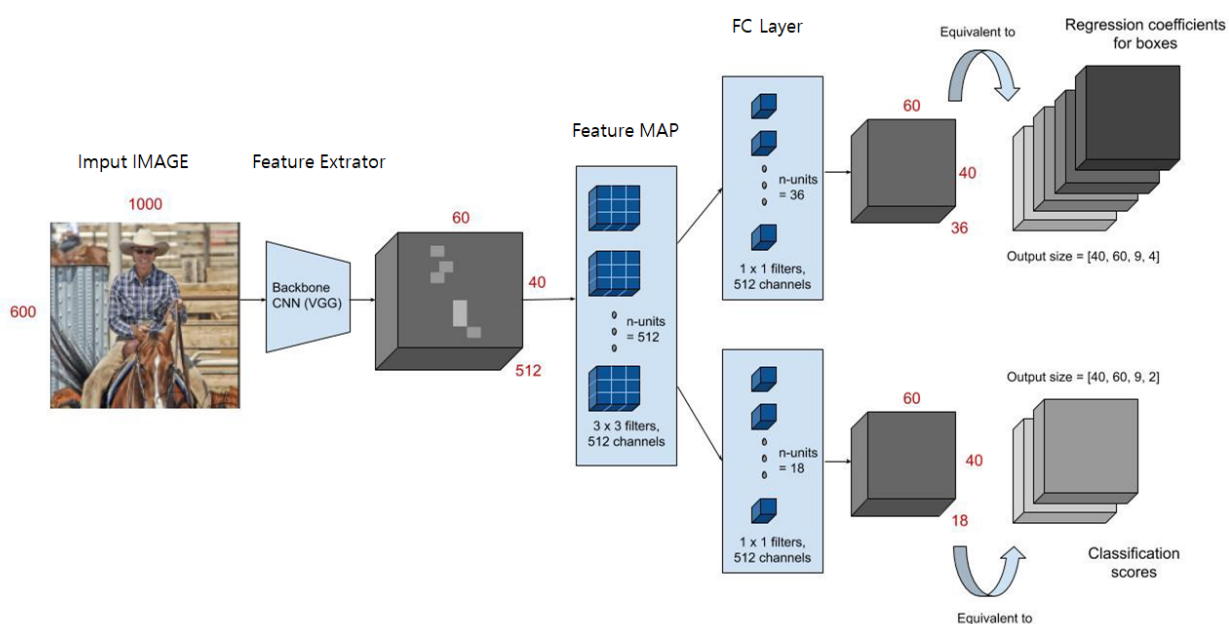


# RCNN

## Object Detection 의 구조

### Object detection의 기본 구조



- bounding box regression 에서는 실제로  $(x, y, w, h)$  Box 의 가운데 좌표인  $(x, y)$ 와 Box의 너비와 높이  $(w, h)$ 를 예측 한다.

### RCNN 계열의 종류

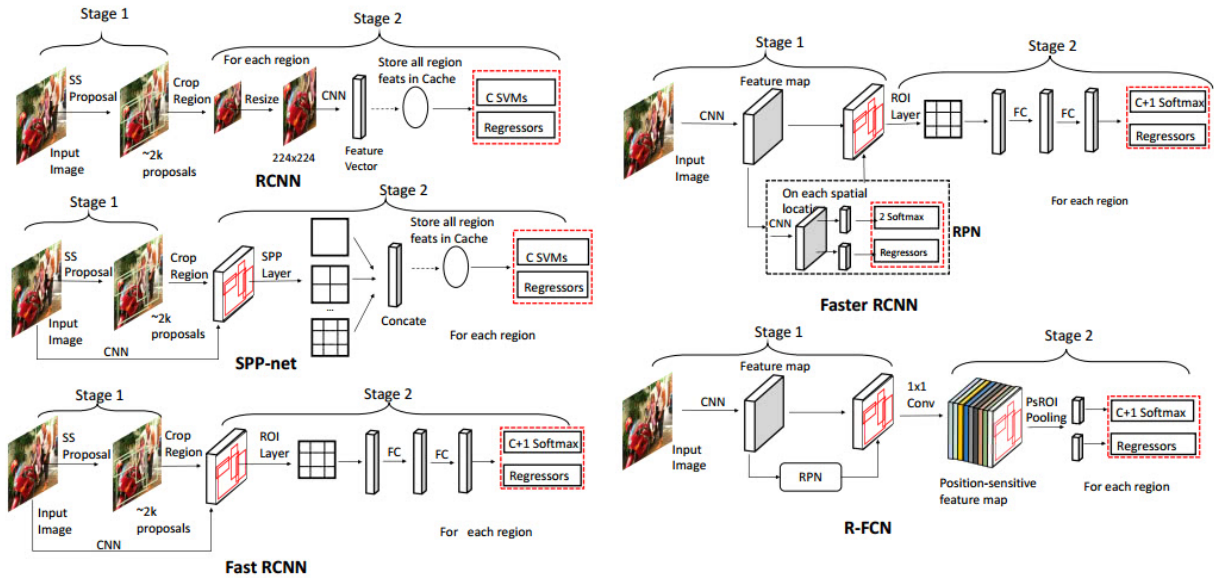
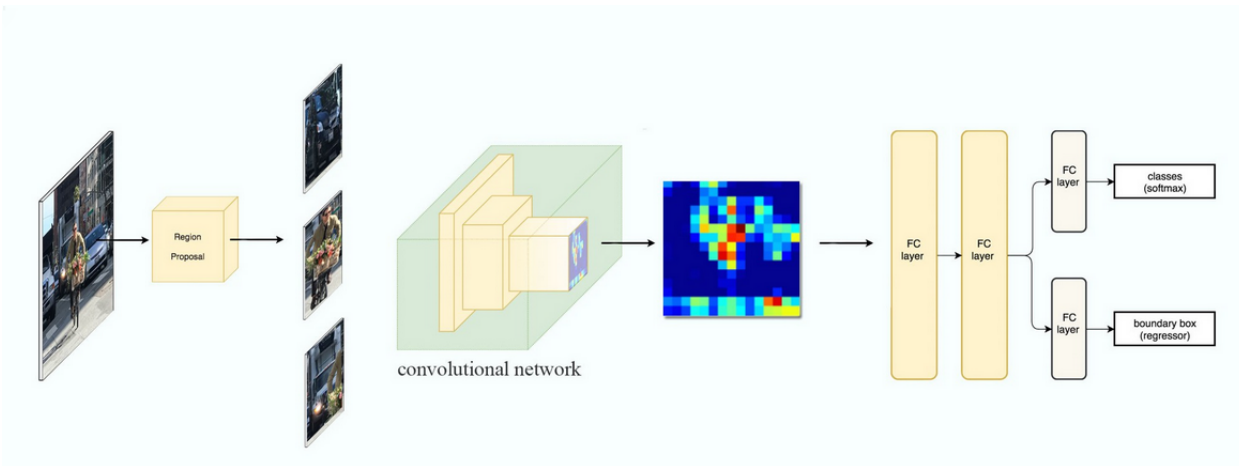


Figure 4: Overview of different two-stage detection frameworks for generic object detection. Red dotted rectangles denote the outputs that define the loss functions.

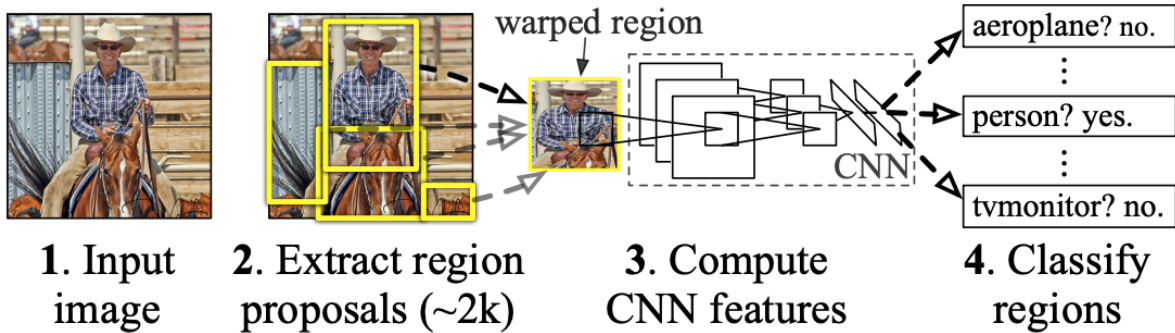
## RCNN



- 기존 CNN 방식에 Region proposal 방식을 더한 알고리즘

## 개요

## R-CNN: *Regions with CNN features*

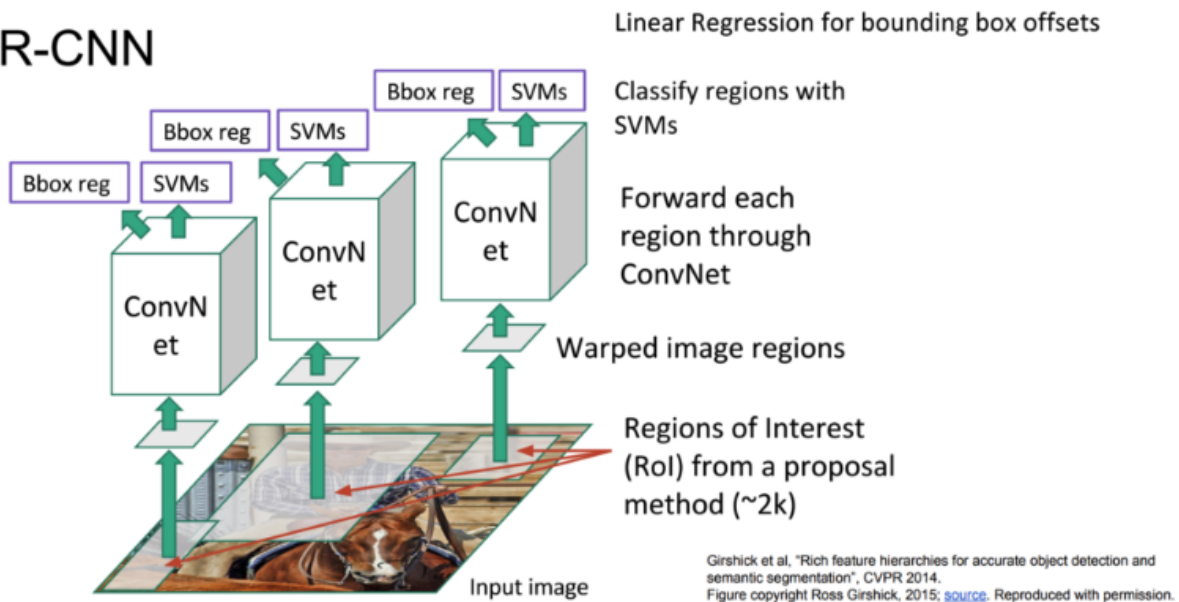


1. 입력 이미지에 Selective Search 알고리즘을 적용하여 물체가 있을만한 박스 2천개를 추출한다.
2. 모든 박스를 잘라내고(crop) 227 x 227로 Resize(warp) 해준다.
3. 미리 Image-Net 데이터를 통해 학습시켜놓은 CNN을 통과시켜 4096 차원의 특징 벡터를 추출한다.
4. 이 추출된 벡터를 가지고 각각의 클래스(사람, 개, 고양이 등등의 Object) 마다 학습시켜 놓은 SVM Classifier를 통과한다.
5. Bounding Box Regression 을 이용하여 박스의 위치를 조정한다.

## Stage 1

### Region Proposal

## R-CNN

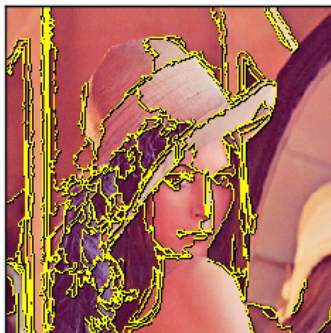


- Selective Search 방법을 이용해 2000개의 Region을 Proposal 함

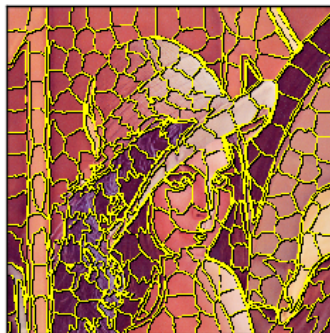
### 1. 작은 크기의 초기 영역을 설정하고(2000개)

- Felzenszwalb's method

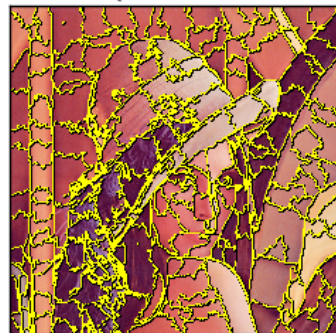
Felzenszwalb's method



SLIC



Quickshift

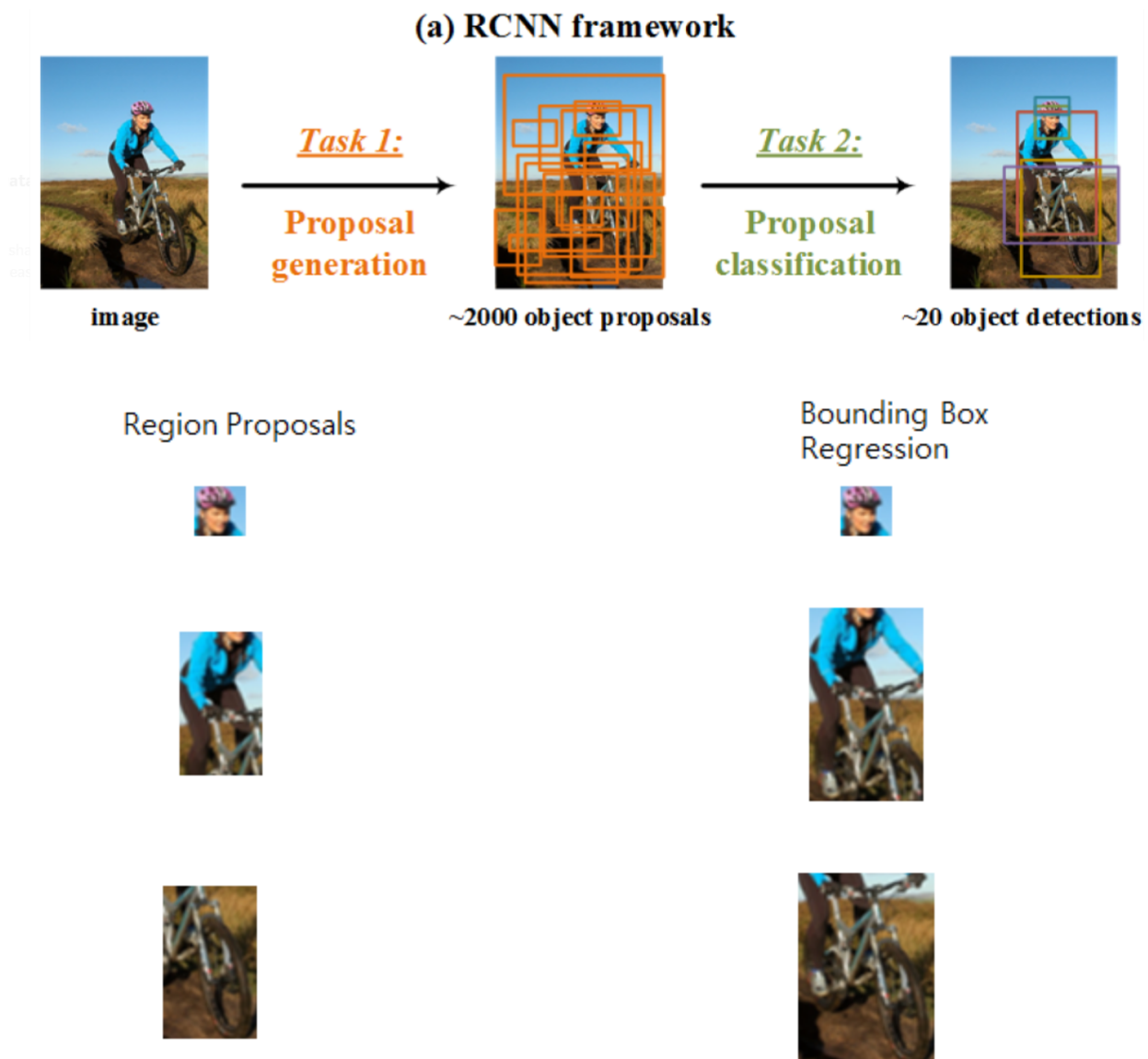


### 2. 하나의 Object로 예측되는 영역을 하나로 병합

- Hierarchical Grouping Algorithm

### 3. ROI(Regions of Interest) 절정

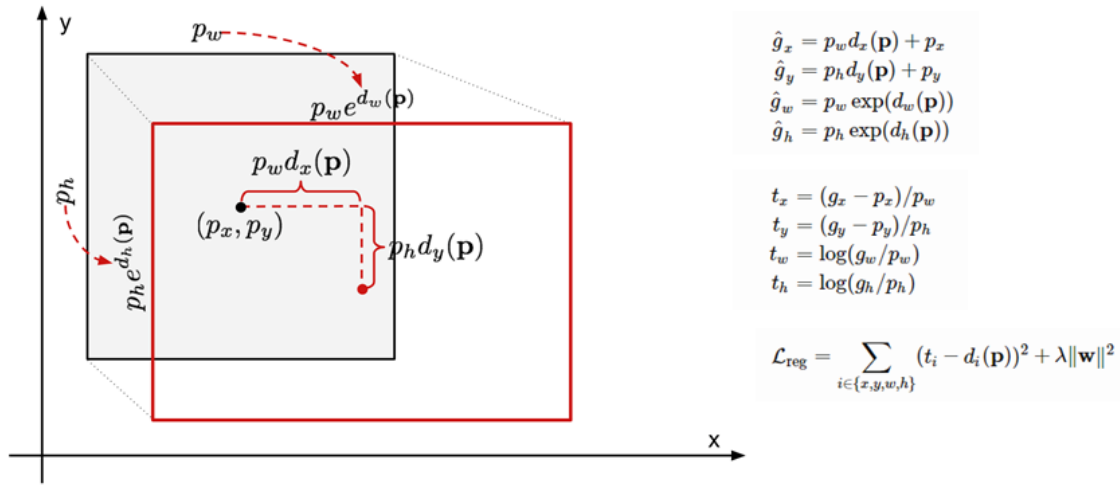
- Region Proposal



## Bounding Box Regression

1. Selective Search를 통해 임의의 Bounding Box 설정
2. 임의의 Bounding Box 와 정답 레이블에 나타난 Bounding Box 의 IOU 값 계산
3. IOU 의 값이 0.5 이상이 되도록 Bounding Box 영역을 조정

- Selective Search 를 통해서 Bounding Box 를 찾아주지만 Bounding Box Regression을 다시하는 이유는 Selective Search가 영역을 정확하게 찾아주지 않기 때문이다.
- <https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html>



$p = (p_x, p_y, p_w, p_h)$  : predicted bounding box coordinate

$g = (g_x, g_y, g_w, g_h)$  : ground truth box coordinates

$\hat{g} = (\hat{g}_x, \hat{g}_y, \hat{g}_w, \hat{g}_h)$  : Optimized Bounding Box

$d(w, p) = (d_x, d_y, d_w, d_h)$  : Transformation Function

- Bounding Box Regression 은  $w$ 값의 최적화를 찾는 것이다.
- 참고 :  $w_* = \underset{\hat{w}_*}{\operatorname{argmin}} \sum_i^N (t_*^i - \hat{w}_*^T \phi_5(p^i))^2 + \lambda \|\hat{w}_*\|^2$

## Stage 2

### CNN Detection

## Region Proposals



## Feature Extraction



## Classifier

얼굴

몸통

자전거

## Warp & Crop

- Selective Search를 통해서 찾아낸 2천개의 박스 영역은 Warp or Crop 을 통해 227 x 227 크기로 리사이즈 된다. (Input 벡터의 크기를 맞추기 위해)
- VGG -16 과 같은 미리 학습되어 있는 CNN 모델을 통과하여 4096 크기의 특징 벡터를 추출. (Image Classification 과정)
- 저자들은 이미지넷 데이터(ILSVRC2012 classification)로 미리 학습된 CNN 모델을 가져온 다음, fine tune하는 방식을 취했습니다.
- fine tune 시에는 실제 Object Detection을 적용할 데이터 셋에서 ground truth에 해당하는 이미지들을 가져와 학습시켰습니다.
- 그리고 Classification의 마지막 레이어를 Object Detection의 클래스 수 N과 아무 물체도 없는 배경까지 포함한 N+1로 맞춰주었습니다.

## R-CNN 정리

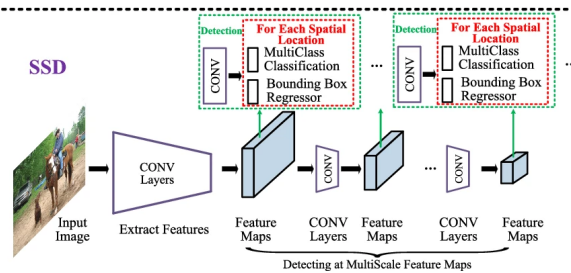
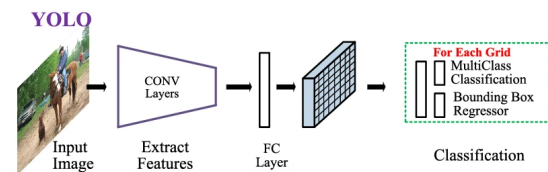
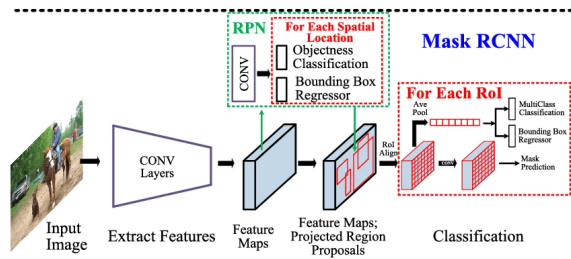
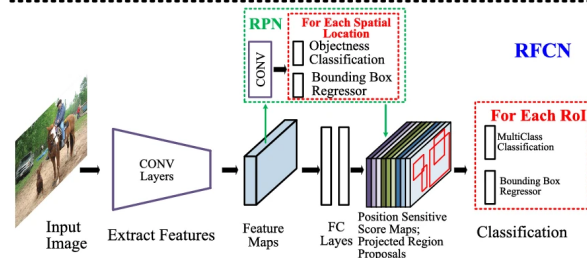
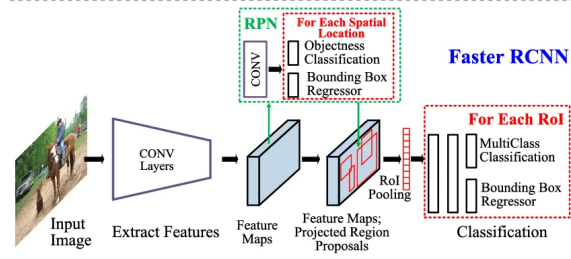
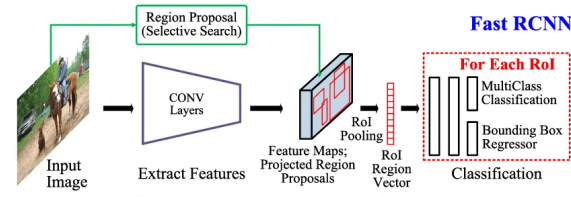
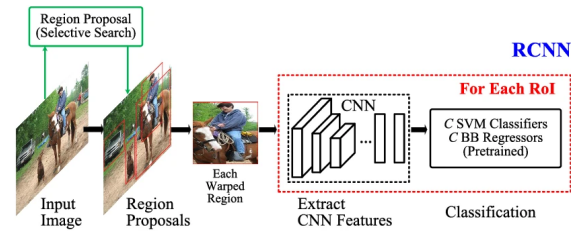
- 매우 높은 Detection 정확도
- 서로 다른 종류의 알고리즘이 섞여있음.
- 1장의 이미지당 2000개의 Region 을 도출하여 CNN Feature MAP 생성

- 한장당 약 30초 소요
- Selective Search가 CPU 를 사용

Recent Advances in Deep Learning for Object Detection - Part 1

<https://www.dlology.com/blog/recent-advances-in-deep-learning-for-object-detection/>





Reducing the Amount of Per Region Computation; Making the Training Process Closer to End to End

<<https://link.springer.com/article/10.1007/s11263-019-01247-4/figures/13>>

<https://medium.com/@prvnk10/object-detection-rcnn-4d9d7ad55067>

RCNN 참고자료