통과시켜 Feature Vector 얻음

feature Vector 이용하여  
classification과 bounding box  
prediction 각각 수행



## ◆ 장점과 단점

- **장점**

- 단 하나의 모델을 동작시켜 feature추출과 Bounding Box의 위치, Class 종류 구분 가능

- **단점**

- 객체 후보 영역 탐지 위해 **Region Proposal 사용**하기 때문에 객체 후보영역에 대해 추출 후 학습하는 **multi-pipeline**구조



# Faster R-CNN

Twoards Real-Time Object Dectection with Region Proposal Networks

# Faster R-CNN

Fast Region-based Convolutional Network Method

82

## ◆ 특징

- 2015년 샤오칭 렌이 제안한 방법
- Feature 추출하는 CNN Network와 Class 구분하는 binary SVM, 그리고 Bounding Box를 보정하는 regressor를 **하나의 네트워크로 통합**
- Region Proposal 대신 **객체 후보영역 추정하는 네트워크 RPN 사용**
- **객체 영역 추정 단계의 병목 현상 개선** → VOC 챌린지, COCO 챌린지 우승
- 모델 유연성이 뛰어나 **다른 모델과 조합 통해 성능 높일 수 있음**
- 객체 탐지 분야에서 활발히 사용

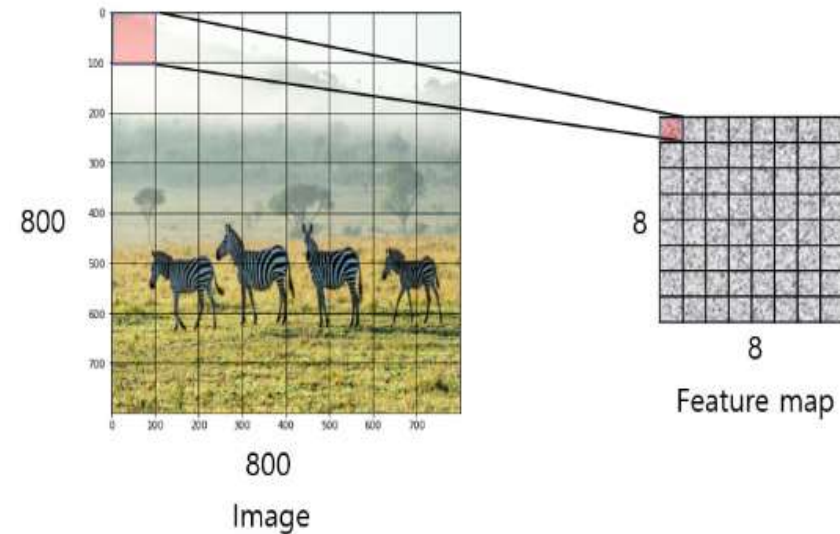
# Faster R-CNN<sub>Fast Region-based Convolutional Network Method</sub>

83

## ◆ 특징

❖ Dense Sampling / Sliding Window → 다양한 크기의 객체 인식 못함!

- 원본 이미지를 일정 간격의 grid로 나눠 각 grid cell을 bounding box로 간주하여 feature map에 encode하는 Dense Sampling 방식 사용
- sub-sampling ratio를 기준으로 grid
- 예) 원본 이미지 크기 800x800  
sub-sampling ratio가 1/100  
CNN 최종 feature map : 8x8(800x1/100)  
8x8개만큼의 bounding box 생성



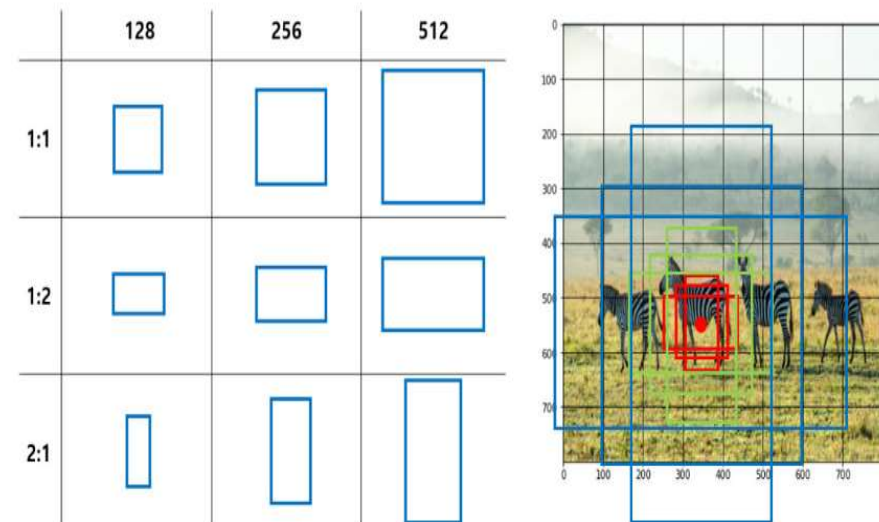
# Faster R-CNN<sub>Fast Region-based Convolutional Network Method</sub>

84

## ◆ 특징

### ❖ Anchor Box

- 지정한 위치에 미리 정의한 서로 다른 크기(scale)와 가로세로비(aspect ratio) 가지는 bounding box 생성 → Anchor Box
- 이미지의 각 grid cell 중심을 기준으로 생성
- 중심점은 고정으로 크기조절
- 다양한 크기의 객체를 포착하는 방법을 제시
- width(w), height(h)
- aspect ratio : width, height 비율












# Faster R-CNN<sub>Fast Region-based Convolutional Network Method</sub>

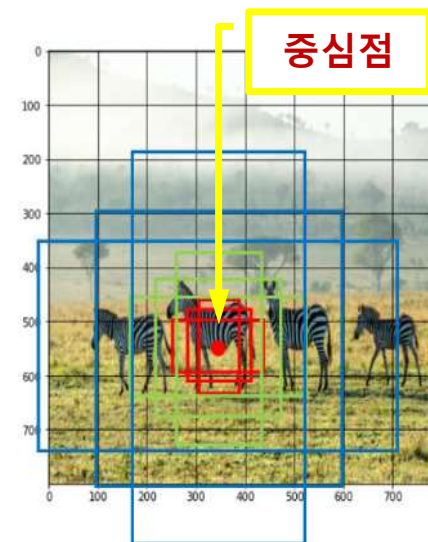
85

## ◆ 특징

### ❖ Anchor Box

- 사전 정의된 9개 Anchor Box
- 원본 이미지 크기 : 600x800
- sub-sampling ratio=1/16
- 생성수 : 1900 (=600/16 x 800/16)
- 총 anchor box 수 : 17100

	128	256	512
1:1			
1:2			
2:1			

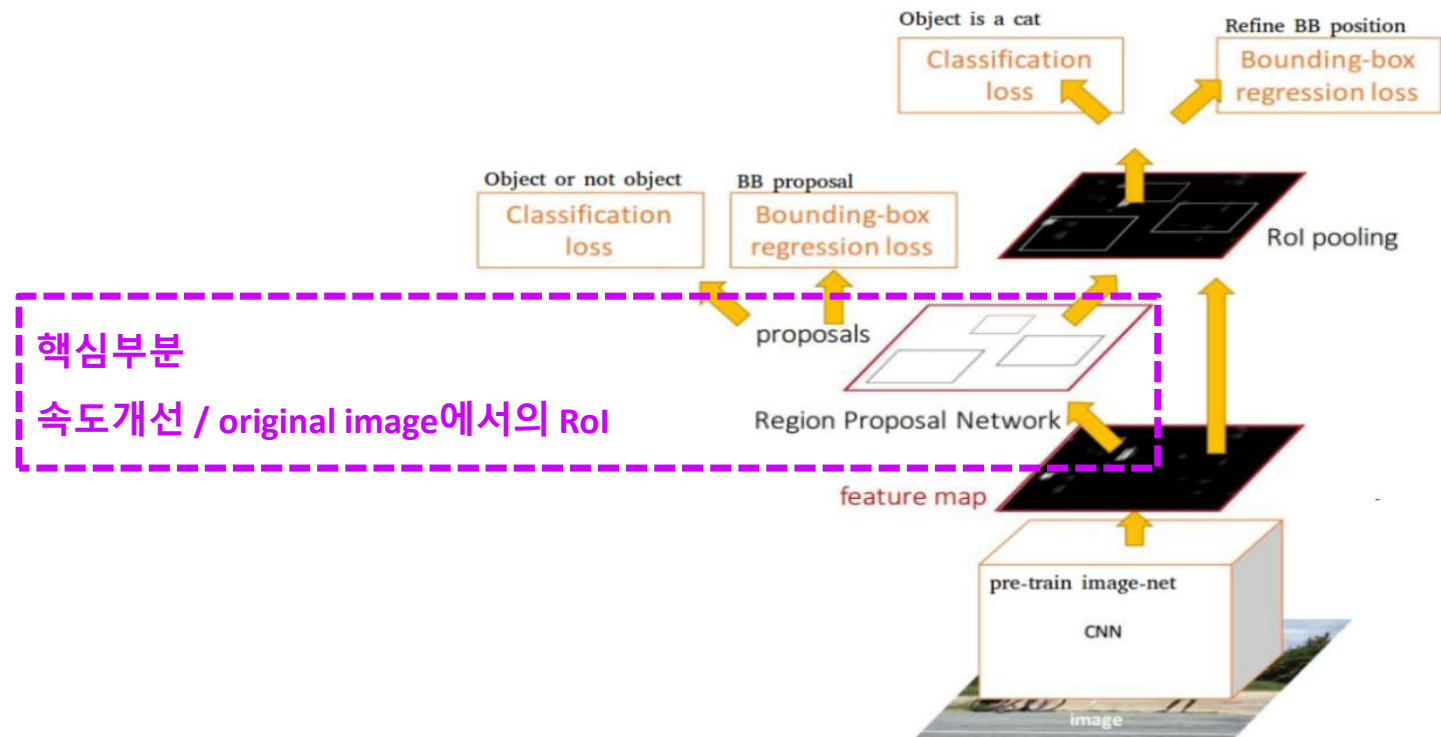


# Faster R-CNN

Fast Region-based Convolutional Network Method

86

## ◆ 구조



## ◆ 구조

### ❖ RPN : 원본 이미지에서 region proposals를 추출하는 네트워크

- 입력 : pre-trained된 VGG 모델에서 추출한 Feature Map
- Region proposal 생성
  - feature map위에  $n \times n$  window를 sliding window
  - object 크기와 비율을 모르므로 k개의 anchor box 미리 정의
  - 가·로세로길이 3종류 x 비율 3종류 = 9개 anchor box 이용
  - NMS(Non Maximum Supression)과정 진행
    - ➔ 가장 높은 score, IOU 일정 이상인 box는 동일한 객체로 판단하여 제거

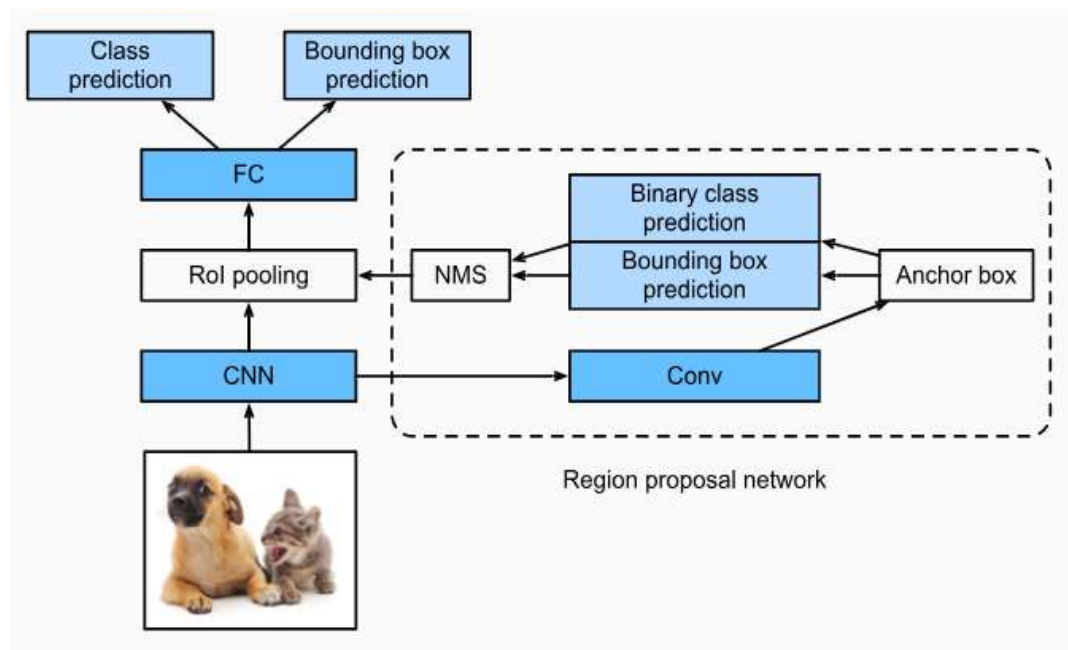
# Faster R-CNN

Fast Region-based Convolutional Network Method

88

## ◆ 구조

❖ RPN : 원본 이미지에서 region proposals를 추출하는 네트워크





# Faster R-CNN

Fast Region-based Convolutional Network Method

89

## ◆ 구조

### ❖ RPN : 원본 이미지에서 region proposals를 추출하는 네트워크

- 입력 : pre-trained된 VGG 모델에서 추출한 Feature Map
- Region proposal 생성
  - feature map위에  $n \times n$  window를 sliding window
  - object 크기와 비율을 모르므로 k개의 anchor box 미리 정의
  - 가·로세로길이 3종류 x 비율 3종류 = 9개 anchor box 이용
  - NMS(Non Maximum Supression)과정 진행
    - ➔ 가장 높은 score, IOU 일정 이상인 box는 동일한 객체로 판단하여 제거