Ailab Seminar #14

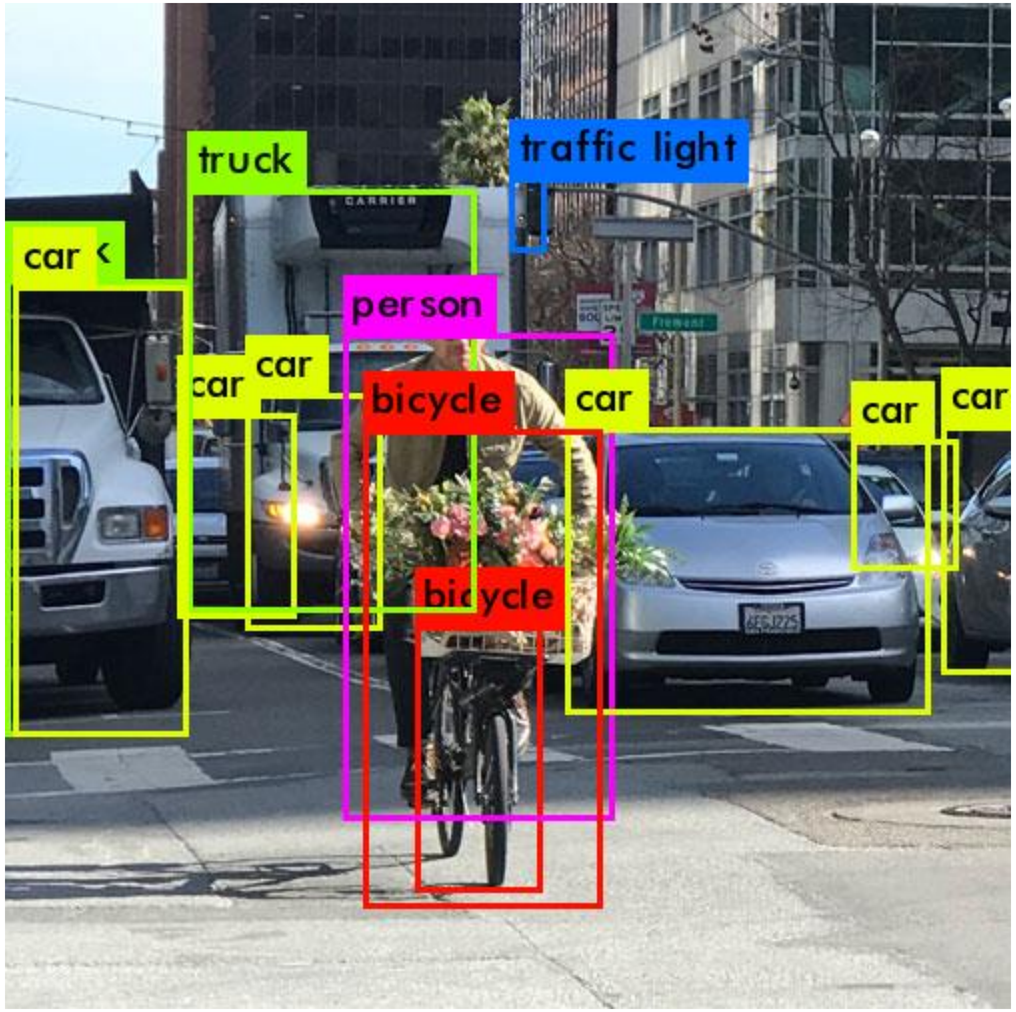
R-CNN vs. YOLO

최원혁

R-CNN vs. YOLO

Abstract

- Task
  - ✓ Object detection



R-CNN vs. YOLO

Abstract

2013 ICCV 2015 NIPS 2015 ICCV 2017

RCNN - Fast RCNN - Faster RCNN - Mask RCNN

2-Stage Detector

1-Stage Detector

anchor based	non anchor based
CVPR 2016 ECCV 2016 ICCV 2017	ECCV 2018 2019 2019
YOLO (v1, v2, v3) SSD - RetinaNet	CornerNet - ExtremeNet - CenterNet

R-CNN series

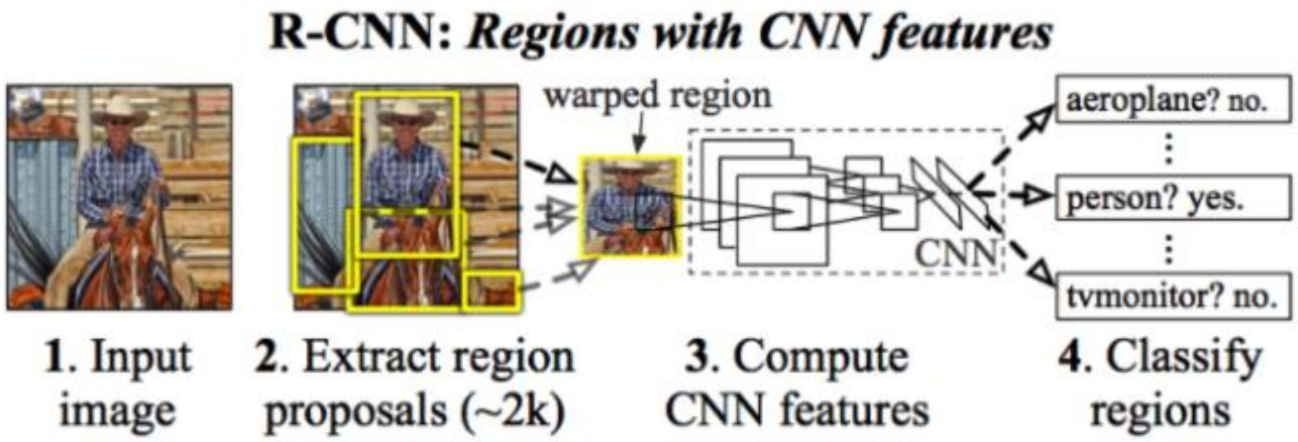
2 stage detector

R-CNN vs. YOLO

2 stage detector

2 stage detector : Regional Proposal과 Classification이 순차적으로 이루어지는 구조

Regional Proposal : 물체가 있을만한 영역을 찾아내는 것



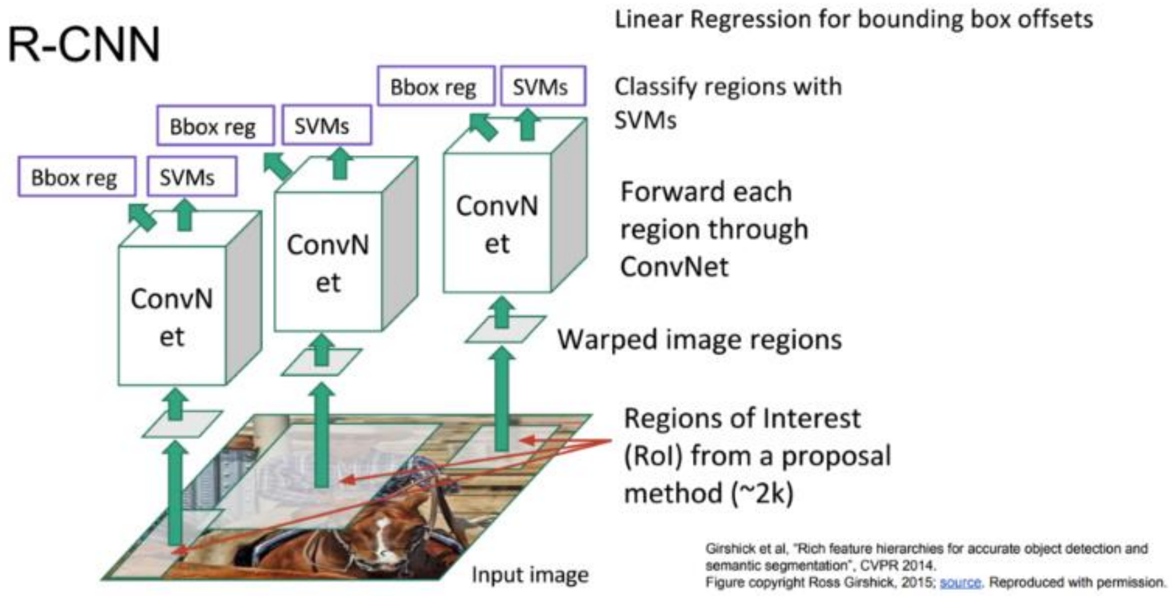
R-CNN vs. YOLO

2 stage detector

	2013		ICCV 2015		NIPS 2015		ICCV 2017
발전 흐름	RCNN	-	Fast RCNN	-	Faster RCNN	-	Mask RCNN

R-CNN vs. YOLO

2 stage detector



Steps

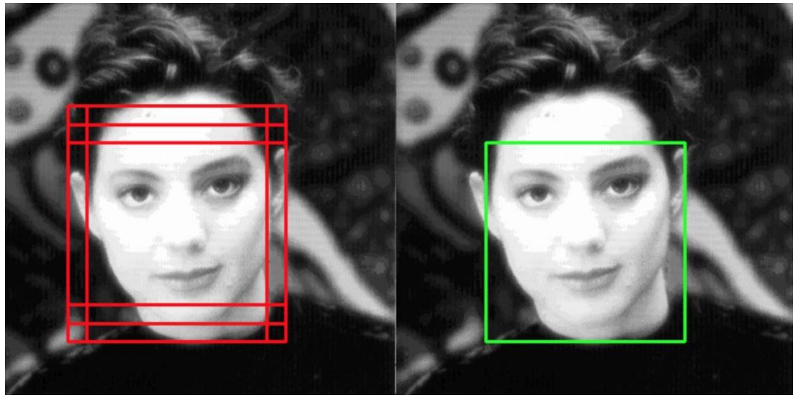
1. Input Image에 Selective Search 알고리즘을 적용하여 물체가 있을만한 박스 2천개를 추출한다.
2. 모든 박스를 227 x 227 크기로 리사이즈(wrap) 한다.
3. CNN Network를 통과시켜 4096 차원의 feature vector를 추출한다.
4. 각각의 클래스마다 학습된 SVM Classifier에 벡터를 넣는다.
5. Bounding Box Regression을 적용하여 박스의 위치를 조정한다.

Region Proposal – Selective Search Algorithm

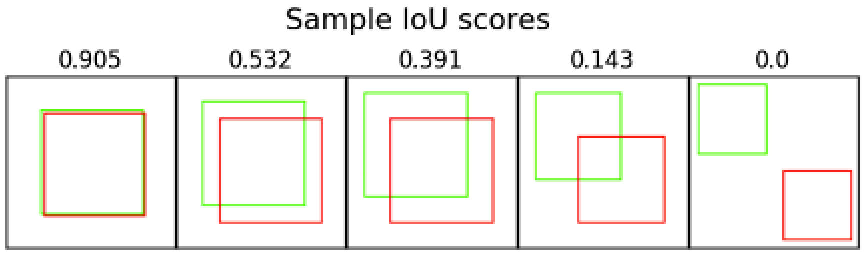
- Bounding box들을 찾아주는 super pixel 기반의 hierarchical grouping algorithm
- 유사성이 높은 region들을 합쳐 나감
- Color, Texture, Size, Fill 들의 가중합으로 유사도를 구함

Non-Maximum Suppression

- 동일한 물체의 여러 박스를 스코어가 가장 높은 박스를 제외하고 제거



- IoU(Intersection over Union)이 0.5보다 크면 동일한 물체의 대상이라고 판단



Bounding Box Regression

- Selective search로 찾은 박스의 위치를 교정
- 하나의 박스를 다음과 같이 표기

$$P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$$

- GT 박스를 다음과 같이 표기

$$G = (G_x, G_y, G_w, G_h).$$

- x, y, w, h를 이동시켜주는 함수

$$d_x(P),\ d_y(P),\ d_w(P),\ \text{and}\ d_h(P).$$

- P를 이동시키는 함수의 식

$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

- D의 인자로 CNN network에서 추출한 feature vector 사용

$$d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P).$$

- Loss, 람다=1000

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2$$

$$t_x = (G_x - P_x)/P_w$$
$$t_y = (G_y - P_y)/P_h$$
$$t_w = \log(G_w/P_w)$$
$$t_h = \log(G_h/P_h).$$

R-CNN vs. YOLO

2 stage detector

	Region Proposal	Classification
R-CNN	Selective Search	SVM
Fast R-CNN	?	?
Faster R-CNN	?	?

R-CNN vs. YOLO

2 stage detector

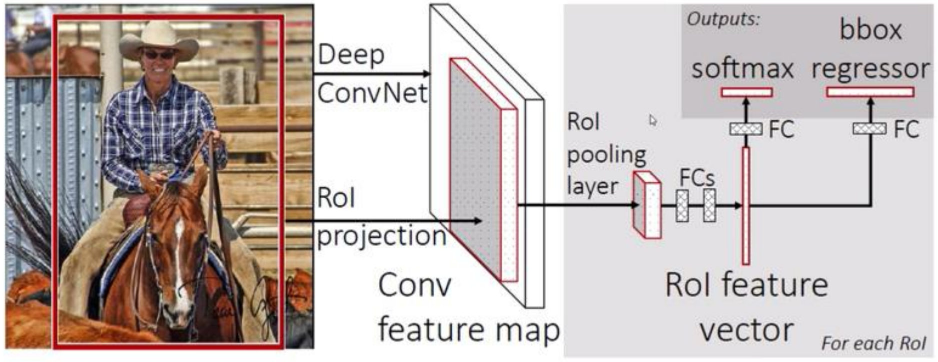
R-CNN의 한계 : 너무 느리다

- ✓ 13s - GPU
- ✓ 54s - CPU

↓

Fast R-CNN

Fast R-CNN



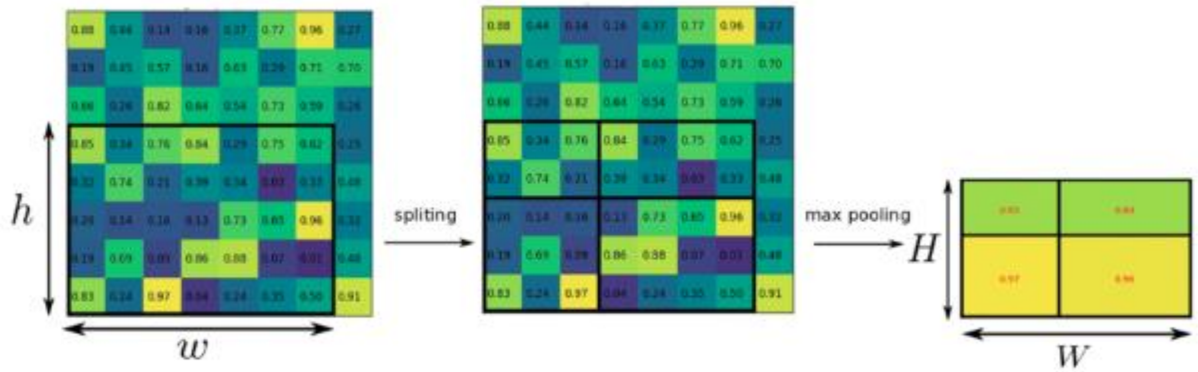
CNN Feature 추출부터 Classification, Bounding box regression까지 모두 하나의 모델로 학습시키자

Steps

1. 전체 이미지를 미리 학습된 CNN Network로 Feature Map 추출
2. Selective Search를 통해서 찾은 각각의 RoI에 Pooling 진행. 고정된 크기의 feature vector 획득
3. Fully connected layer들을 통과한 뒤, 두 개의 branc로 나뉨
4. 하나는 softmax를 통과하여 해당 RoI의 classification 진행
5. 나머지는 Bounding Box Regression을 통해서 박스의 위치를 조정

RoI Pooling

- 임의 크기의 Region Proposal을 정해진 output size로 만들기 위한 Pooling layer
- Split = h/H x w/W



input image and region proposal   pooling section   result

Fast R-CNN

Multi-task Loss

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v),$$

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

in which

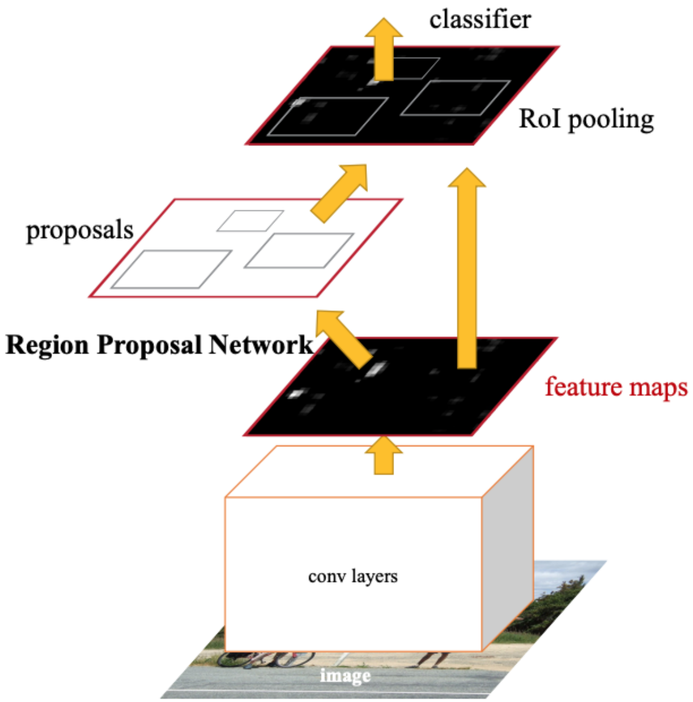
$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

R-CNN vs. YOLO

2 stage detector

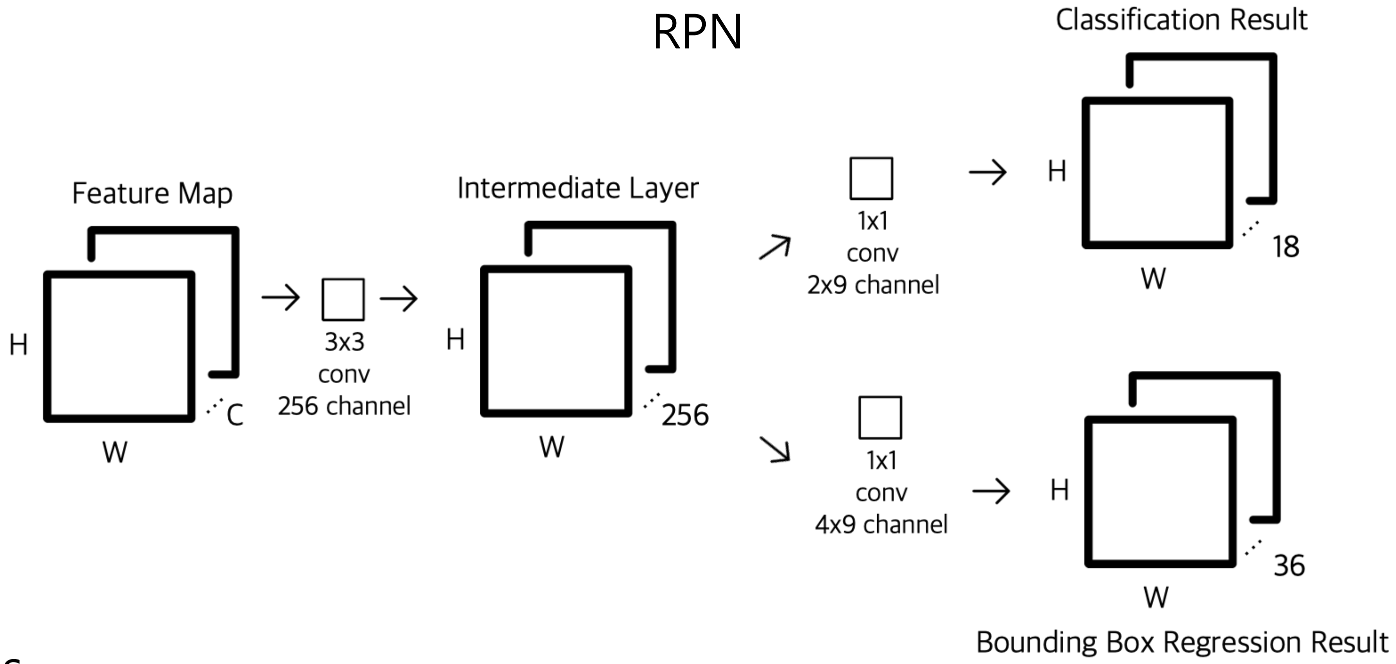
	Region Proposal	Classification
R-CNN	Selective Search	SVM
Fast R-CNN	Selective Search	Softmax Layer
Faster R-CNN	?	?

Faster R-CNN



Selective search algorithm을 Region Proposal Network로 대체하자

RPN



Steps

1. CNN Network를 통해서 뽑아낸 feature map을 3x3 Conv을 통해 Intermediate Layer을 만든다.
2. 1x1 Conv를 통해서 Classification과 Bounding Box Regression 값을 계산한다.
3. Classification = 2(yes or no) x anchor 개수
4. Bounding Box Regression = 4(x, y, w, h) X anchor 개수
5. Classification을 통해서 얻은 물체의 확률 값을 정렬하고 높은 순으로 K개의 앵커 뽑음
6. K개의 앵커에 Bounding Box Regression을 적용
7. Non-Maximum Suppression을 적용하여 RoI 구함

Faster R-CNN

Multi-task Loss

$$L(\{p_i\},\{t_i\}) = \underbrace{\frac{1}{N_{cls}}}_{\text{Minibatch 사이즈}} \sum_{\underbrace{i}_{\text{anchor}}} L_{cls}(p_i, p_i^*) + \lambda \underbrace{\frac{1}{N_{reg}}}_{\text{전체 anchor 개수}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

R-CNN vs. YOLO

2 stage detector

	Region Proposal	Classification
R-CNN	Selective Search	SVM
Fast R-CNN	Selective Search	Softmax Layer
Faster R-CNN	RPN	Softmax Layer

R-CNN vs. YOLO

2 stage detector

Evaluation

System	Time	07 data	07 + 12 data
R-CNN	~ 50s	66.0	-
Fast R-CNN	~ 2s	66.9	70.0
Faster R-CNN	**~ 198ms**	**69.9**	**73.2**

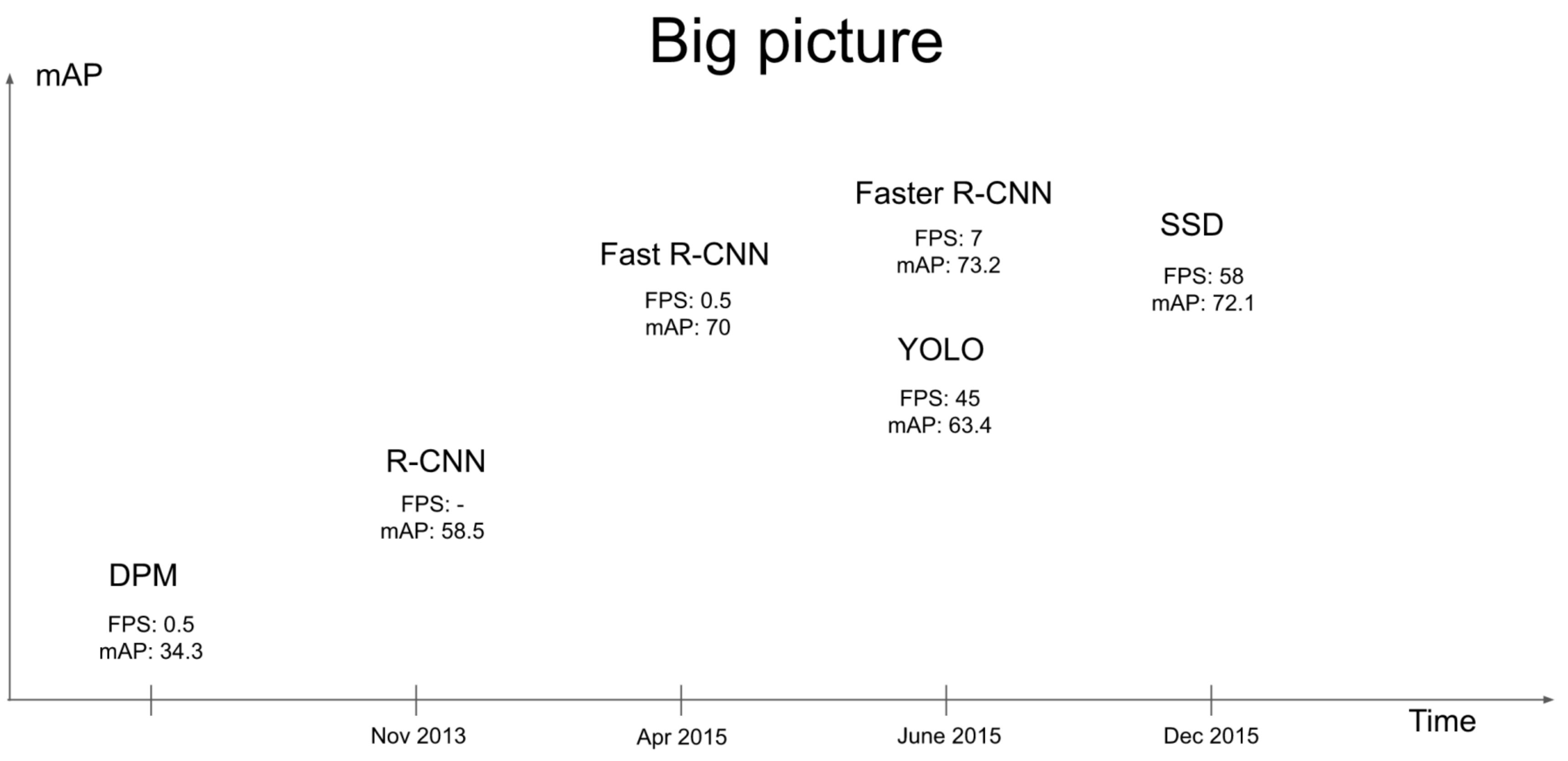
Detection $\boxed{\text{mAP}}$ on PASCAL VOC 2007 and 2012, with VGG-16 pre-trained on ImageNet Dataset

YOLO series

1 stage detector

R-CNN vs. YOLO

1 stage detector

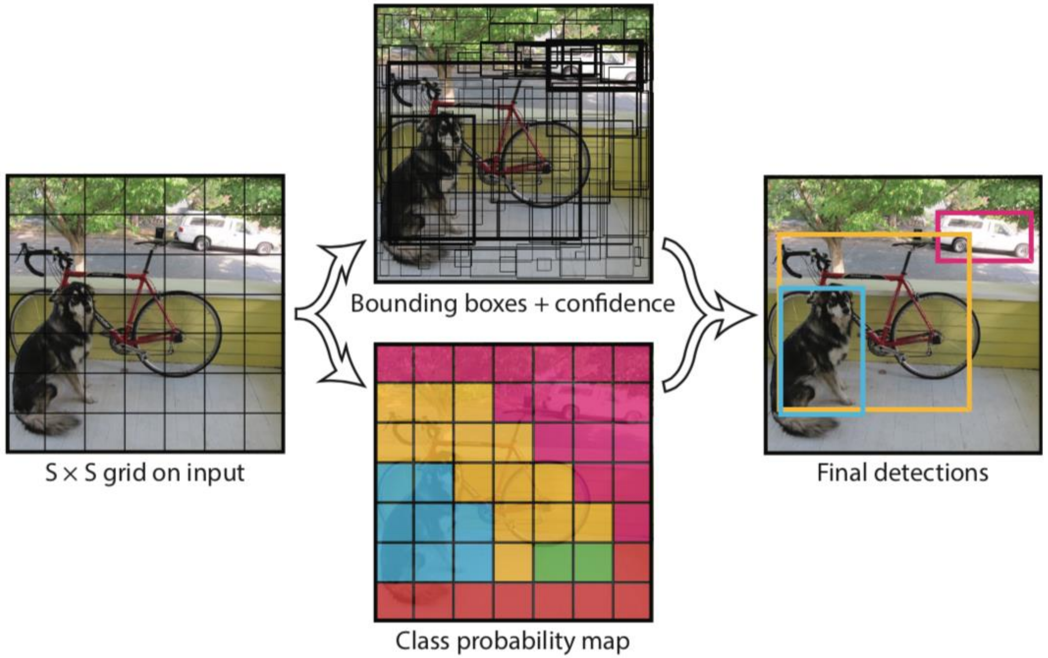


Результаты на тестовой выборки Pascal VOC 2007. Обучение на trainval sets 2007+2012

deepsystems.io 2

R-CNN vs. YOLO

1 stage detector



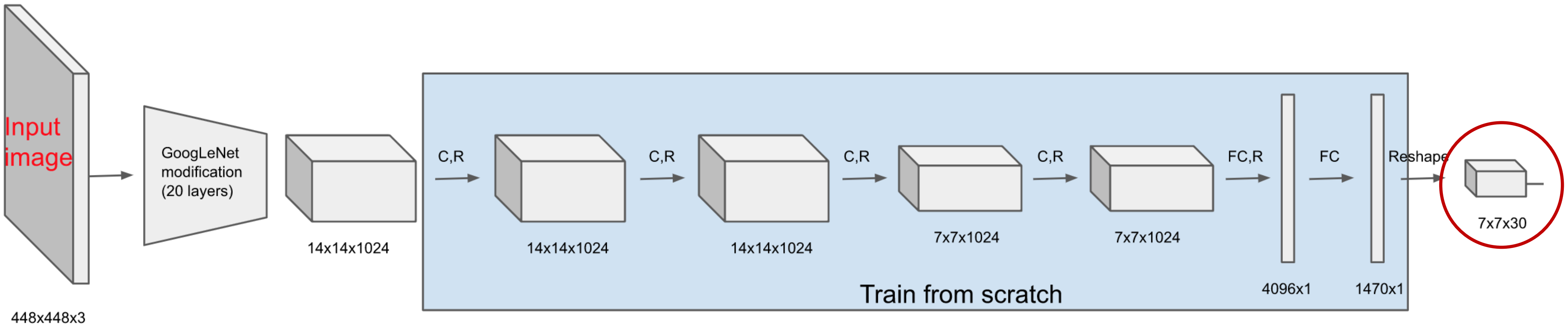
Steps

1. 입력 이미지를 S x S 그리드 영역으로 나눈다
2. 각 그리드 영역에서 물체가 있을만한 영역에 해당하는 B개의 Bounding Box를 예측 (x, y, w, h)
3. 박스의 신뢰도를 나타내는 Confidence를 계산. Pr(Object) X IoU

R-CNN vs. YOLO

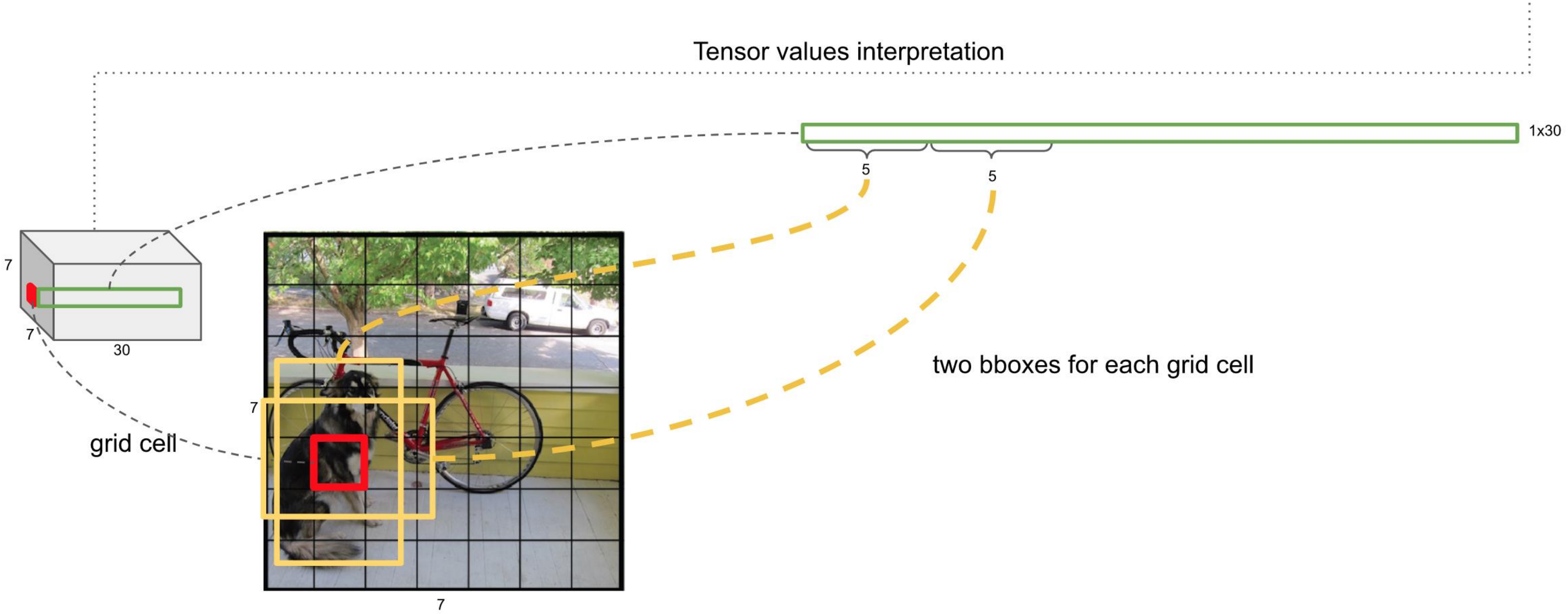
1 stage detector

YOLO Network



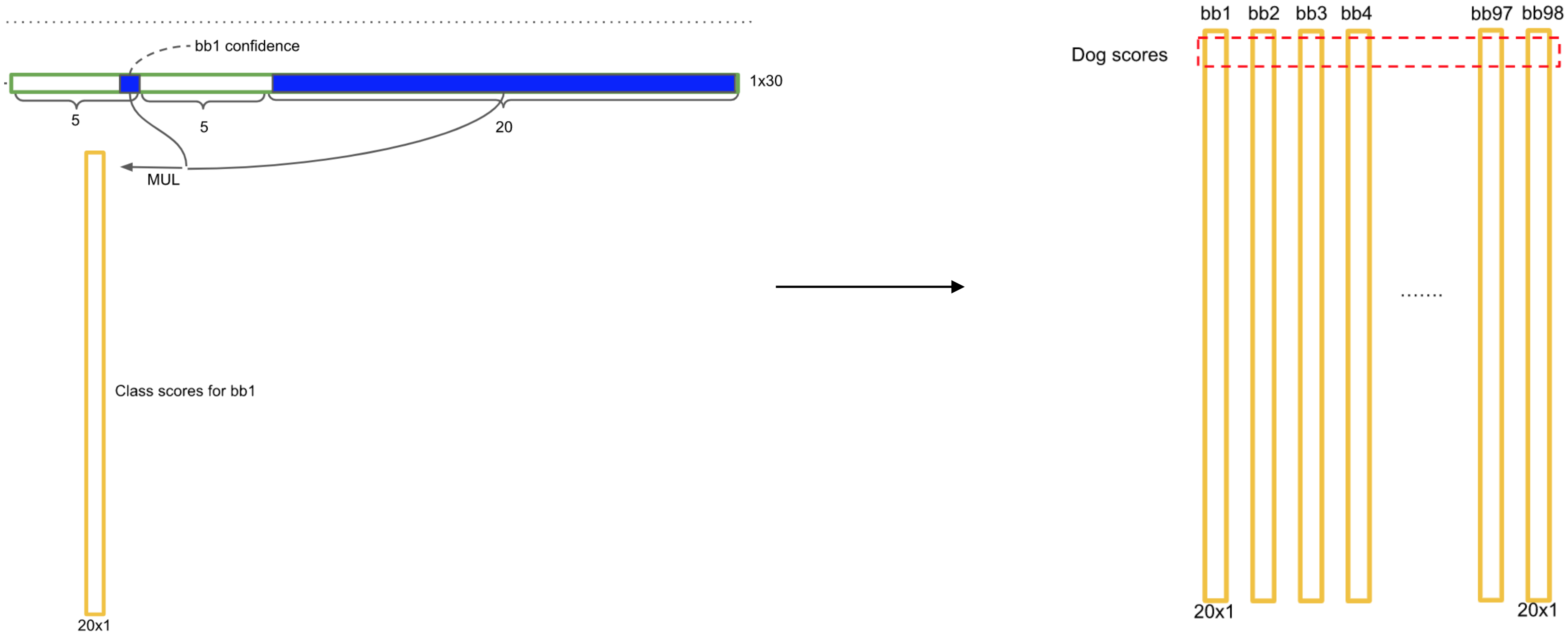
R-CNN vs. YOLO

1 stage detector



- 7x7은 이미지의 그리드를 의미
- 30차원의 벡터값
  - ✓ 앞의 10차원 값 : 2개의 Bounding Box 값(x, y, w, h, C), 2개의 Bounding Box 값은 hyperparameter
  - ✓ 다음 20차원 값 : Class 확률 값. 20개의 Class.

1 stage detector

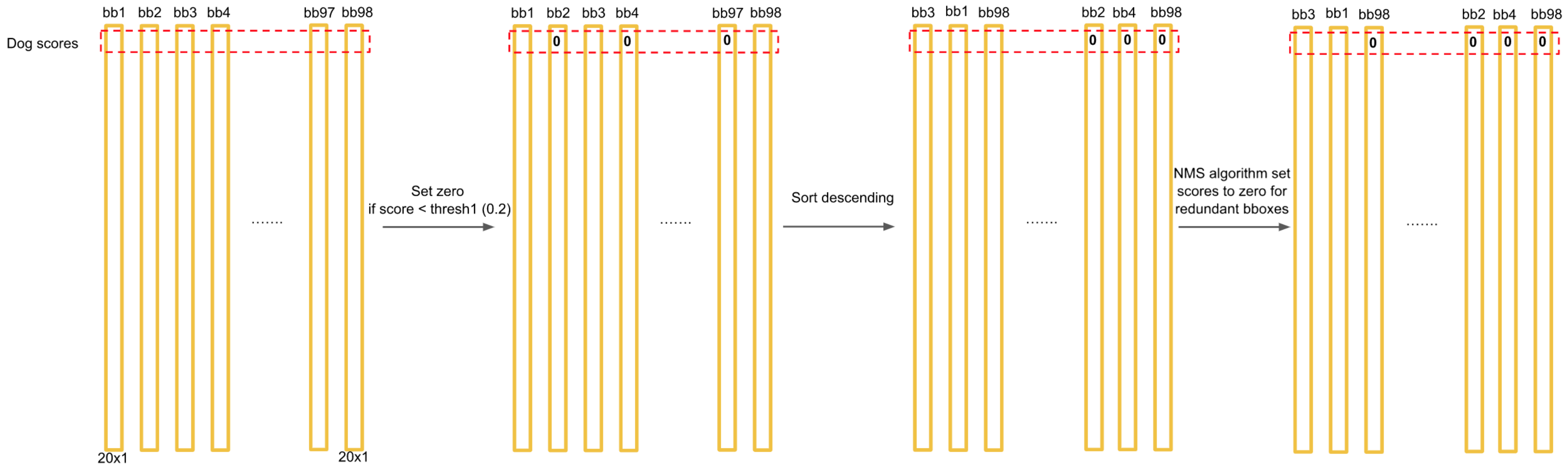


Bounding Box confidence value * Class = 해당 박스의 특정 클래스 확률

7x7x2 = 98

R-CNN vs. YOLO

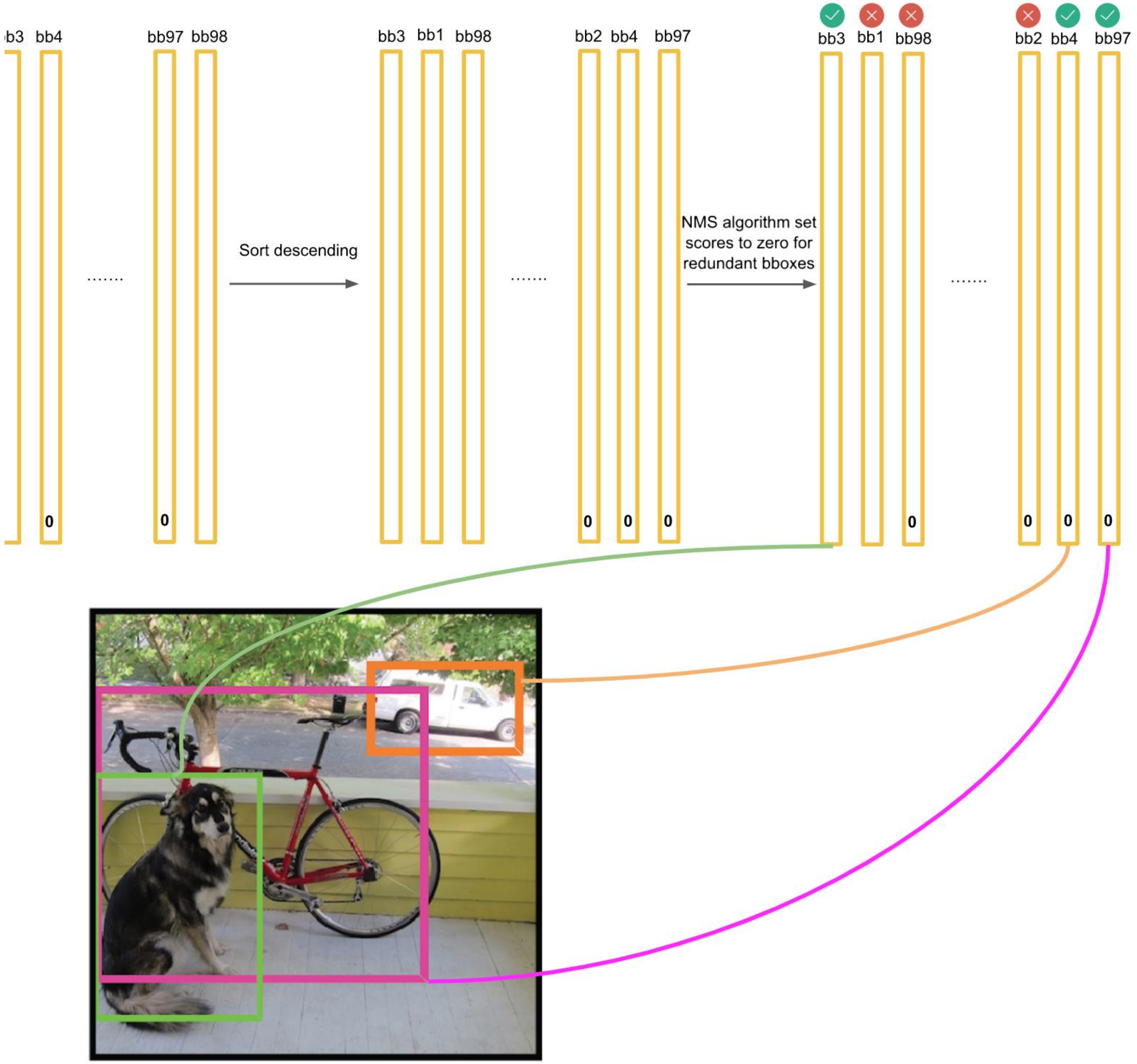
1 stage detector



NMS : non-Maximum supperssion

R-CNN vs. YOLO

1 stage detector



YOLO Loss

Prediction 된 i 인덱스의 j번째 bounding box

논문에서 5로 설정

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

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

W, h는 비율 값이기 때문에 root 사용

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

Object를 검출못한 i 인덱스의 j BB

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

모든 물체가 있다고 판단된 인덱스 i

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad (3)$$

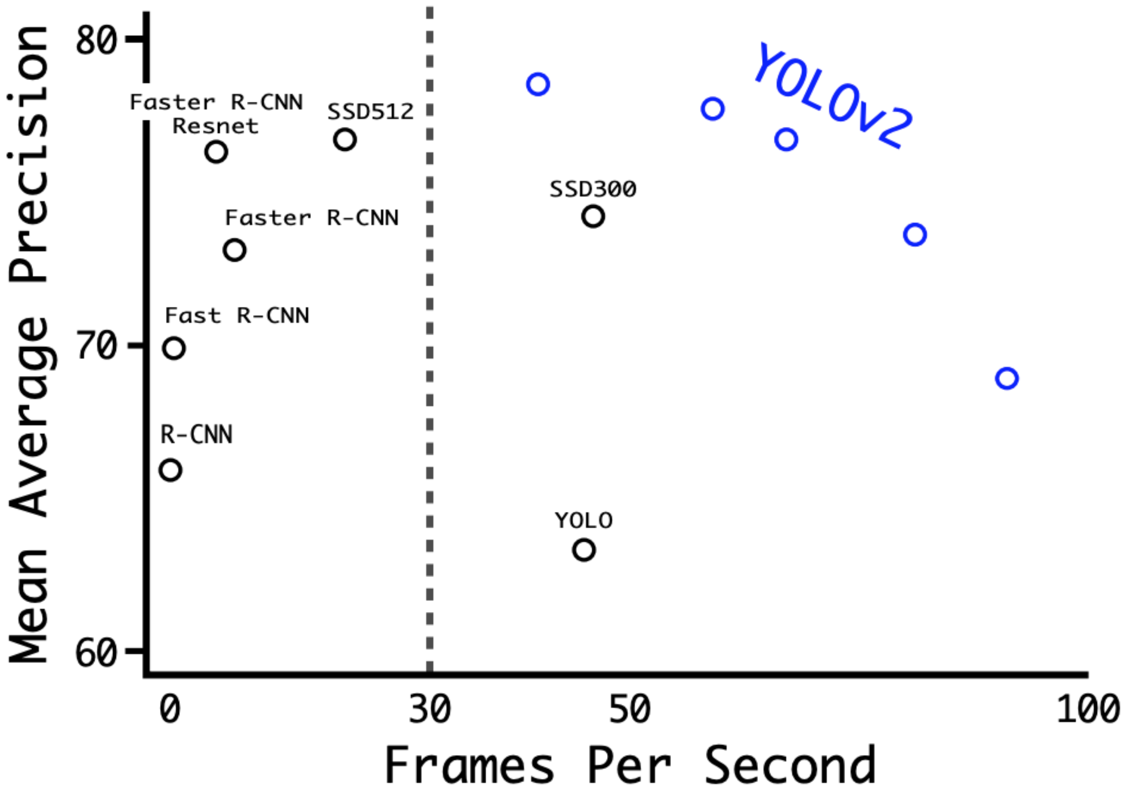
YOLO v2

- Batch Normalization 적용
- 높은 해상도 이미지로 백본 CNN 네트워크 fine tune
- Anchor Box 개념 적용하여 학습 안정화
- 높은 해상도의 feature map을 낮은 해상도 feature map에 합치기

R-CNN vs. YOLO

1 stage detector

evaluation



Reference

- 갈아먹는 머신러닝, https://yeomko.tistory.com/category/%EA%B0%88%EC%95%84%EB%A8%B9%EB%8A%94%20Object%20Detection
- Deepsystems.io, https://docs.google.com/presentation/d/1aeRvtKG21KHdD5lg6Hgyhx5rPq_ZOsGjG5rJ1HP7BbA/pub?start=false&loop=false&delayms=3000&slide=id.p
- 프라이데이, https://ganghee-lee.tistory.com/35
- Hyu-ailab-ai-seminar, https://github.com/HYU-AILAB/ai-seminar/blob/master/season_10/02.%20Feature%20Pyramid%20Networks%20for%20Object%20Detection/191210_FPN_%EC%A0%95%EC%A7%80%EC%9D%80.pdf
- Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
- Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2015.
- Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.
- Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

End