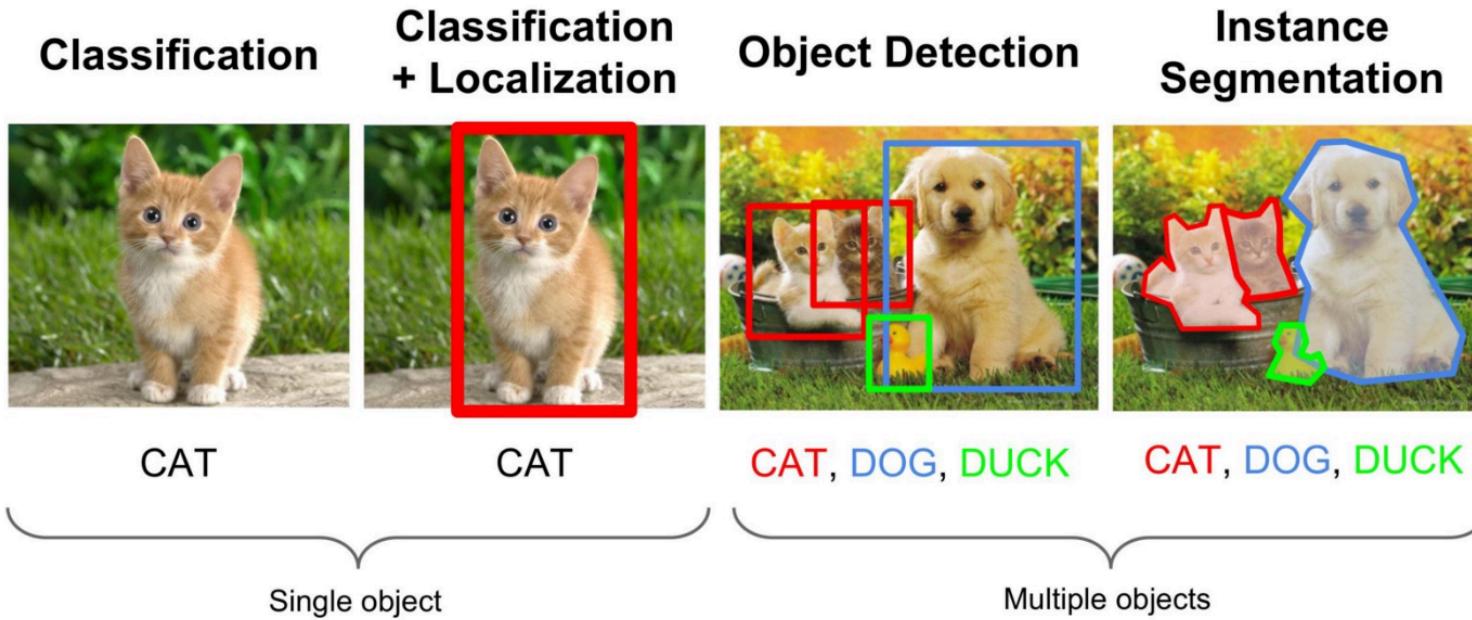


R-CNN & Fast R-CNN & Faster R-CNN

허성실

Object detection

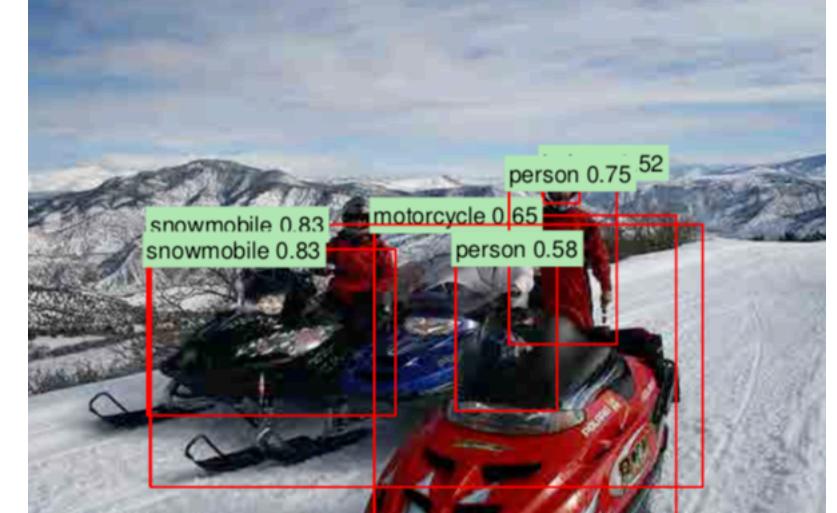


- CNN을 이용하여 사람이 직접 분류한 결과보다 더 뛰어난 결과를 주게 되어 좀더 복잡한 Object Detection 시도.
- 입력 영상이 주어졌을 때 영상 내에 존재하는 모든 카테고리에 대해 classification과 localization을 수행하는 것이 object detection.

R-CNN

R-CNN의 목표

- Input: image
- Outputs: Bounding box + class labeling



- 그렇다면 어떻게 Bounding Box의 위치를 찾아낼 수 있을까?
-> Selective Search 이용

Selective search



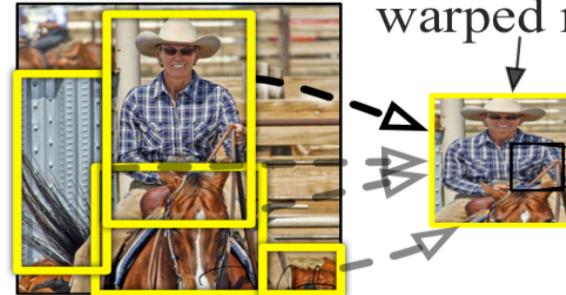
- 픽셀 단위로 유사한 영역(색, 질감 등)을 가진 것들끼리 조금씩 merge해서 최종적으로 region proposal라고 불리는 영역을 약 2000개 가량 생성.
- 이전에는 sliding window 방식을 썼으나 너무 느려서 Region proposal 알고리즘으로 물체가 있을 법한 영역을 빠른 속도로 찾아내도록 함.

R-CNN process

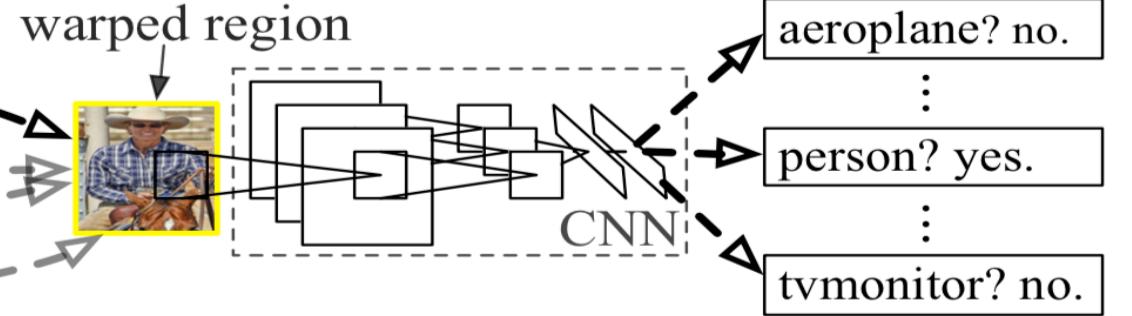
R-CNN: *Regions with CNN features*



1. Input image



2. Extract region proposals (~2k)



3. Compute CNN features

4. Classify regions

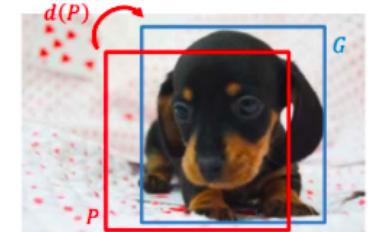
(1) Input image

(2) Selective search로 2000개 가량의 region proposal 을 추출

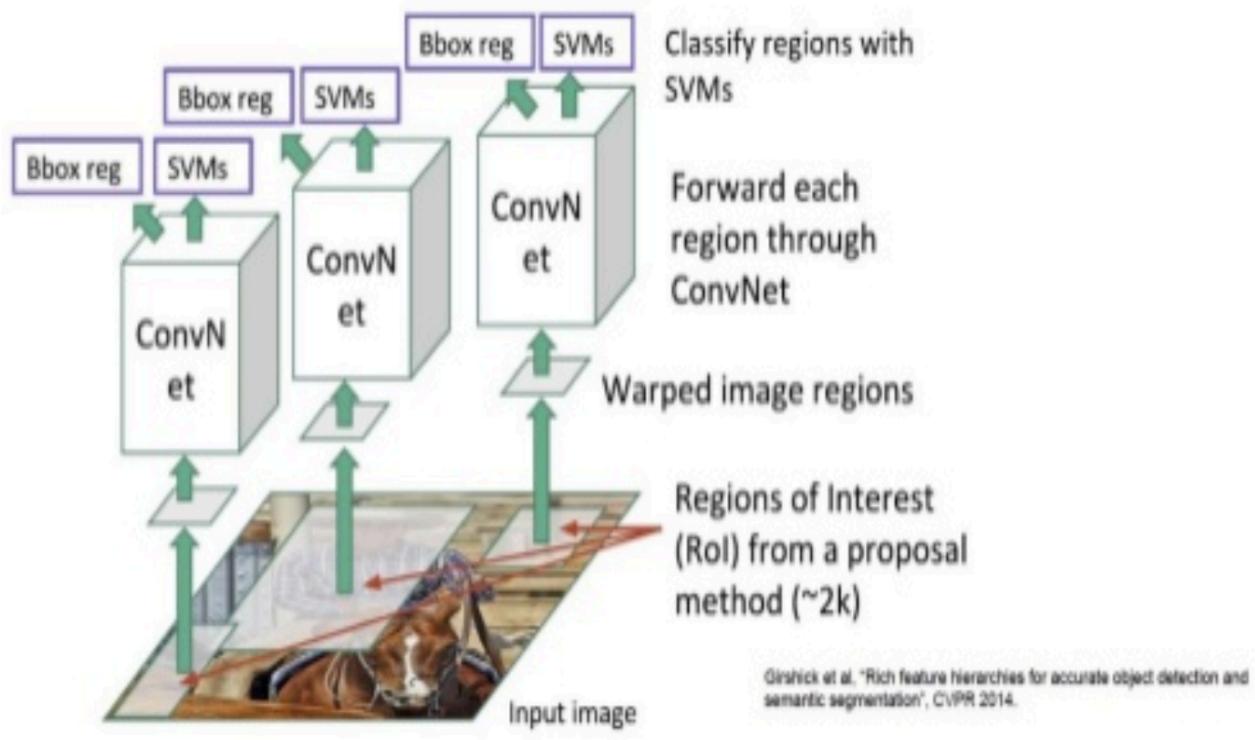
(3) Region proposal warping 하여 크기 맞춤.(fully connected 사용하므로)

(4) CNN으로 각 proposal의 feature 추출

(5) SVM으로 classification, bounding box regression(linear 모델 사용)



R-CNN 문제점

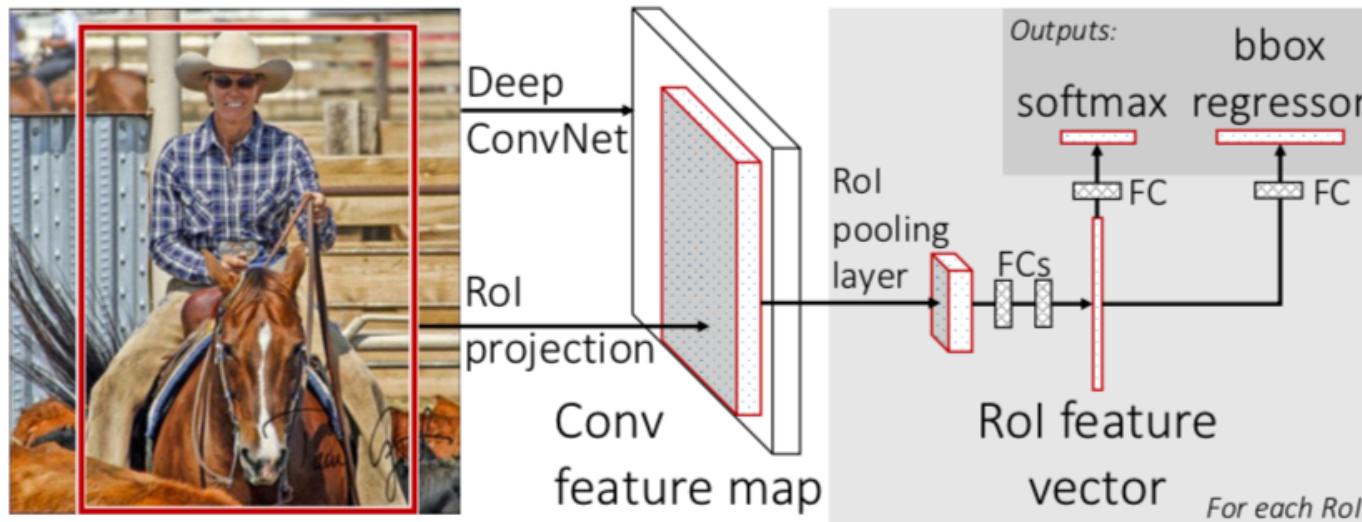


1. 속도가 느림 : 모든 region proposal에 대해 CNN 연산

2. 학습 과정이 복잡, 3개의 모델을 각각 학습 시켜야 함: Feature inference + Classification + Bounding Box regression

Fast R-CNN

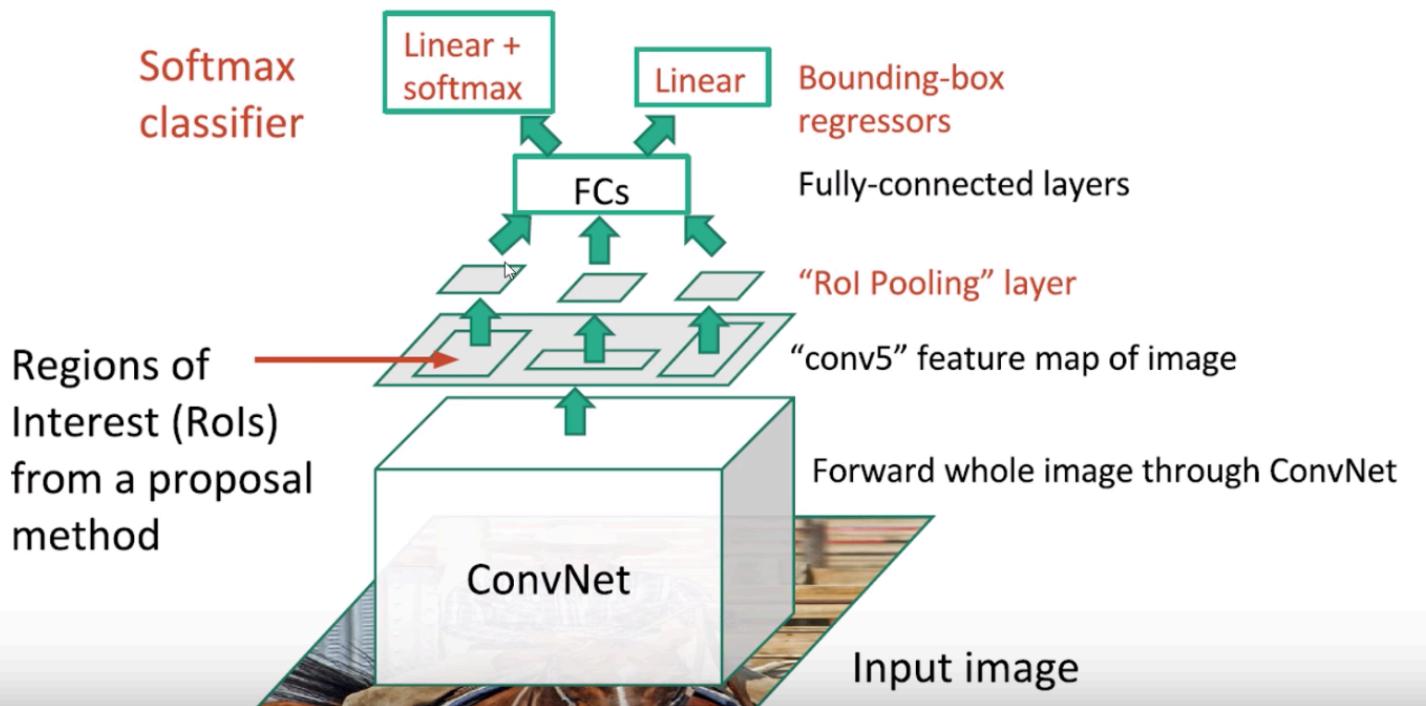
R-CNN 개선



- R-CNN의 속도를 개선한 연구
- 세 개의 모델을 하나의 네트워크로 합침

Fast-R-CNN process

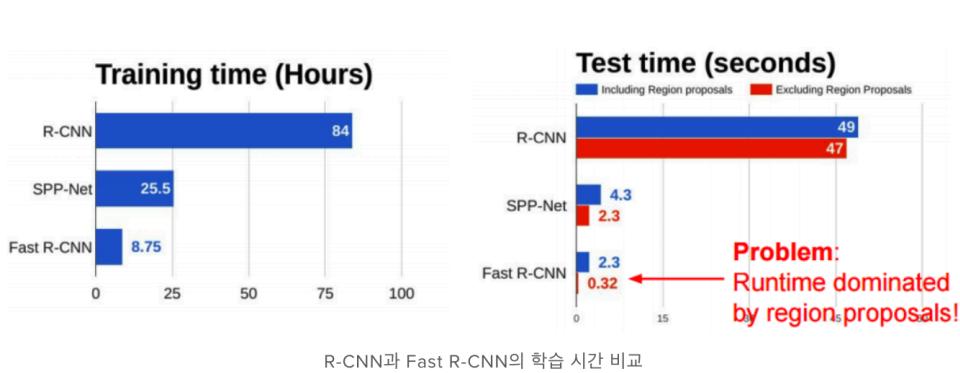
Fast R-CNN



- 1. Input image
- 2. CNN을 통해 Feature map 추출과 동시에 RoI 추출(Selective search 사용, convolution 한번만 수행)
- 3. 추출된 Feature map에 RoI를 Projection
- 4. Bounding box마다 RoI pooling을 하여 크기 맞춘 후, fully connected layer에 넣음.
- 5. Softmax classifier 와 Bounding box regressor로 결과물 출력

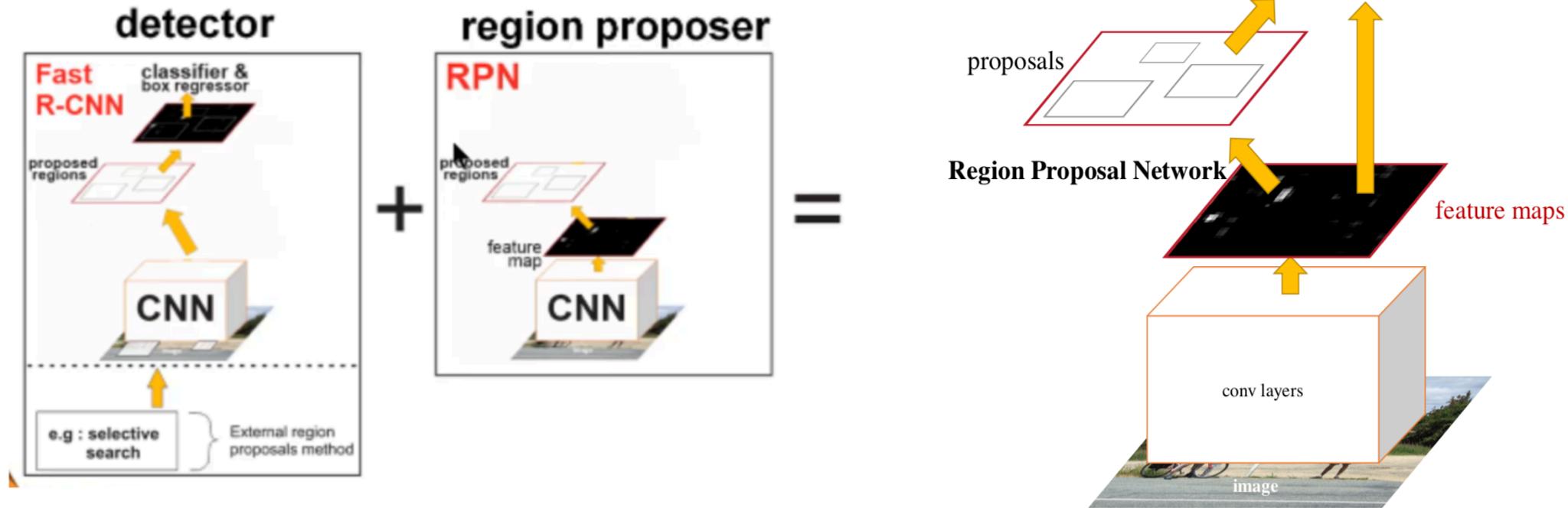
Fast R-CNN 특징 및 문제점

- 특징
 - 같은 image 의 proposal 들이 convolution layer 공유
 - Warping 대신 RoI pooling 사용하므로 이미지 손실이 줄어듬
 - 전체 네트워크가 end to end로 한번에 학습
 - R-CNN 보다 더 빠르고 정확해짐
- 문제점
 - Selective Search 연산량이 많아 전체 성능의 bottleneck이 됨.



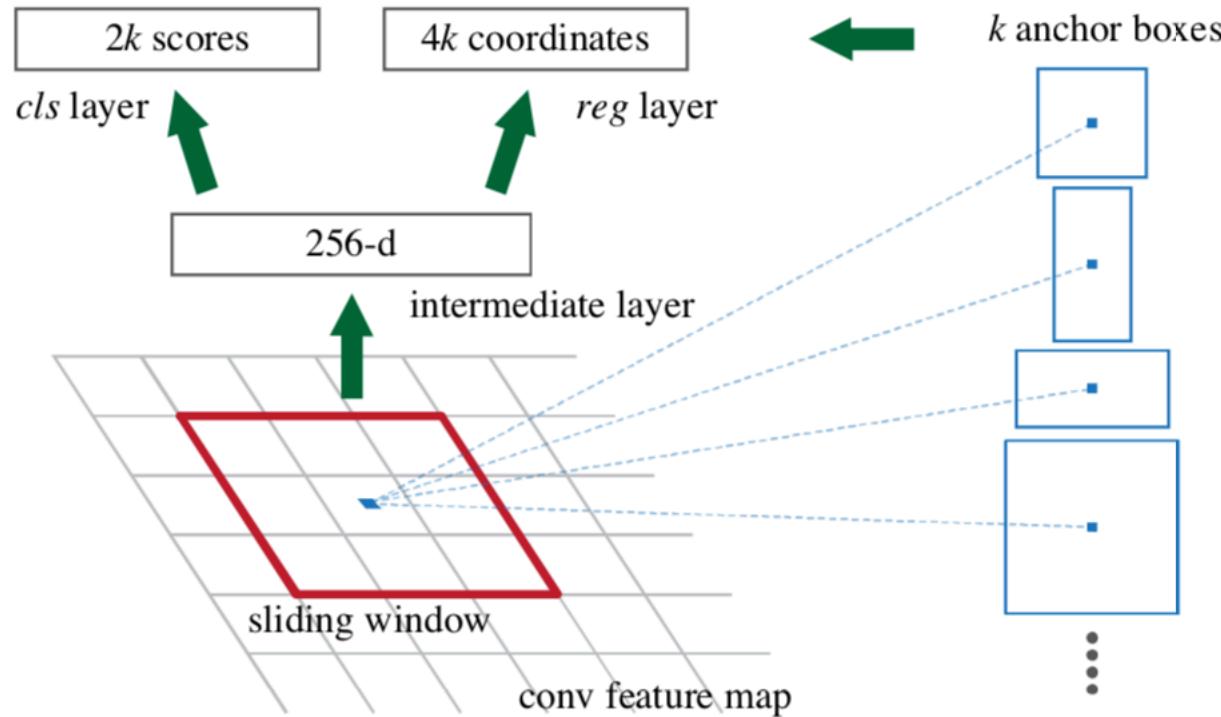
Faster R-CNN

Fast R-CNN 개선



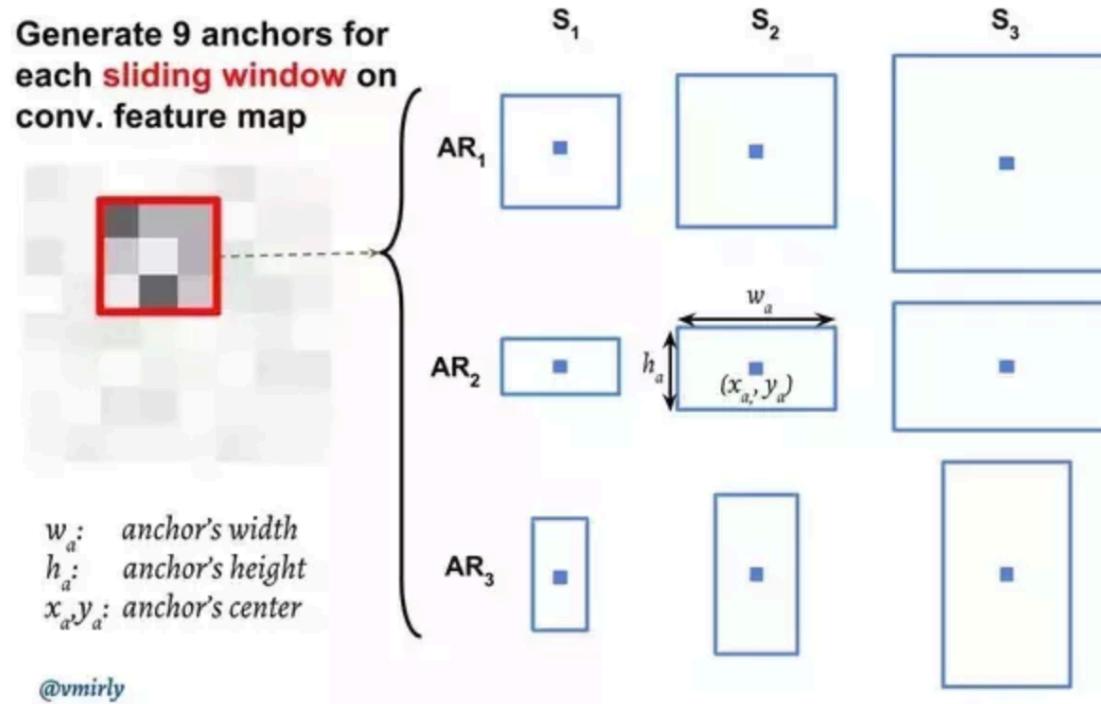
- Bottleneck이었던 Selective search 알고리즘을 빼고, CNN결과를 region proposal에 이용해 보려는 시도
- CNN 내부에 region proposal을 생성할 수 있는 Region Proposal Network(RPN 추가)

RPN 네트워크



- 1. base network에서 생성한 feature map 위에 $n \times n$ spatial window(주로 3×3)을 슬라이드 시킴.
- 2. 각각의 sliding-window가 찍은 지점마다, 한번에 여러 개의 region proposal을 예측하게 됨.

RPN 네트워크

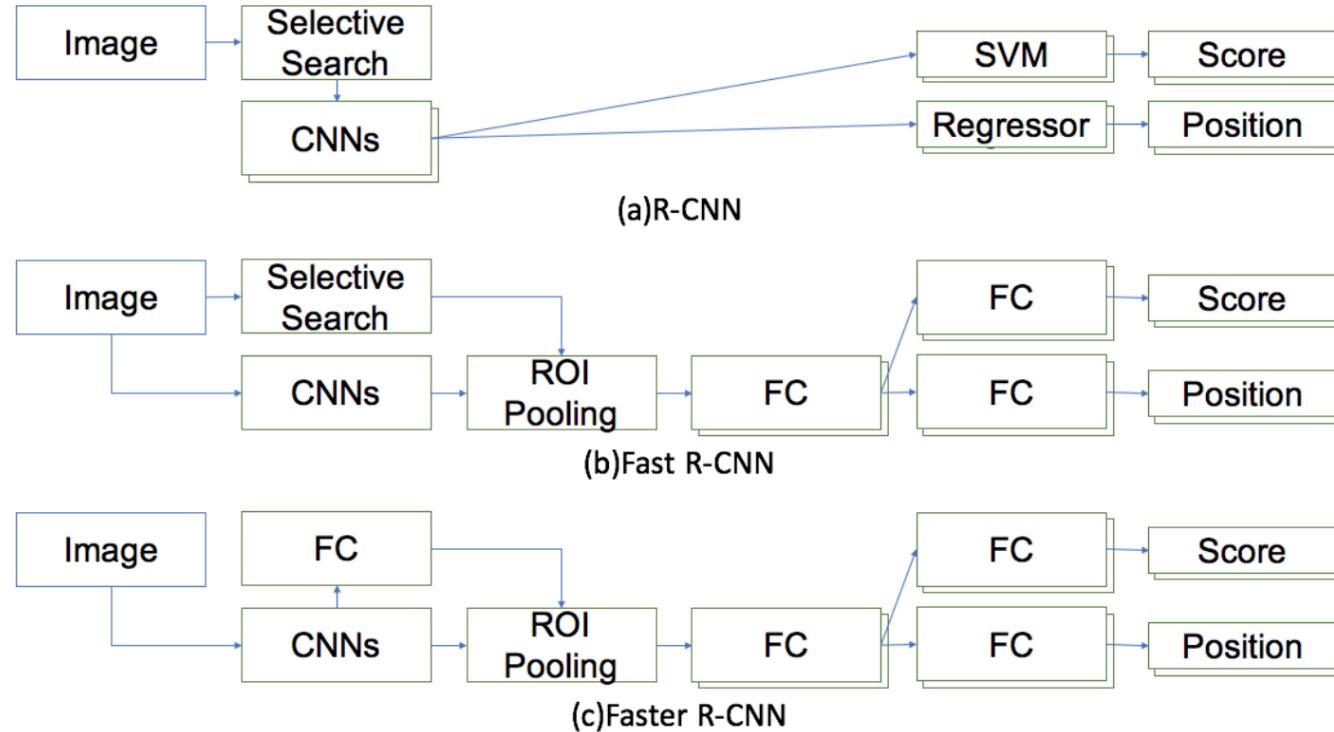


- 각 sliding window의 지점마다 9개의 앵커가 존재(3개의 서로 다른 종횡비와 3개의 서로 다른 크기가 조합). -> 이는 수많은 윈도우의 크기로 sliding하지 않기에 기존 sliding window보다 연산량도 훨씬 작음.

Faster R-CNN의 문제점

- ROI pooling 할 때 ROI의 크기와 CNN에서 정해진 feature map의 크기가 나눴을 때 정수배로 떨어지지 않으면 버림 또는 반올림을 해야하는데 여기서 약간의 오차가 생김. 정확한 좌표가 필요한 곳에서 문제가 생길 수 있음.

Summary



| | R-CNN | Fast R-CNN | Faster R-CNN |
|--------------------------------------|------------|-------------|--------------------|
| Test time per image (with proposals) | 50 seconds | 2 seconds | 0.2 seconds |
| (Speedup) | 1x | 25x | 250x |
| mAP (VOC 2007) | 66.0 | 66.9 | 66.9 |

reference

- 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.
- <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852>
- 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.