



Fast R-CNN Faster R-CNN

CONTENTS

/01

R-CNN

/02

Fast R-CNN

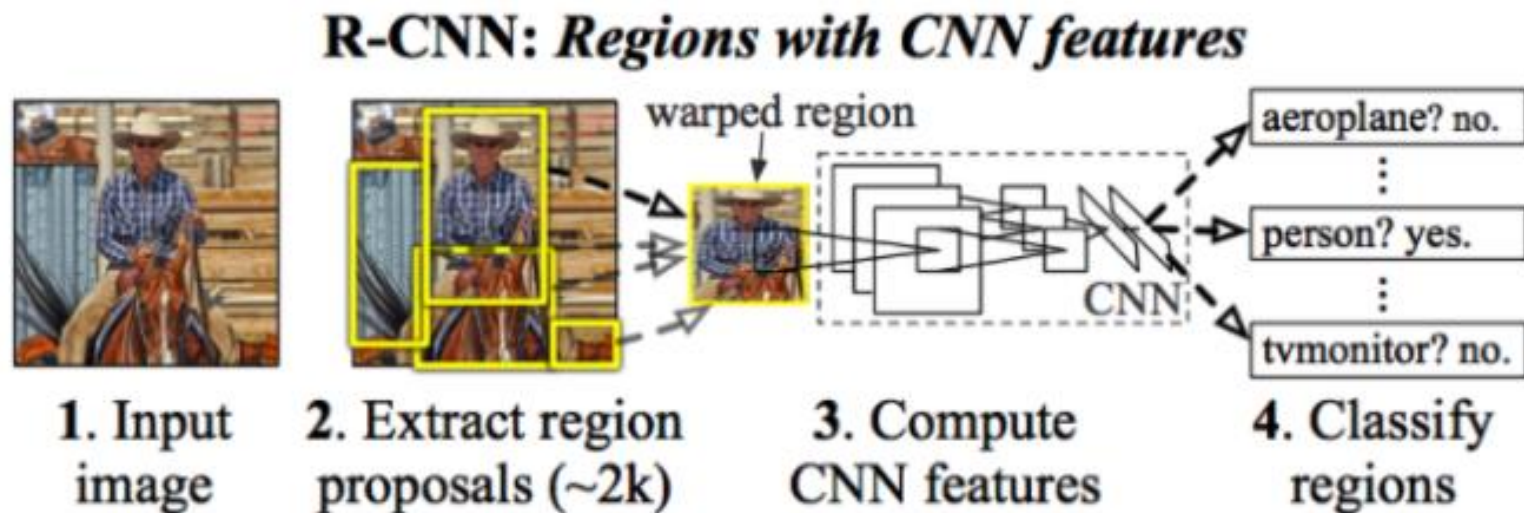
/03

Faster R-CNN

/04

Summary

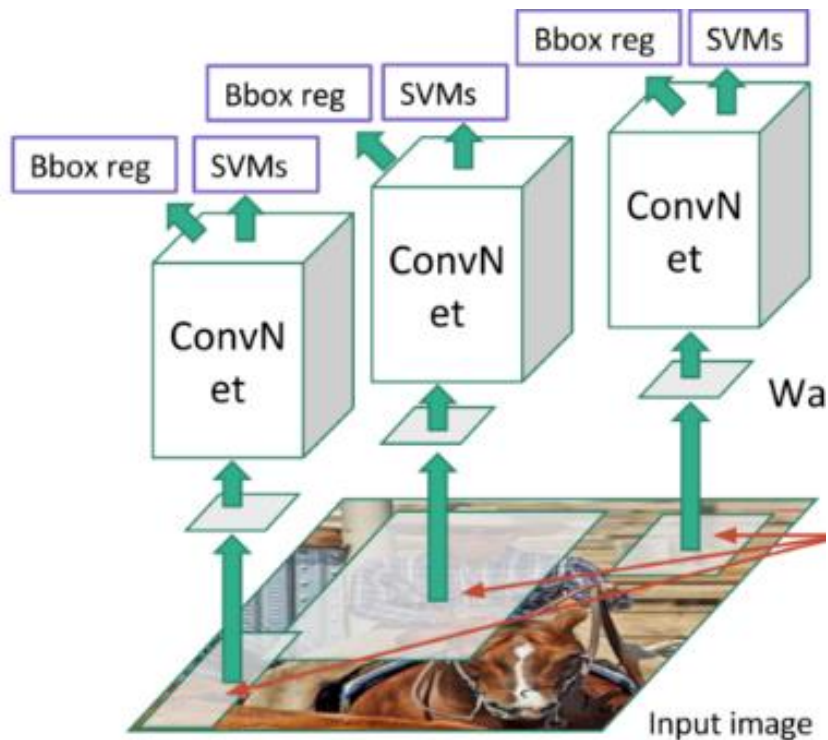
01 R-CNN R-CNN이란?



1. Input 이미지에 **Selective Search** 알고리즘을 적용하여 2,000개의 **region proposal**을 생성
2. region 리사이즈(warp)
3. CNN을 통해 각 proposal마다 고정된 길이의 **feature vector**를 추출
4. 각 region마다 category-specific linear **SVM**을 적용하여 classification을 수행

01 R-CNN

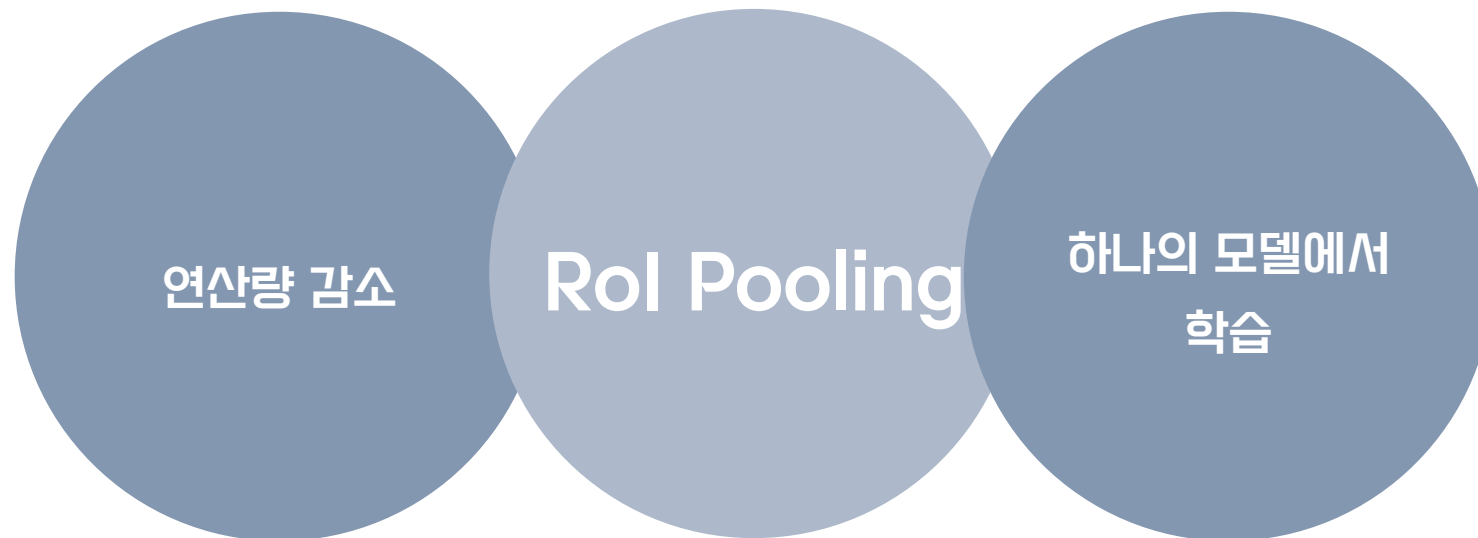
R-CNN의 한계점



- RoI마다 CNN 연산을 함으로써 **속도저하**
→ Selective Search로 2000개의 region proposal을 뽑고,
각 영역마다 CNN을 수행하기 때문.
- CNN, SVM, Bounding Box Regression 총 3가지 모델이 **여러 단계로 학습됨**
→ 각 region proposal에 대해 ConvNet forward pass를 실행할 때 연산을 공유하지 않음.
SVM, bounding box regression에서 학습한 결과가 CNN을 업데이트 시키지 못한다.

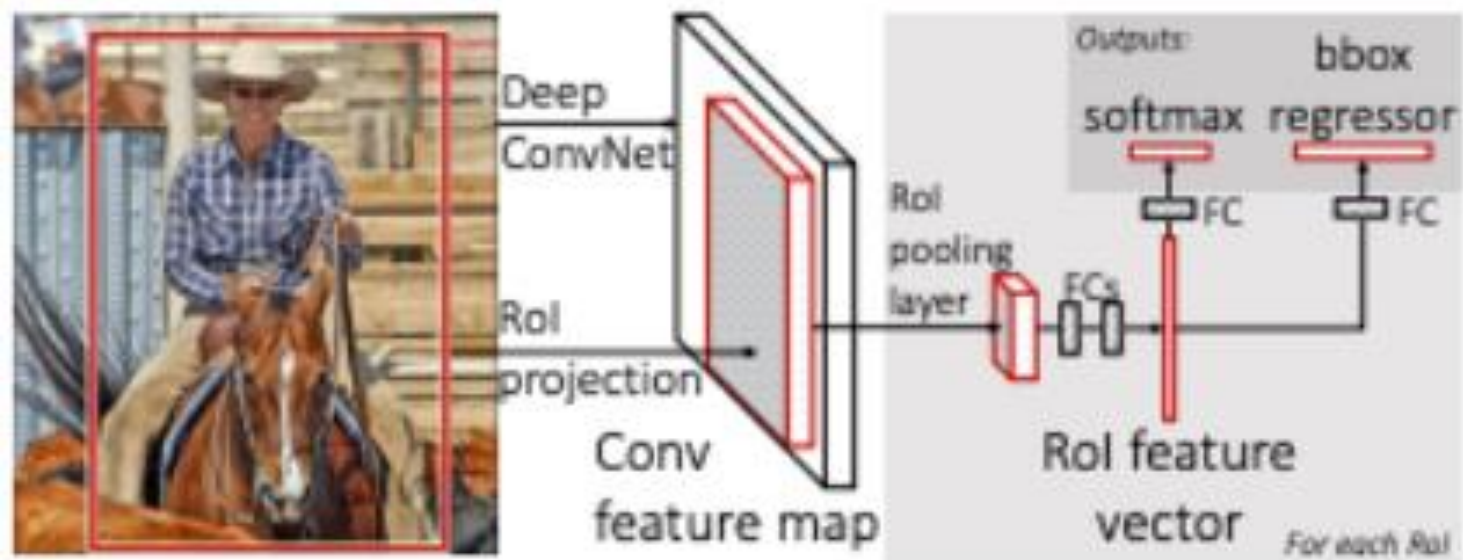
02 Fast R-CNN

R-CNN의 한계점 극복



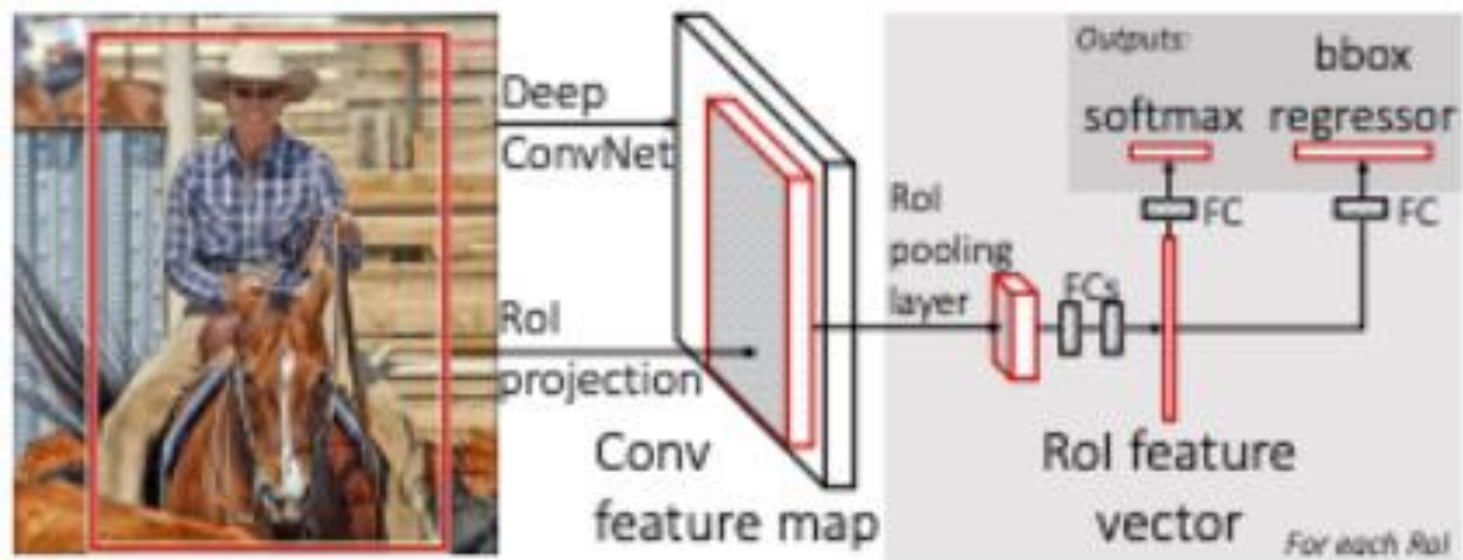
CNN feature 추출부터 classification, bounding box regression까지 모두 **하나의 모델**에서 학습시키자!

02 Fast R-CNN 프로세스



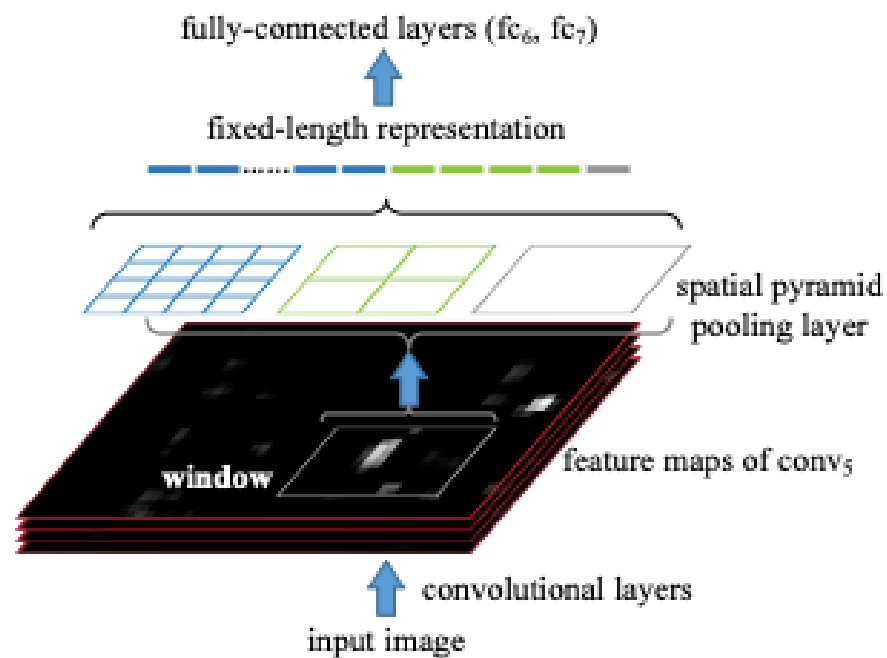
1. 전체 이미지를 CNN에 통과시켜 feature map을 추출한다.
2. Selective Search를 통해 RoI를 찾는다. (R-CNN과 동일)
3. 각각의 RoI에 대해 RoI Pooling을 진행하여 고정된 크기의 feature vector를 얻는다.

02 Fast R-CNN 프로세스



4. feature vector는 FC layer를 통과한 뒤, 두 브랜치로 나뉘게 된다.
5. 하나는 softmax를 통과하여 RoI에 대해 object classification을 한다.
6. 하나는 bounding box regression을 통해 selective search로 찾은 box의 위치를 조정한다.

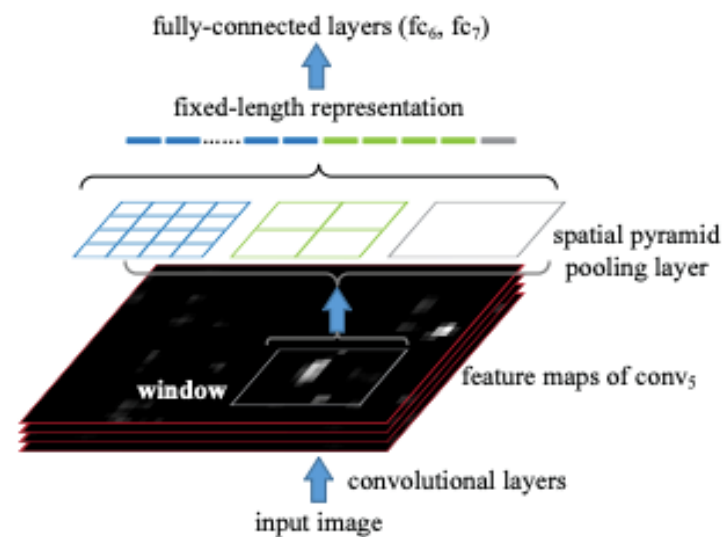
02 Fast R-CNN Spatial Pyramid Pooling



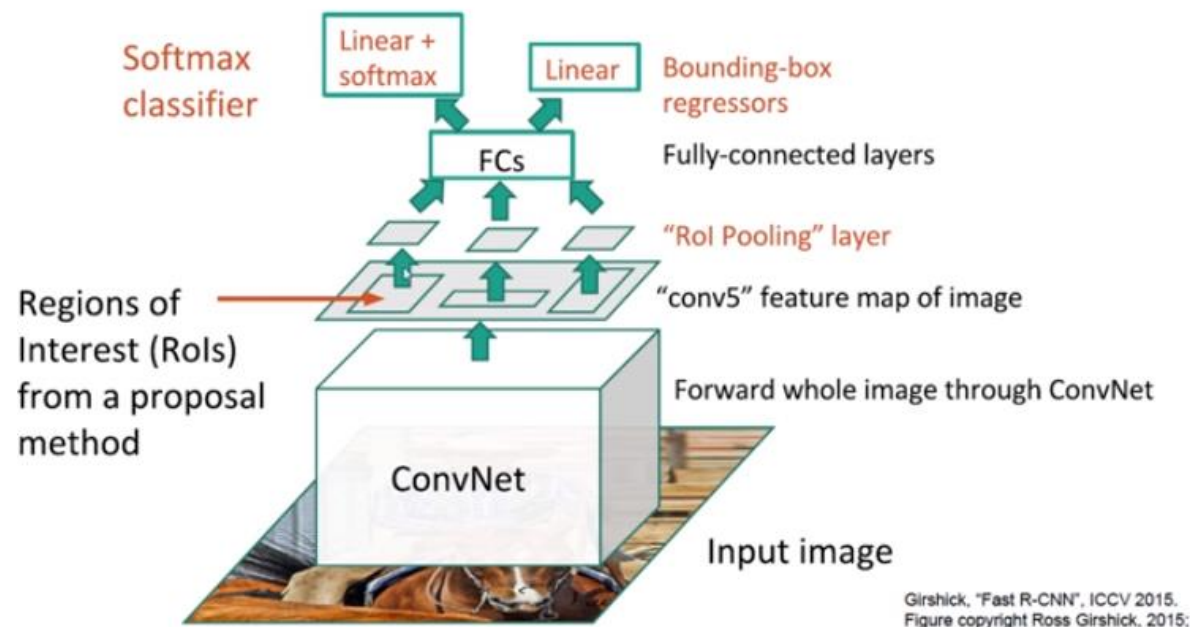
입력 이미지의 크기에 관계 없이 Conv layer들을 통과시키고,
FC layer 통과 전에 feature map들을 동일한 크기로 조절해주는 pooling을 적용하자!

02 Fast R-CNN Spatial Pyramid Pooling

1. 이미지를 CNN에 통과시켜 feature map을 추출한다.
2. 미리 정해진 영역으로 feature map을 나눠준다.
3. bin 내에서 max pooling을 적용하여 각 bin마다 하나의 값을 추출하고, 최종적으로 피라미드 크기만큼 max값을 추출하여 3개의 피라미드의 결과를 쪽 이어 붙여 고정된 크기 vector를 만든다.
(피라미드 한 칸을 bin이라고 한다.)
4. 만들어진 vector가 FC layer input으로 들어간다.



02 Fast R-CNN RoI Pooling

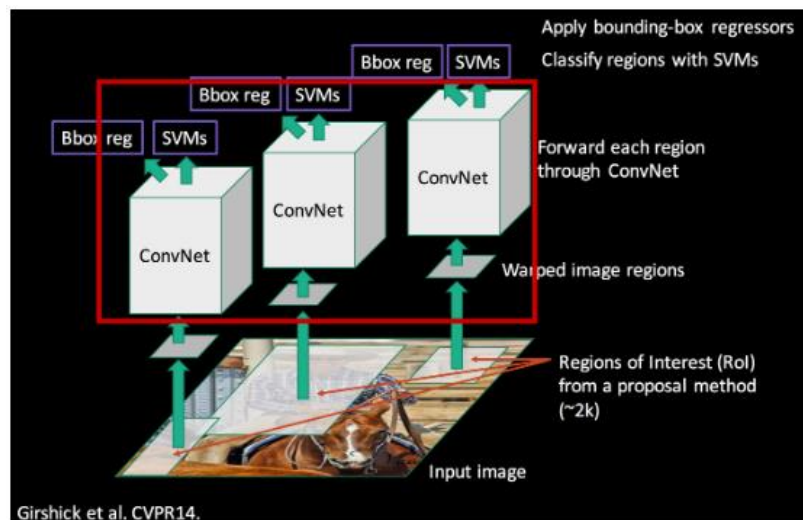


Fast R-CNN는 7x7 사이즈인 1개의 피라미드를 적용시킨 SPP로 구성되어 있다.

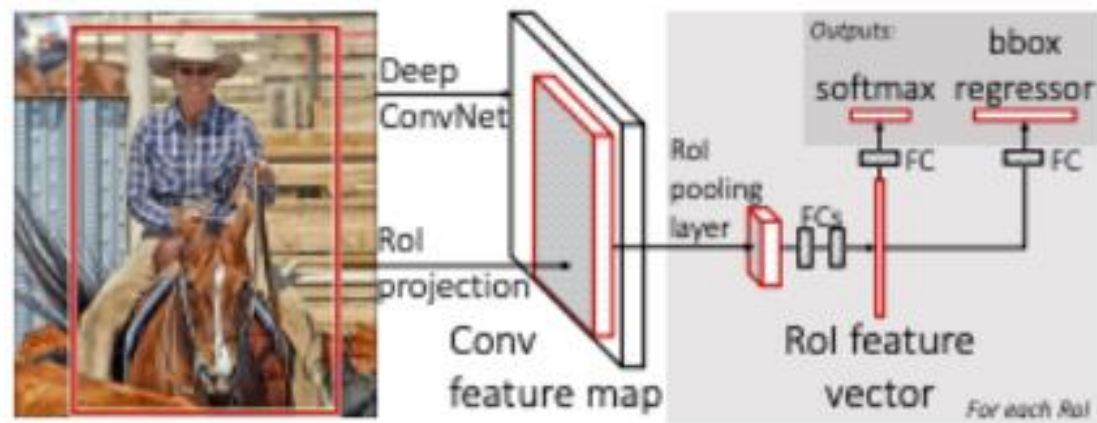
Fast R-CNN에서 적용된 1개의 피라미드 SPP로 고정된 크기의 feature vector를 만드는 과정을 RoI Pooling이라고 한다.

이를 통해 모든 2000개의 region proposal마다 해야 했던 2000번의 CNN 연산이 1번으로 줄어든다.

02 Fast R-CNN end-to-end



R-CNN



Fast R-CNN

Fast R-CNN에서는 RoI Pooling을 추가함으로써 동일한 데이터가 각각 softmax(classification), bounding box regression(localization)으로 들어간다.

즉, 연산을 공유한다.

02 Fast R-CNN Summary

key - point

RoI Pooling

⇒

- ① 1번의 CNN 연산
- ② end-to-end로 한 번에 학습

한계점

Selective Search 알고리즘의 사용

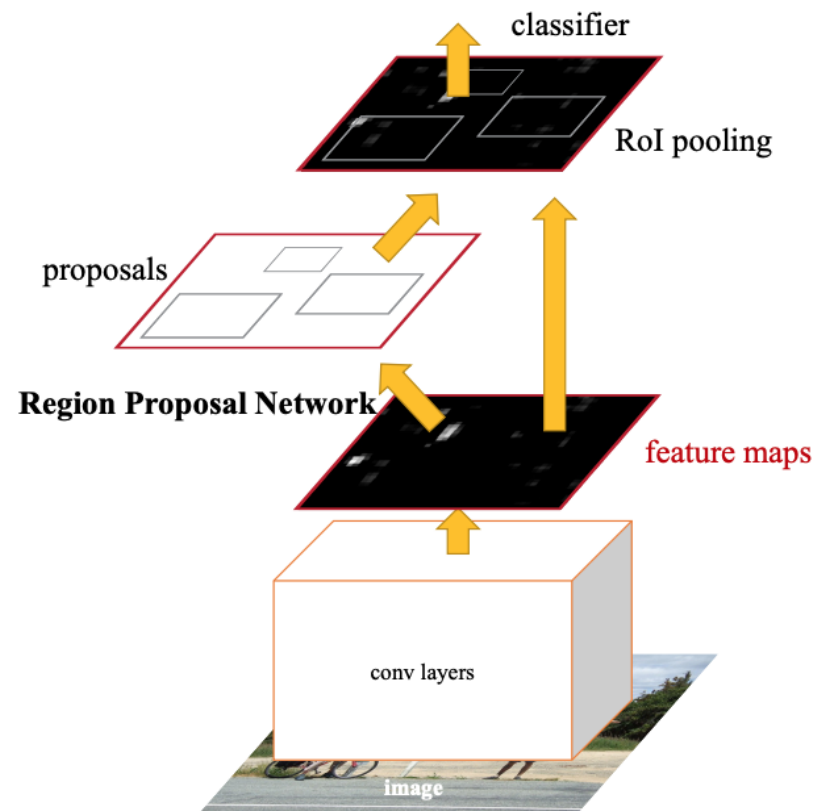
03

Faster R-CNN

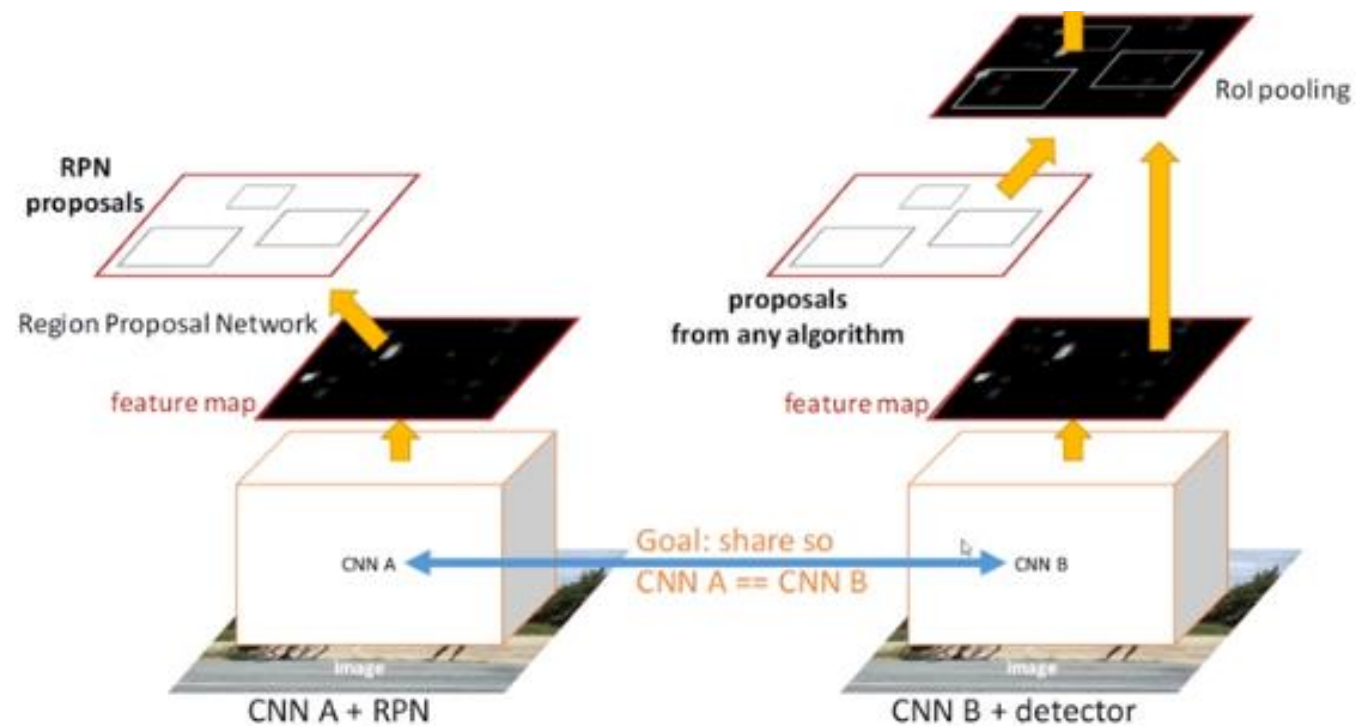
Introduction

Fast R-CNN + RPN

Selective Search를 사용하지 않고
Region Proposal도 같은 네트워크 안에서 같이 수행하자!



03 Faster R-CNN Introduction

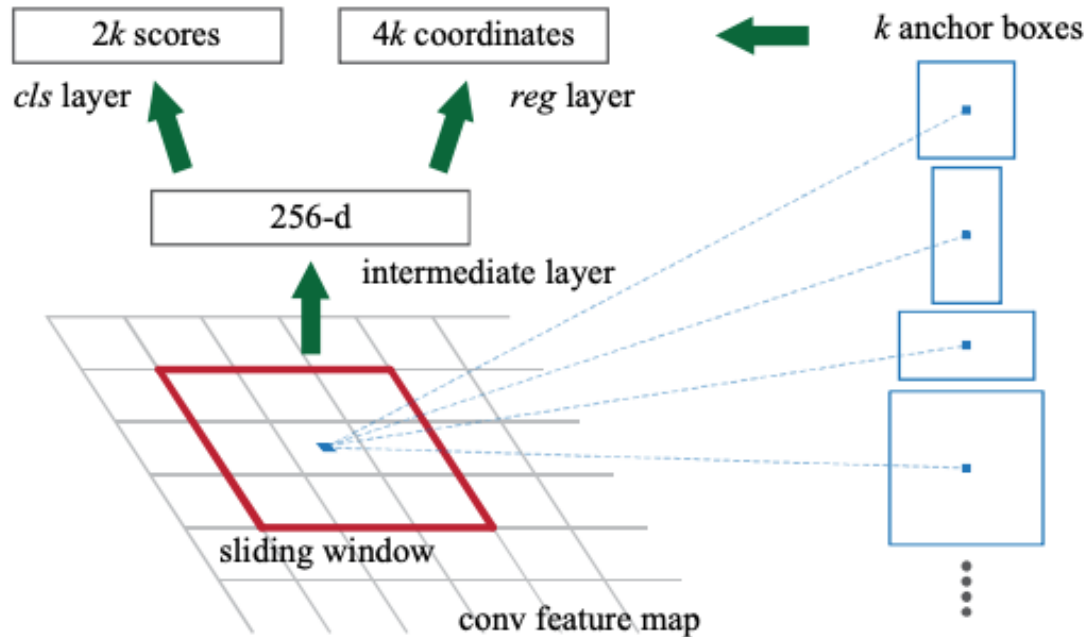


CNN 네트워크의 공유

03

Faster R-CNN

Region Proposal Network



1. CNN 모델에서 뽑아낸 feature map을 input으로 넣는다.
2. feature map을 sliding하고 k개의 Anchor Box를 통해 영역을 정한다.
3. input에 대해 conv filter 256개를 연산한다.
4. Classification Layer와 Bounding box regression을 통해 물체가 있는 곳을 학습한다.

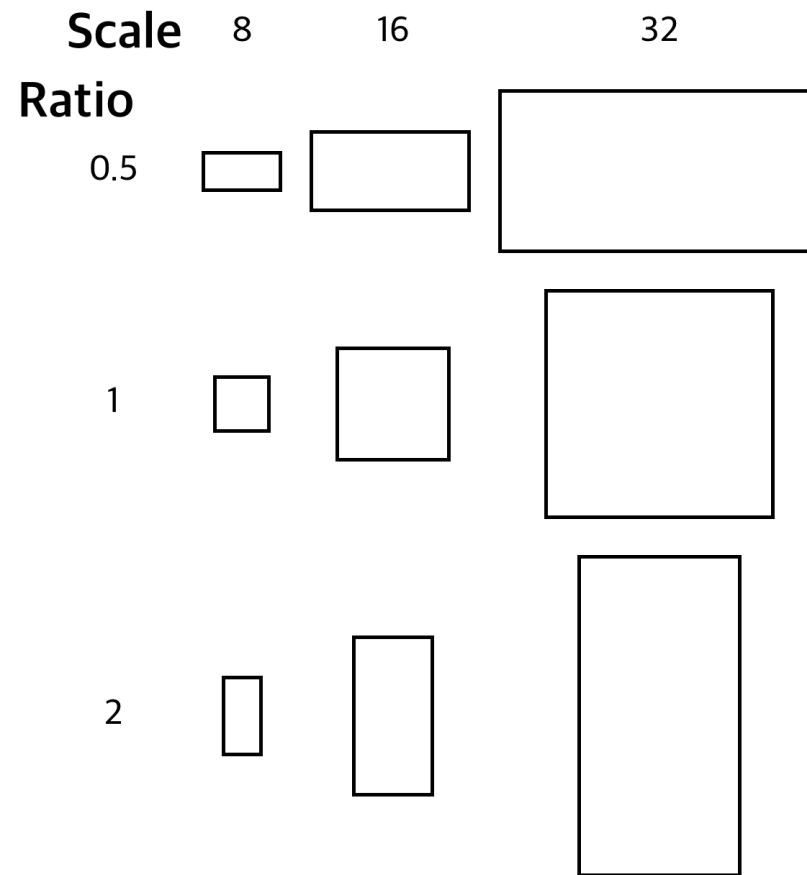
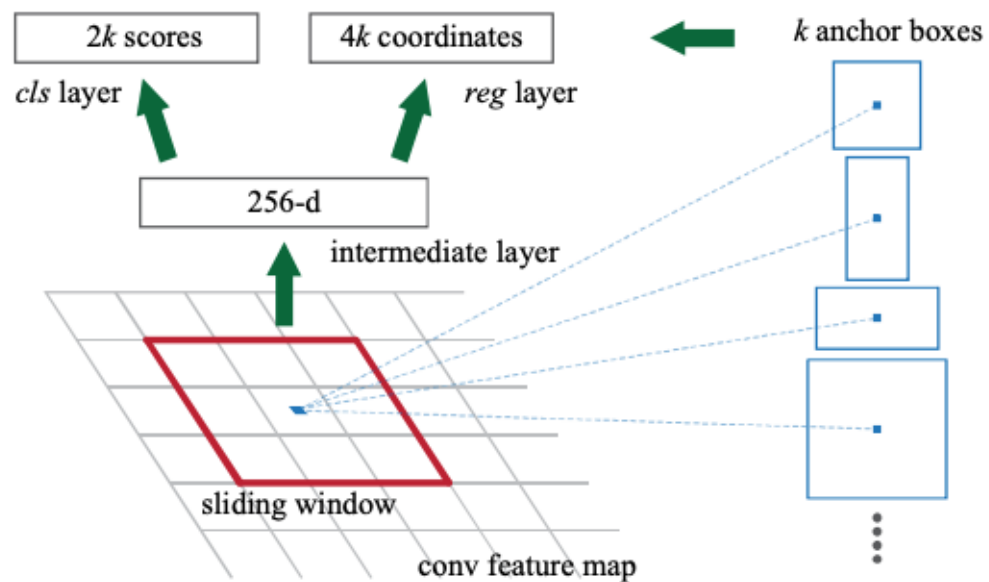
03

Faster R-CNN

Region Proposal Network

Anchor Box

sliding window의 각 위치에서 Bounding Box의 후보로 사용되는 상자

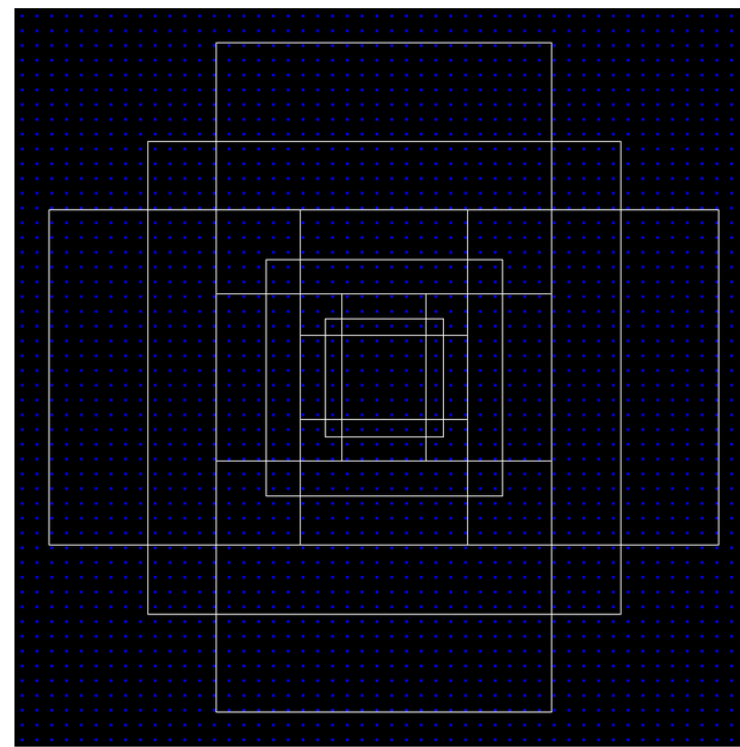
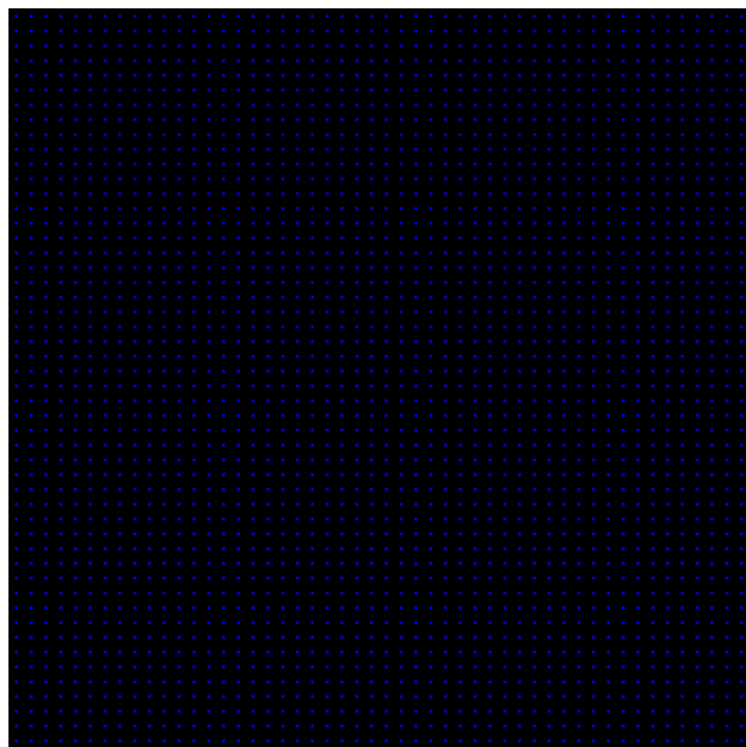


Scale(8, 16, 32), Ratio(0.5, 1, 2)

03 Faster R-CNN

Region Proposal Network

Anchor Box

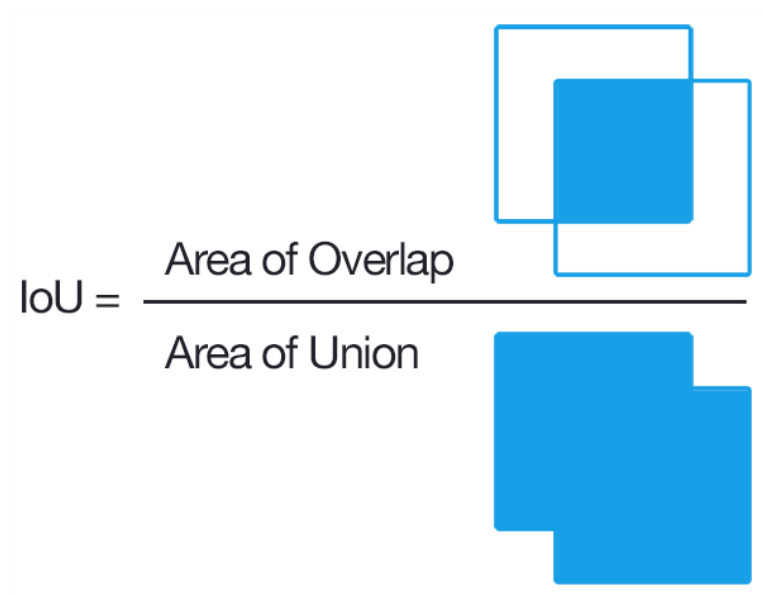


800 x 800 x 3 input image → 50 x 50 x 512 feature map

03 Faster R-CNN Training RPN

Anchor box를 기준으로 그 안에 물체가 있는지 없는지 학습

이미지와 GT(Ground Truth Box)가 들어왔을 때 각 Anchor마다 Anchor가 물체를 감싸고 있는지, Background를 감싸고 있는지 Labeling

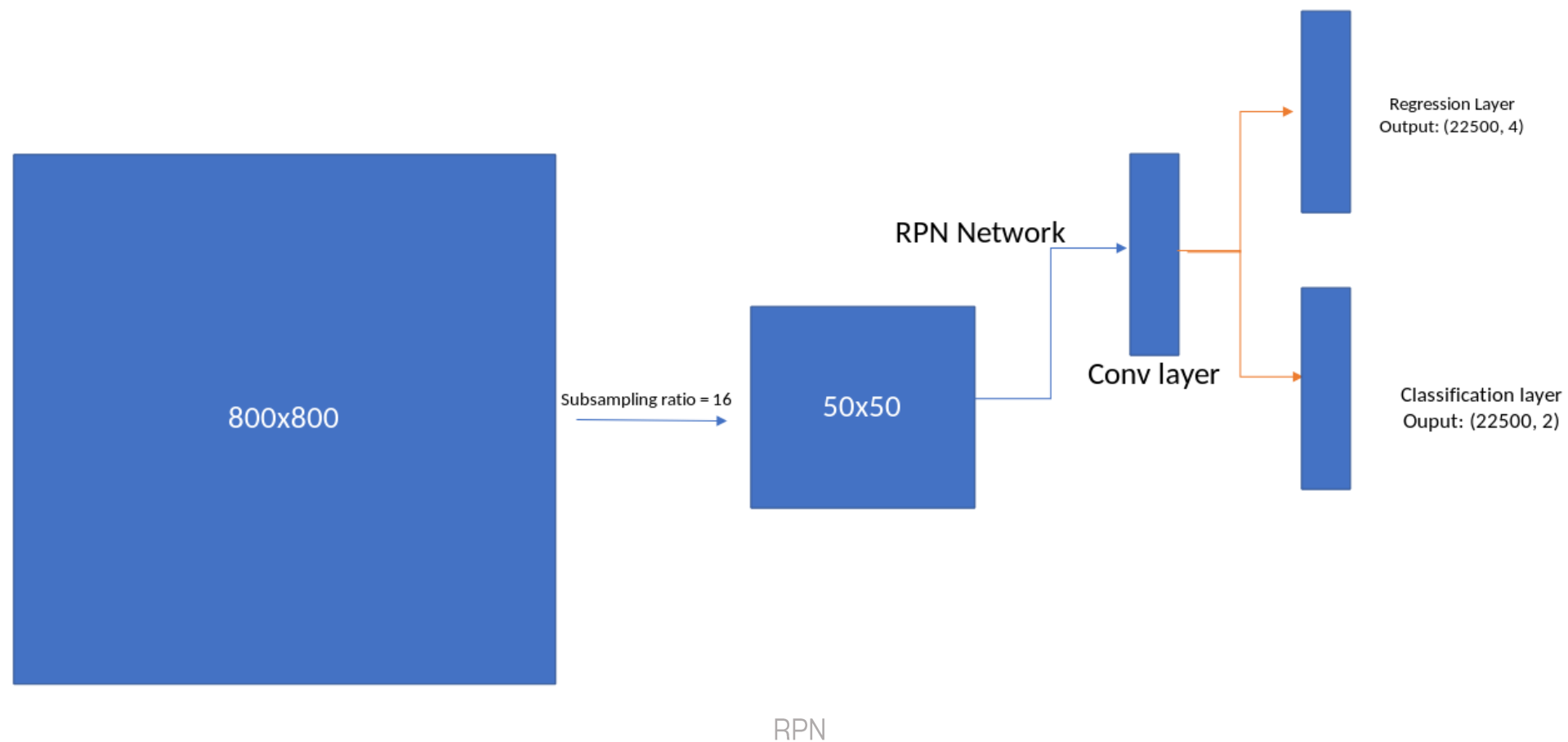


교집합 / 합집합

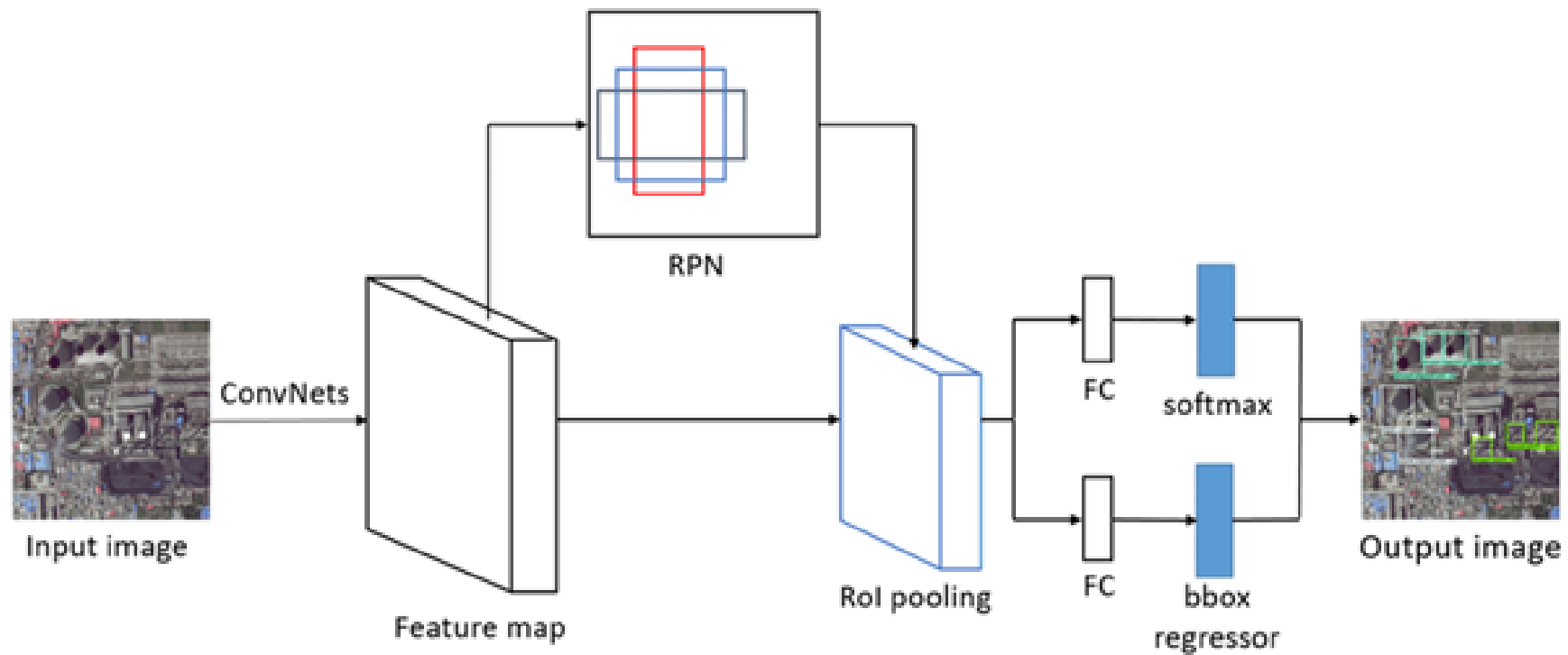
1. GT Label은 만들어진 각 22,500개의 Anchor들과 GT의 IoU를 계산해서 IoU가 0.7보다 크면 1(Positive), IoU가 0.3보다 작으면 0(Negative)로 두고 나머지는 -1로 둔다.
2. 1의 값이 나온 Anchor수가 많지 않을 수 있어서 GT마다 IoU가 가장 높은 Anchor 1개를 뽑아 이 또한 1로 Labeling

03 Faster R-CNN

Training RPN



03 Faster R-CNN 프로세스



03

Faster R-CNN

Summary

key - point

① RPN

② Sharing convolutional features (computation)

Summary

R-CNN vs. Fast R-CNN vs. Faster R-CNN

R-CNN

Rol (Selective Search) \Rightarrow Conv \Rightarrow SVM, Bbox reg

Fast R-CNN

Conv \Rightarrow Rol(Selective Search) \Rightarrow Rol pooling layer \Rightarrow Softmax, Bbox reg

Faster R-CNN

Conv \Rightarrow Rol(Region Proposal Network) \Rightarrow Rol pooling layer \Rightarrow Softmax, Bbox reg

감사합니다.