R-CNN, Fast R-CNN, Mask R-CNN

2021 Aug 13th

KNOWLEDGE INNOVATION RESEARCH CENTER

SungRae Hong

Journey

1. R-CNN
   A. Overview
   B. Architecture
   C. Selective Search
   D. Bounding Box Regression

2. Fast R-CNN
   A. Shortages of R-CNN
   B. Overview
   C. RoI Pooling

3. Faster R-CNN
   A. Architecture
   B. RPN(Region Proposal Network)
   C. NMS(Non Maximum Suppression)

Summary

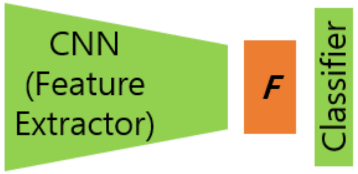
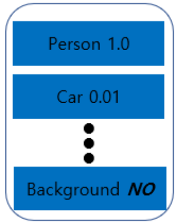
+Reference

R-CNN

(Girshick et al., CVPR 2014)

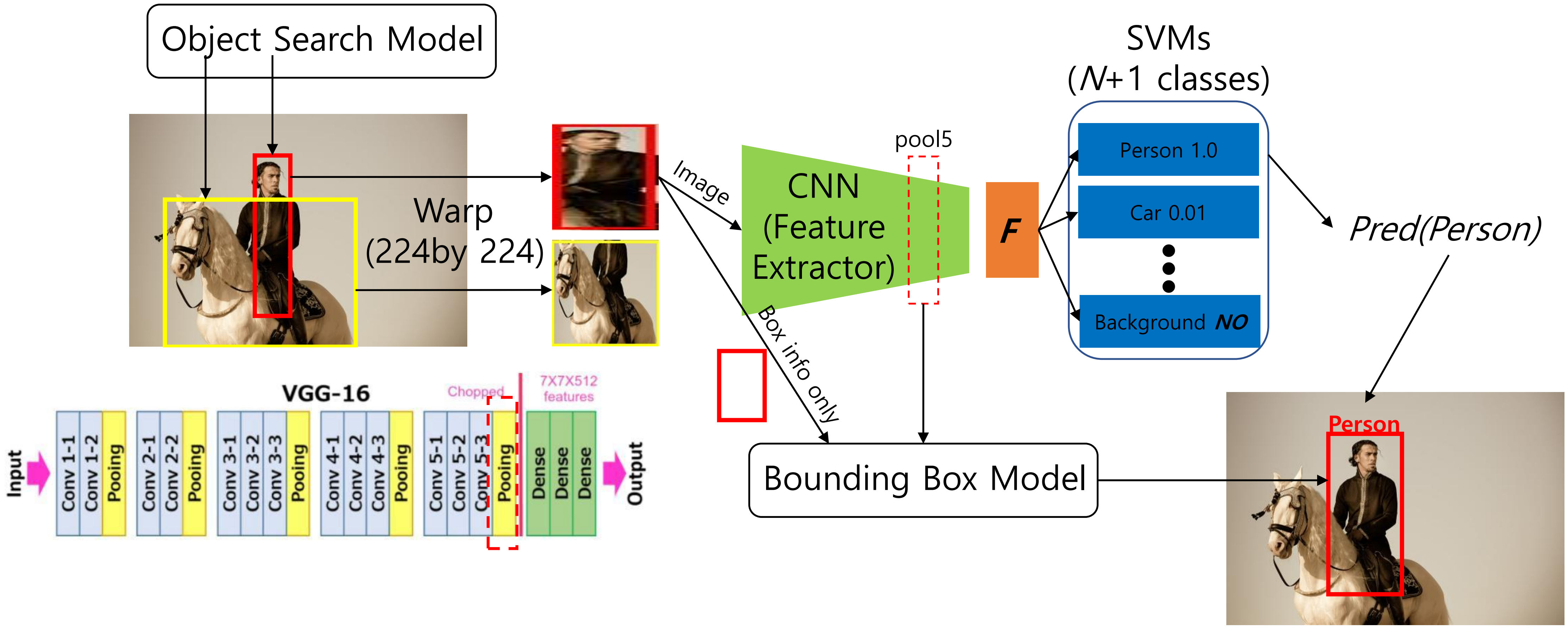
1. R-CNN

A. Overview

Model	Input	Target	Remarks
	Pretrain : ImageNet2013	Pretrain : Classification label	Remove *Classifier* when fine-tuned
	Fine-tuning : Detection dataset	Fine-tuning : Detection annotations	
Bounding Box Model	Recommanded region(patch) and bounding boxes	Ground truth bounding box(IoU=1)	Label same as GT if IoU is greater than 0.5 else label 'Background'
SVMs (*N*+1 classes) 	Feature vector from CNN Layer	Ground truth class	

1. R-CNN

B. Architecture



https://prateekvjoshi.com/2015/12/15/how-to-compute-confidence-measure-for-svm-classifiers/

1. R-CNN

C. Selective Search

Algorithm 1: Hierarchical Grouping Algorithm

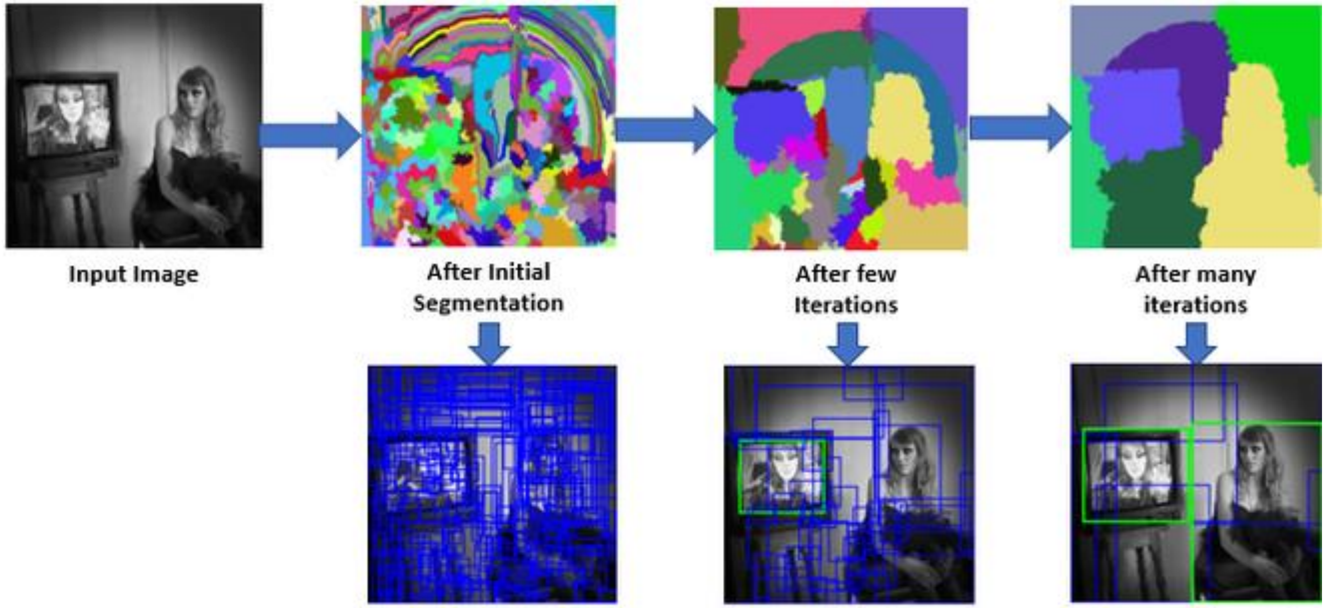
Input: (colour) image
Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \cdots, r_n\}$ using [13]
Initialise similarity set $S = \emptyset$
foreach *Neighbouring region pair* (r_i, r_j) **do**
    Calculate similarity $s(r_i, r_j)$
    $S = S \cup s(r_i, r_j)$

while $S \neq \emptyset$ **do**
    Get highest similarity $s(r_i, r_j) = \max(S)$
    Merge corresponding regions $r_t = r_i \cup r_j$
    Remove similarities regarding $r_i : S = S \setminus s(r_i, r_*)$
    Remove similarities regarding $r_j : S = S \setminus s(r_*, r_j)$
    Calculate similarity set S_t between r_t and its neighbours
    $S = S \cup S_t$
    $R = R \cup r_t$

Extract object location boxes L from all regions in R

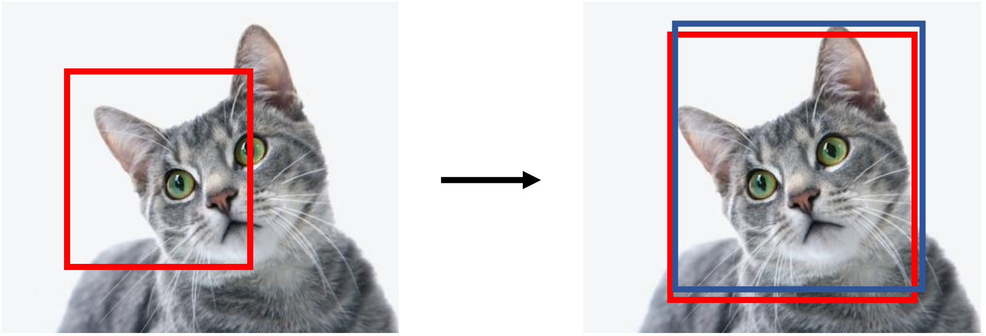
Selective Search 알고리즘



☞ Selective Search is a rule-based algorithm that aggregates segmentations using color, texture and brightness.

1. R-CNN

D. Bounding Box Regression



$\{(P^i, G^i)\}_{i=1,\dots,N}$ 이고 $P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$

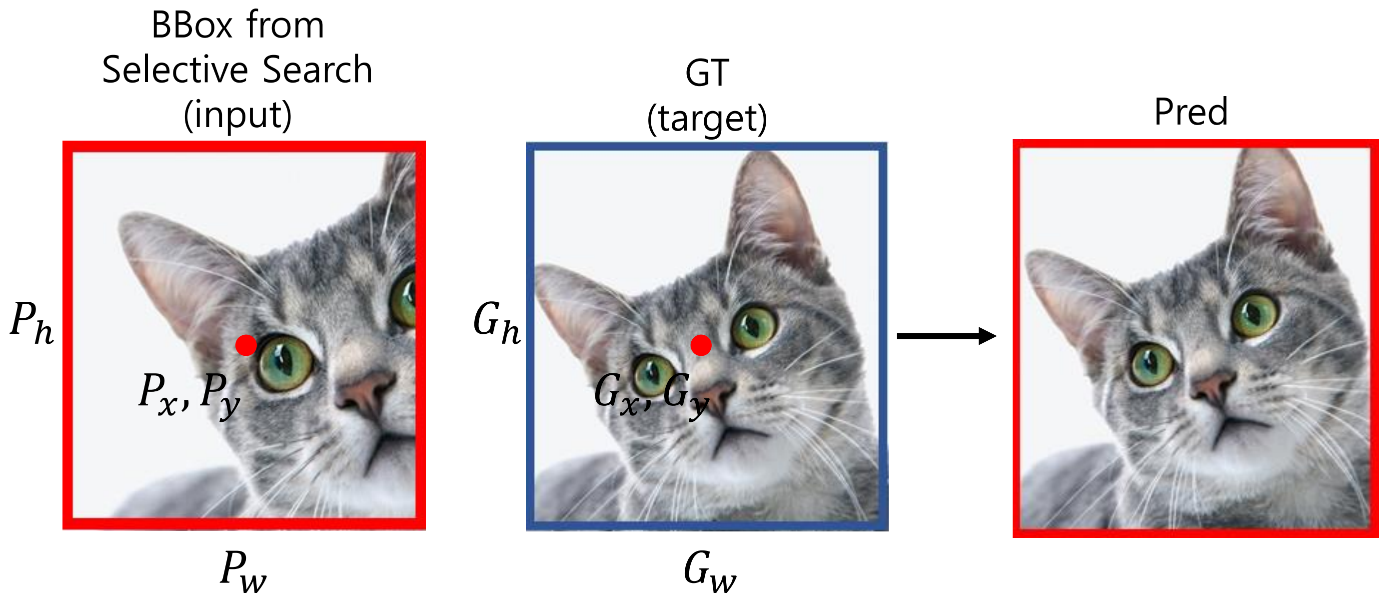
RoI GT

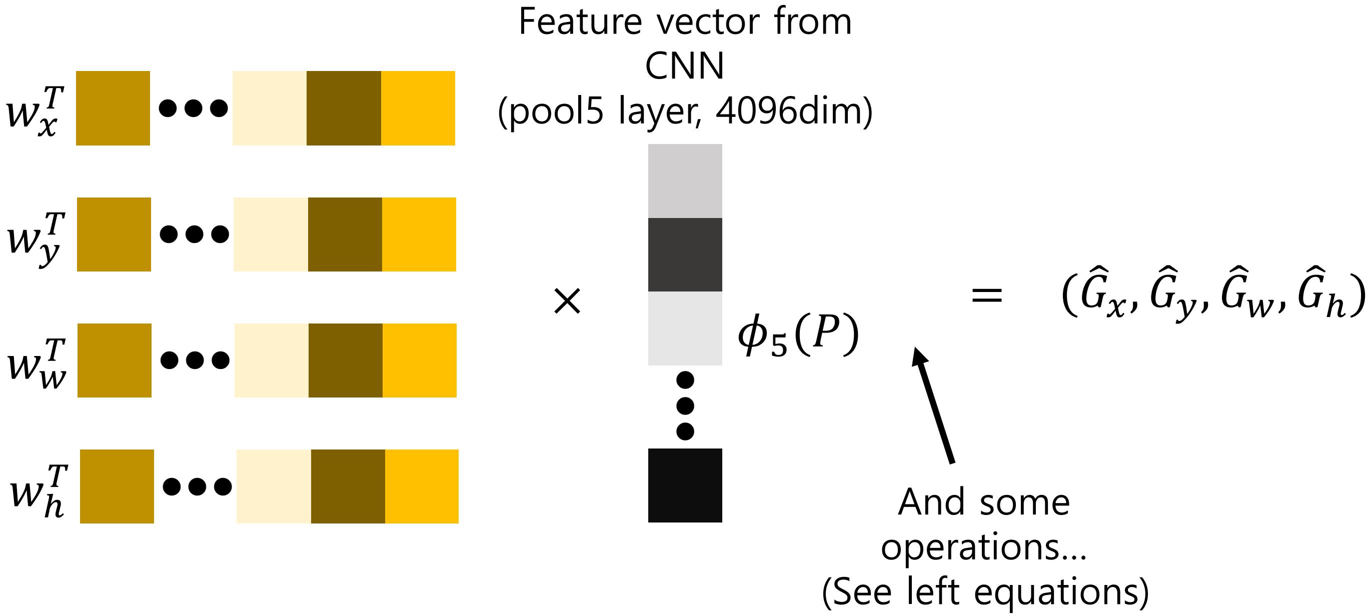
$$\hat{G}_x = P_w d_x(P) + P_x \qquad t_x = (G_x - P_x)/P_w$$
$$\hat{G}_y = P_h d_y(P) + P_y \qquad t_y = (G_y - p_y)/P_h$$
$$\hat{G}_w = P_w \exp(d_w(P)) \qquad t_w = \log(G_w/P_2)$$
$$\hat{G}_h = P_h \exp(d_h(P)) \qquad t_h = \log(G_h/P_h)$$

Learnable vector

$$d_\star(P) = \mathbf{w}_\star^T \phi_5(P)$$ = Scalar

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\arg\min} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^T \phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2$$



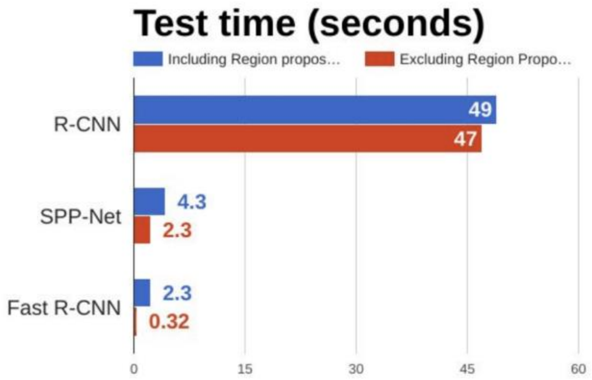


Fast R-CNN

(Girshick et al., ICCV 2015)

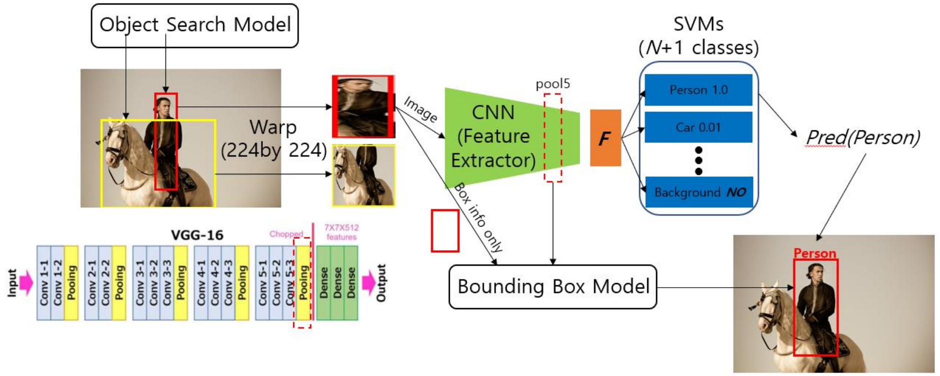
2. FR-CNN

A. Shortages of R-CNN



Too slow!

- Because it forwards 2K feature maps for each images



Too complicated!

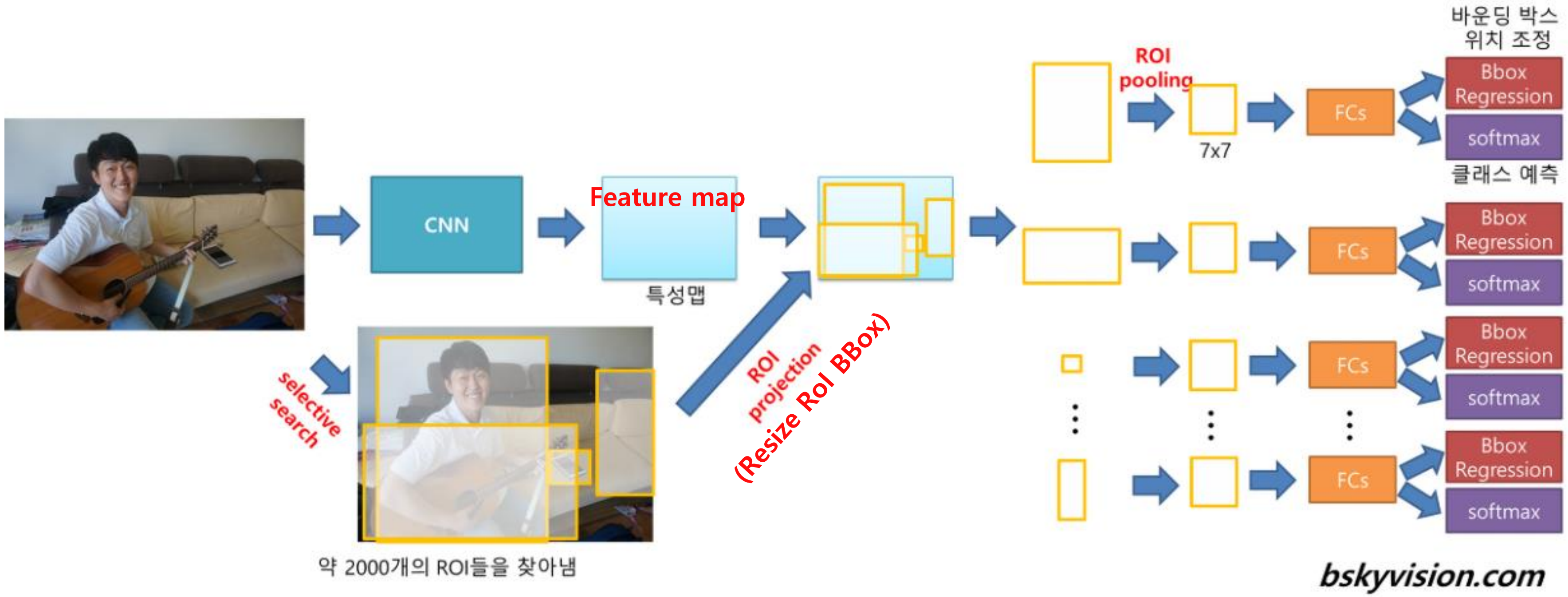
- It has complex architecture

Too inefficient!

- Three networks should be learned with different inputs and outputs. It is hard to find globally optimized solution. In short, it is not an End-To-End learning

2. FR-CNN

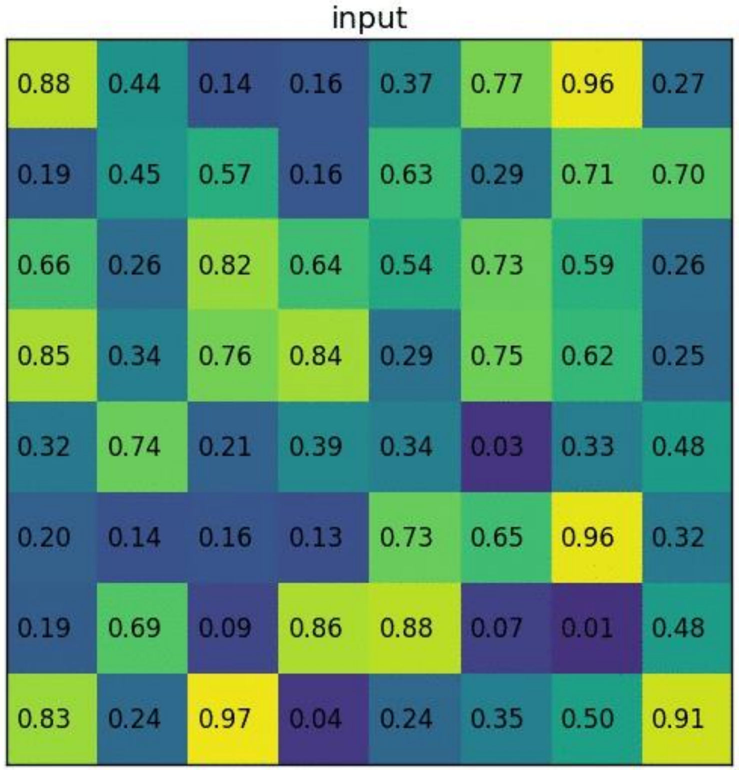
B. Overview



1. Get a feature map from CNN layer
2. Project(or Resize) RoI BBox to the feature map
3. RoI pooling to resize irregular RoIs to fixed size of vector
4. Two sibling FC Layers : One for softmax classification
5. The other for BBox regression

2. FR-CNN

C. RoI Pooling



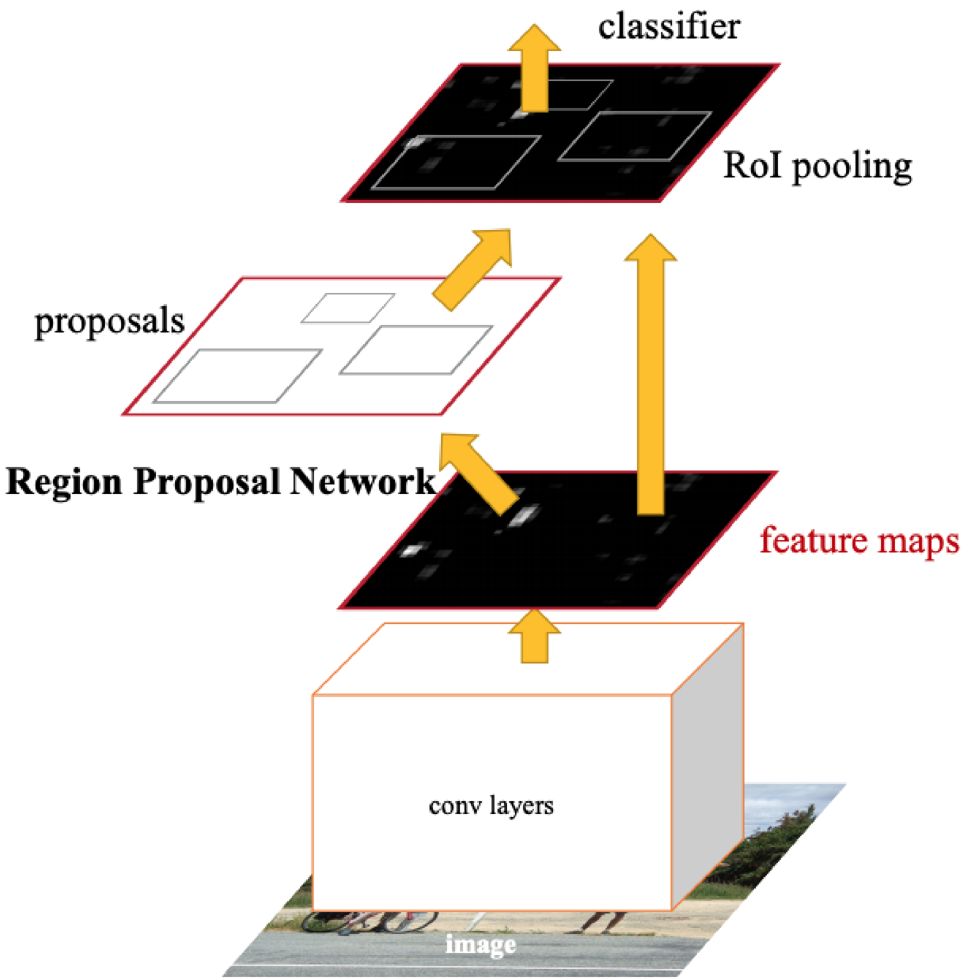
https://blog.airlab.re.kr/2019/10/Fast-R-CNN

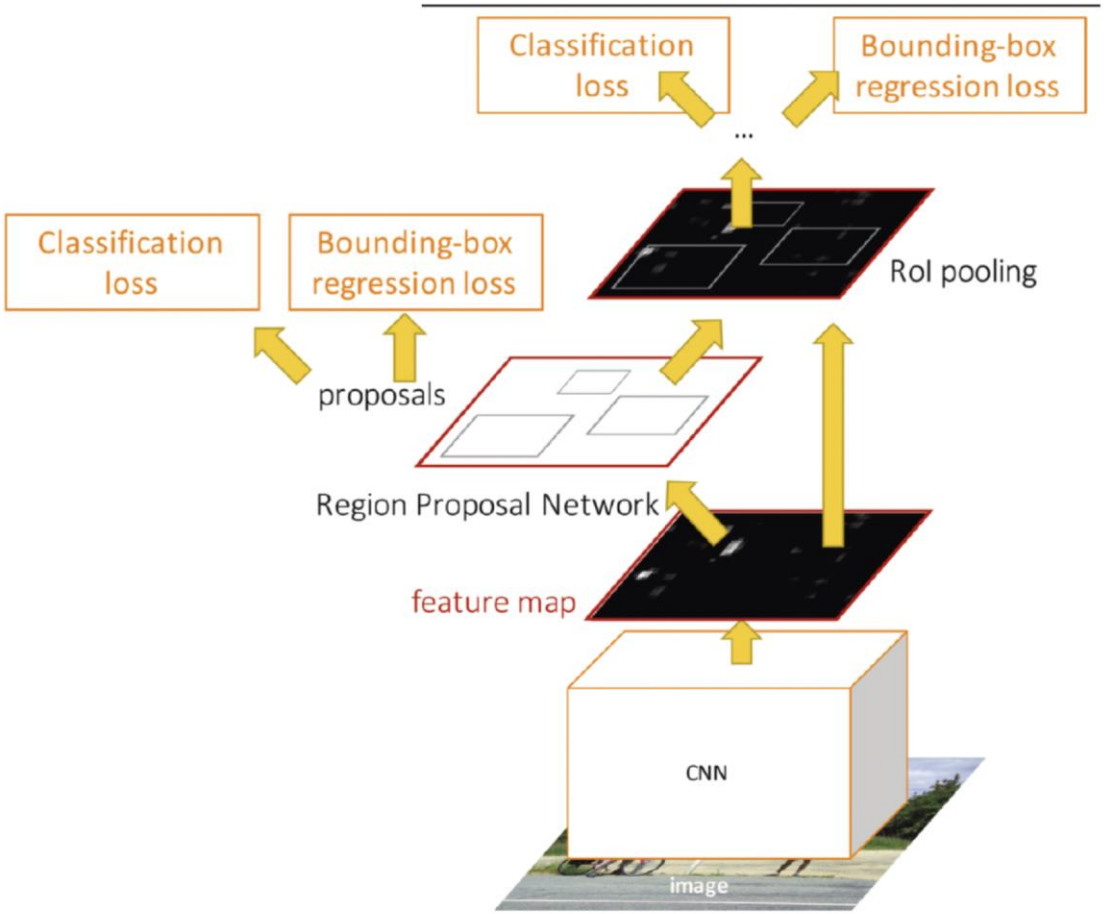
Faster R-CNN

(Girshick et al., NIPS 2015)

3. Faster R-CNN

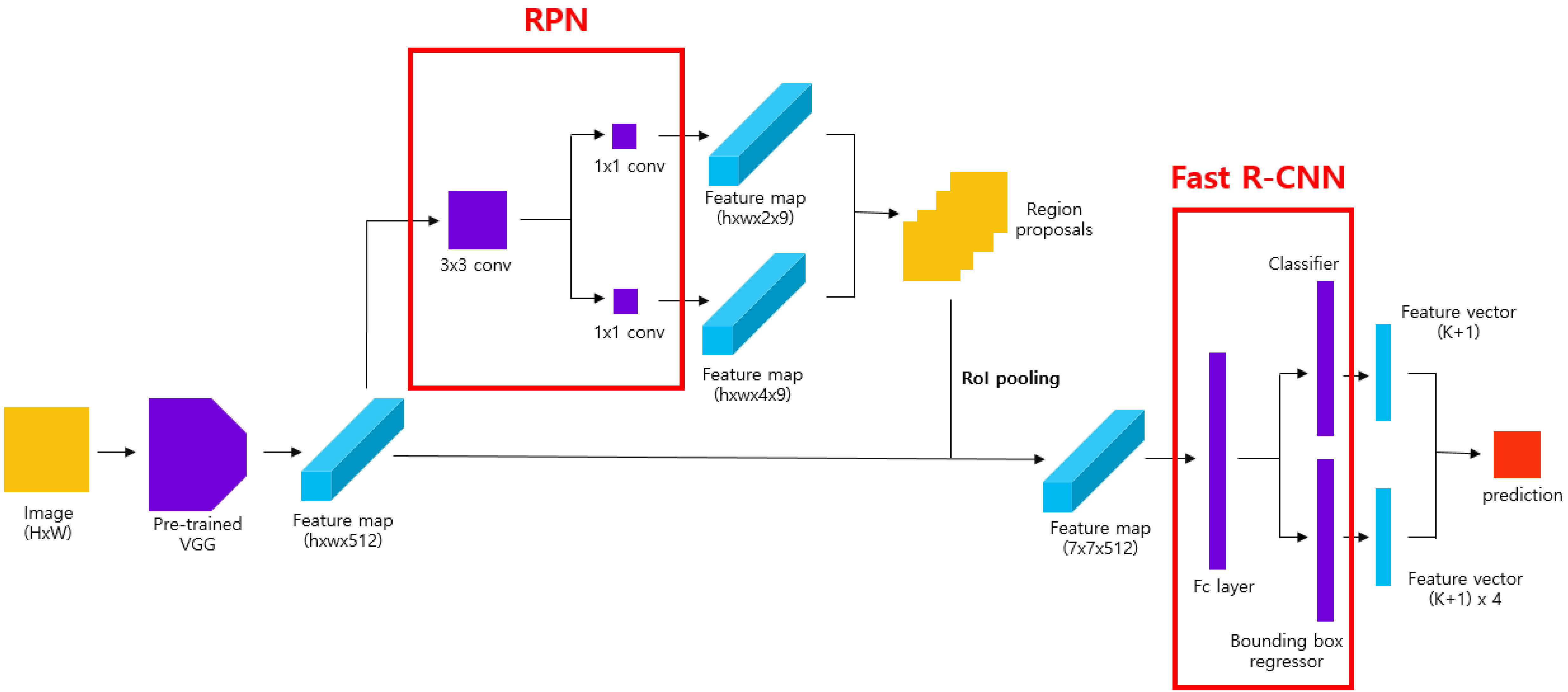
A. Architecture





3. Faster R-CNN

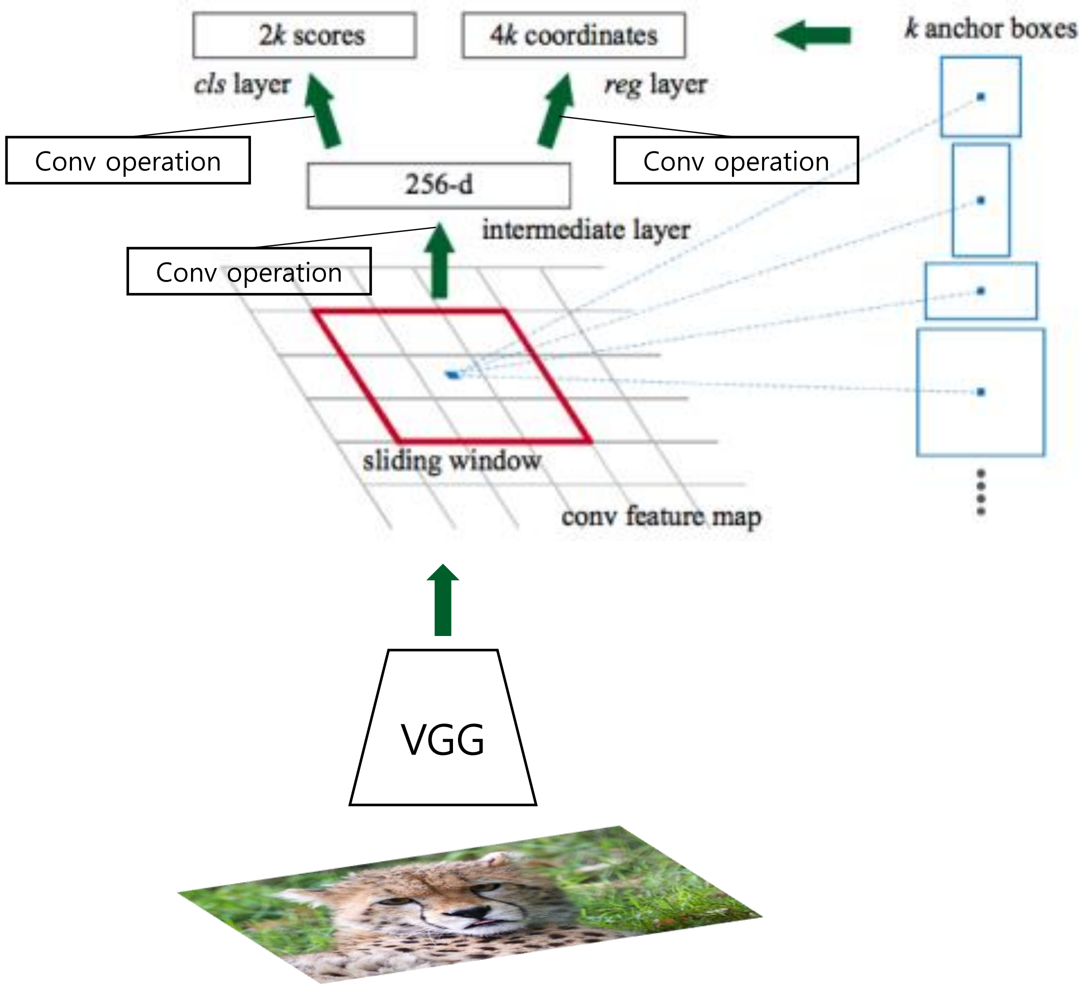
A. Architecture

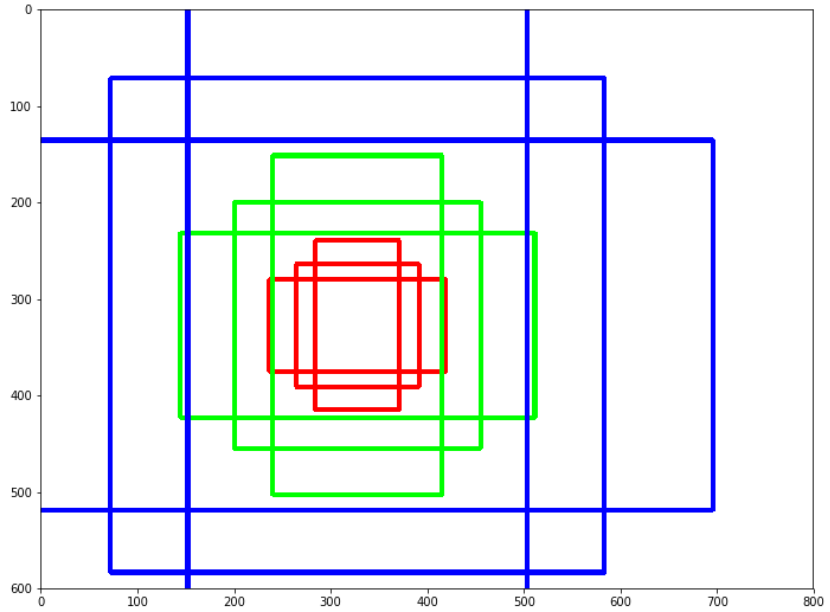


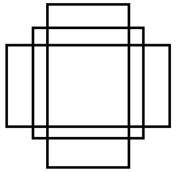
https://herbwood.tistory.com/10

3. Faster R-CNN

B. RPN







3 Boxes : 1x1, 2x1, 1x2

x

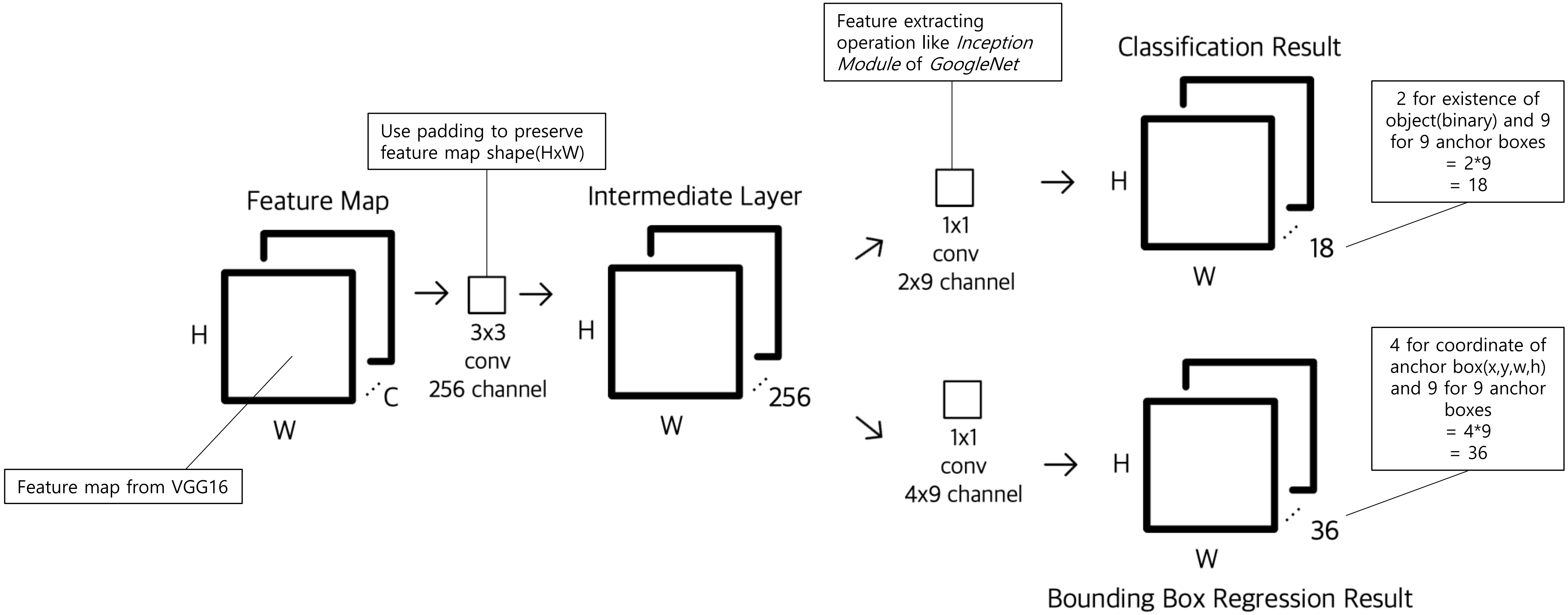
3 ratio

=

9 Anchor Boxes

3. Faster R-CNN

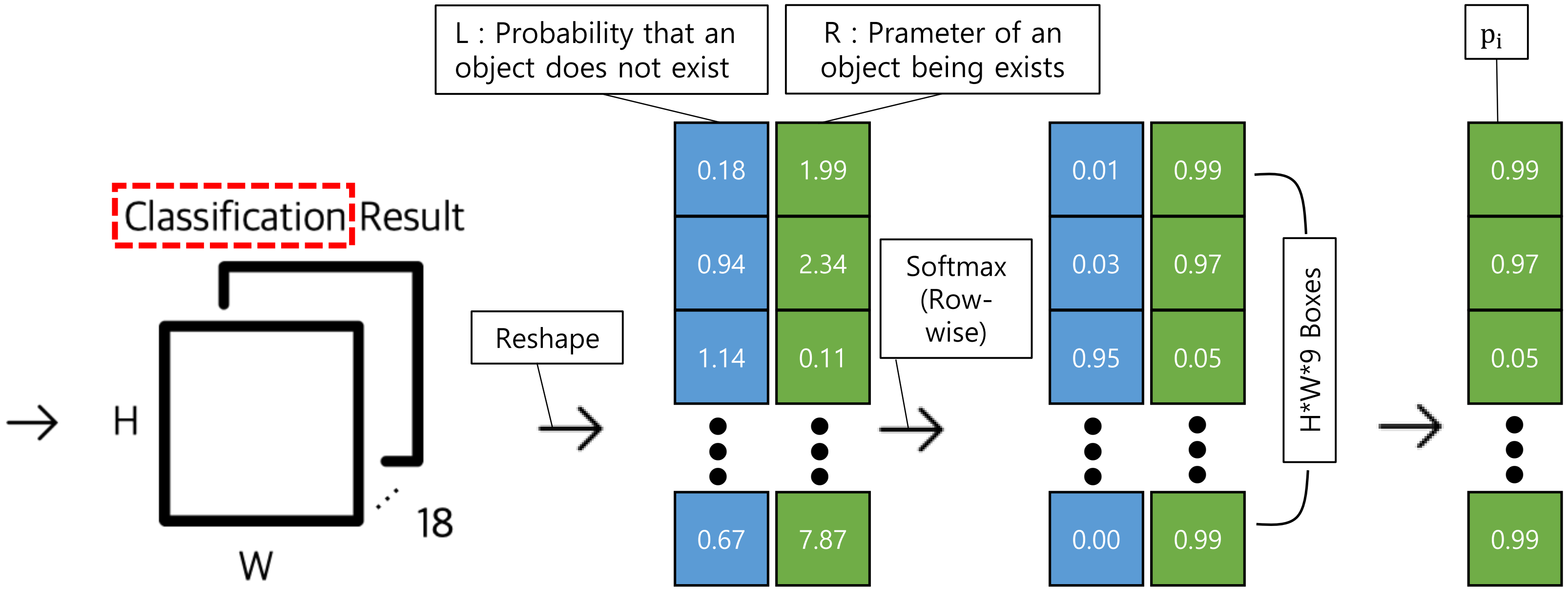
B. RPN



https://yeomko.tistory.com/17

3. Faster R-CNN

B. RPN



$$L(\{p_i\},\{t_i\}) = \boxed{\frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*)} + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*).$$

$$p^* = \begin{cases} 1 & \text{If IoU>0.7} \\ 0 & \text{If IoU<0.3} \end{cases}$$

o.w Don't consider

p_i : Predicted probability of anchor

p_i^* : Ground-truth label(1:positive, 0: negative)

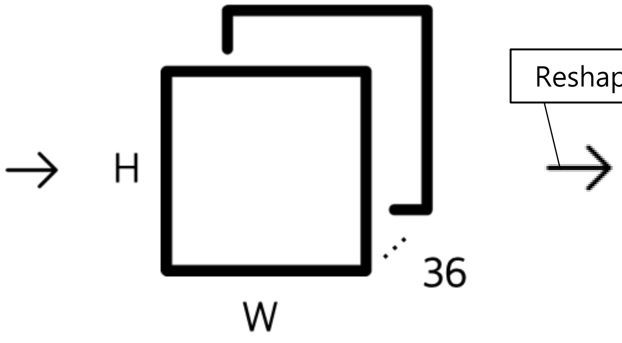
t_i : Predicted BBox

t_i^* : Ground-truth BBox

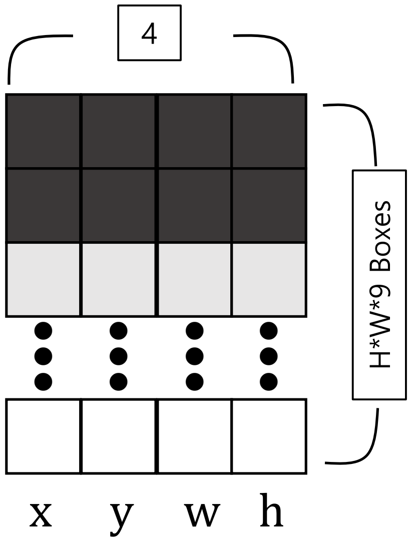
λ : Balancing parameter

3. Faster R-CNN

B. RPN



Bounding Box Regression Result



$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \boxed{\lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*).}$$

$$p^* = \begin{cases} 1 & \text{If IoU>0.7} \\ 0 & \text{If IoU<0.3} \end{cases}$$

o.w Don't consider

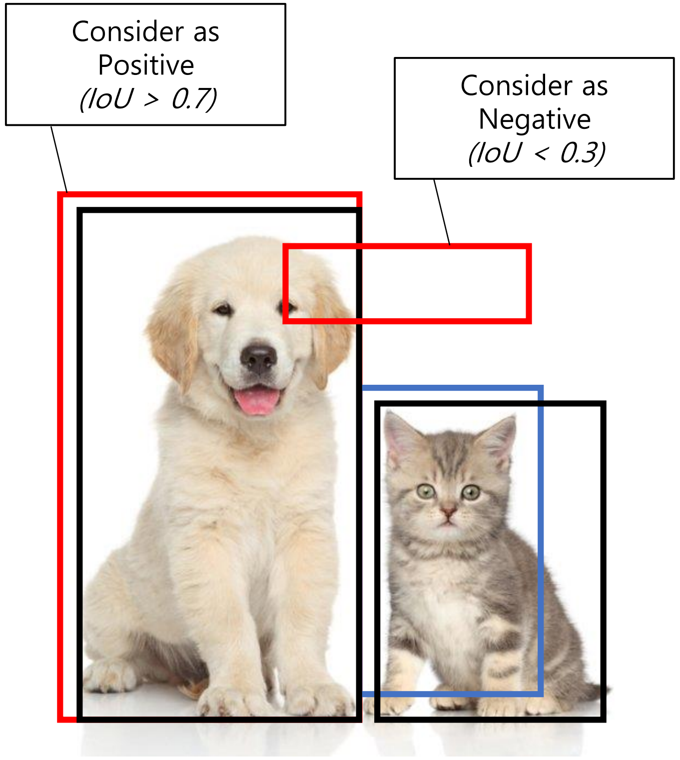
p_i : Predicted probability of anchor

p_i^* : Ground-truth label(1:positive, 0: negative)

t_i : Predicted BBox

t_i^* : Ground-truth BBox

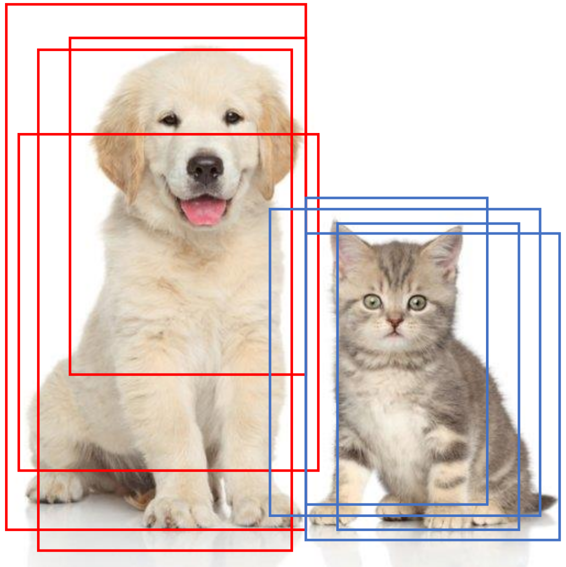
λ : Balancing parameter



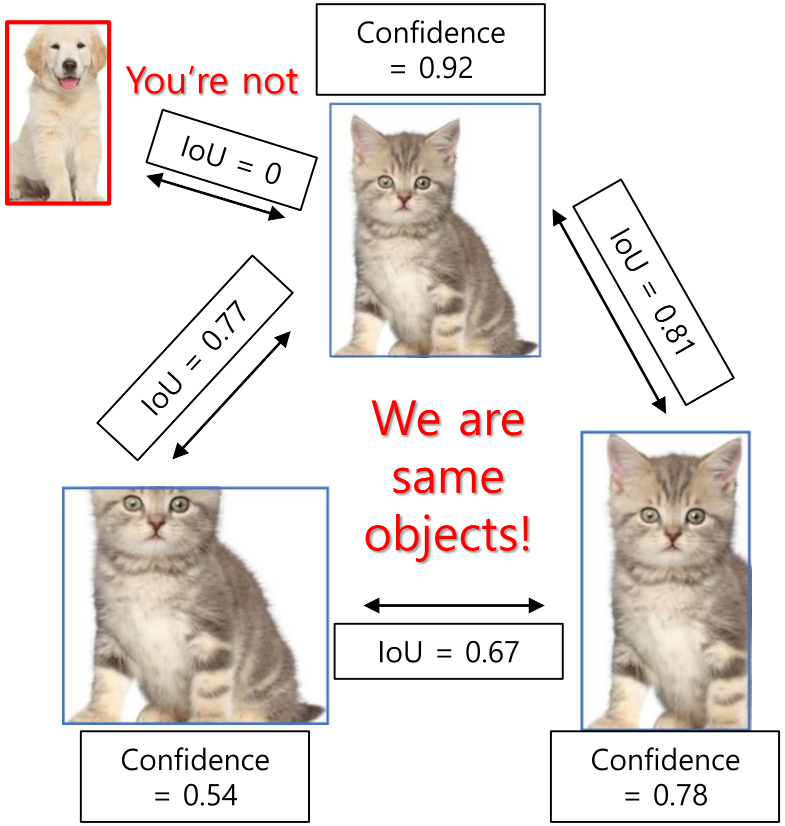
Black : ground-truth

3. Faster R-CNN

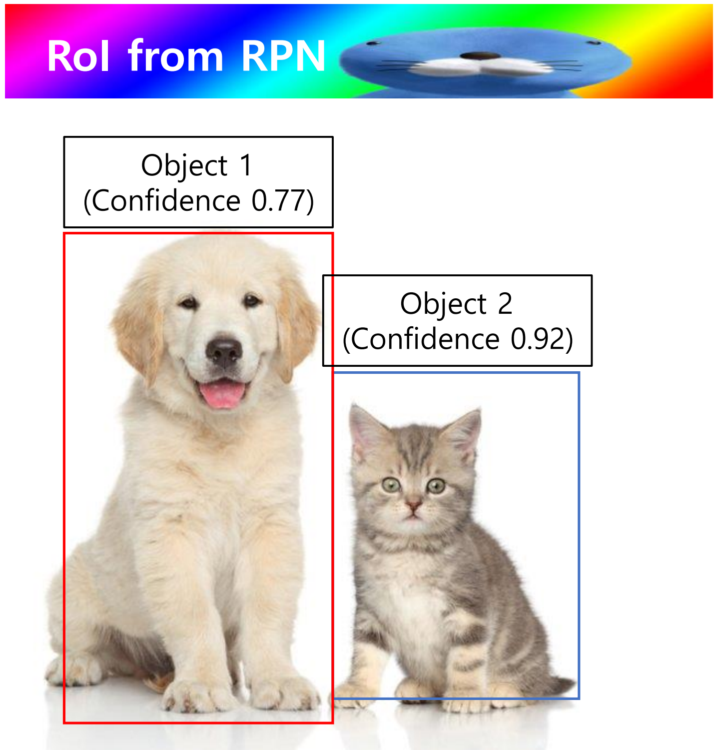
C. NMS(Non Maximum Suppression)



- RPN recommended TOP-k RoI. But it seems too many overlaps
- NMS solves this problem

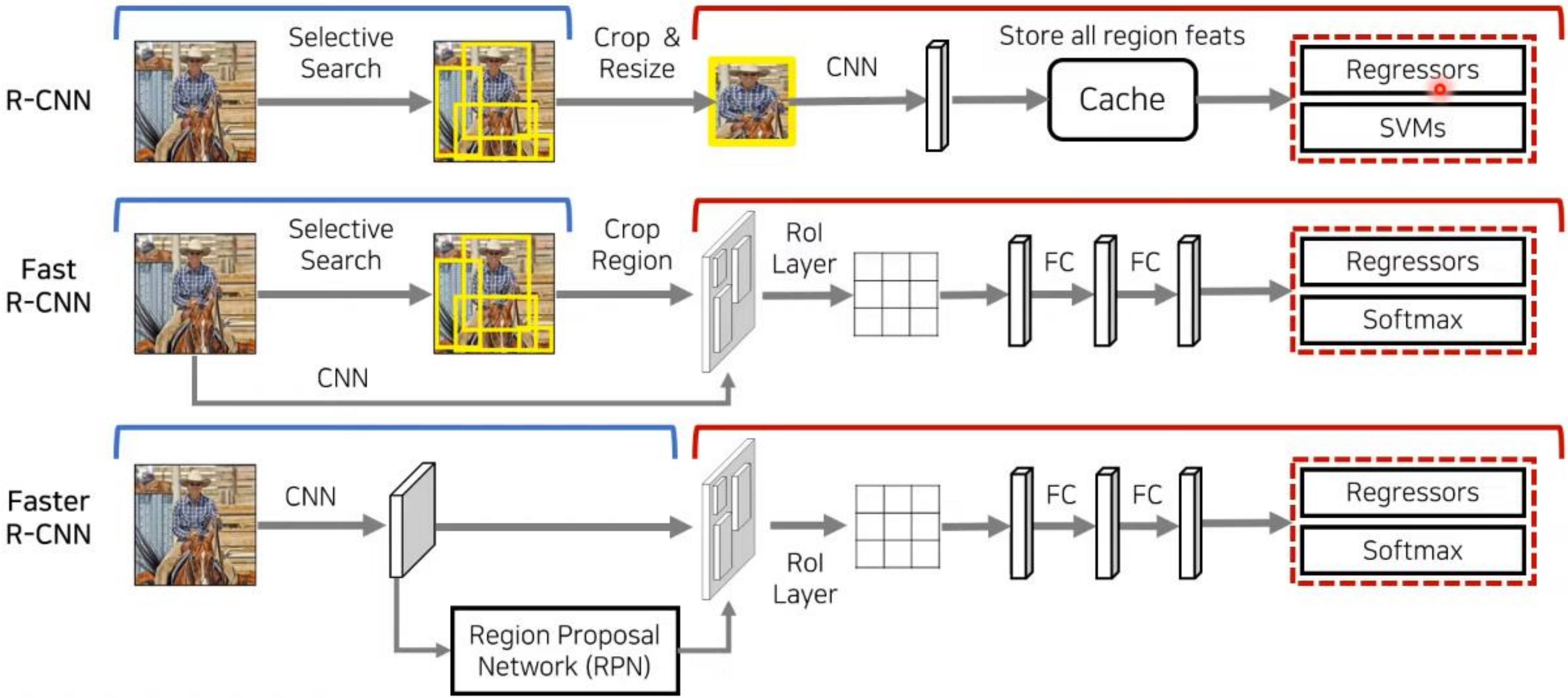


- Objects are considered the same if the IoU is 0.6(or 0.9) or bigger
- For example blue boxes have IoU which is bigger than 0.6. RPN don't know what it is, but the network considers they are same object



- Now, NMS made it more plausible
- Remember, NMS&RPN just recommend RoI. They don't judge what it is.

Summary



YouTube : 나동빈
https://www.arxiv-vanity.com/papers/1908.03673/

Mask R-CNN

(He et al., CVPR 2017)
To be continued…

Reference

[1] Rich feature hierarchies for accurate object detection and semantic segmentation(Girshick et al., 2014)
[2] https://ganghee-lee.tistory.com/37
[3] Selective Search for Object Recognition(J. R. R. Uijlings et al., 2012)
[4] Fast R CNN(Girshick et al., 2015)
[5] bskyvision.com
[6] https://blog.airlab.re.kr/2019/10/Fast-R-CNN
[7] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks(Girshick et al., 2016)
[8] YouTube : 나동빈
[9] https://herbwood.tistory.com/10
[10] https://yeomko.tistory.com/17
[11] Recent Advances in Deep Learning for Object Detection(Wu et al.,)
[12] https://prateekvjoshi.com/2015/12/15/how-to-compute-confidence-measure-for-svm-classifiers/
[13] https://github.com/shkim960520/faster-rcnn-for-studying/blob/main/model/region_proposal_network.py

Thank You

Any Questions?