Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

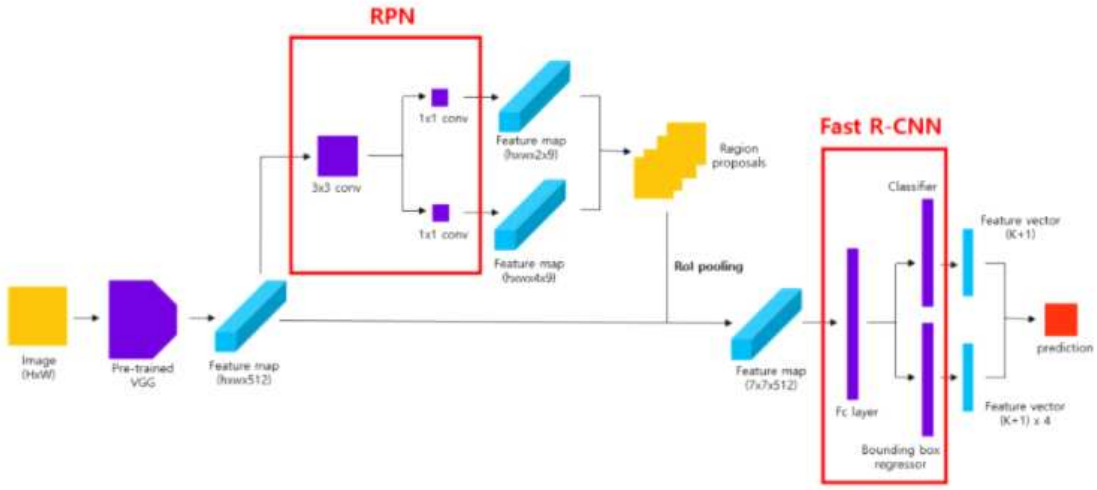
(Faster R-CNN : 지역 제안 네트워크를 통한 실시간 객체 감지)

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations.
Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck.
In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features---using the recently popular terminology of neural networks with 'attention' mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model, our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

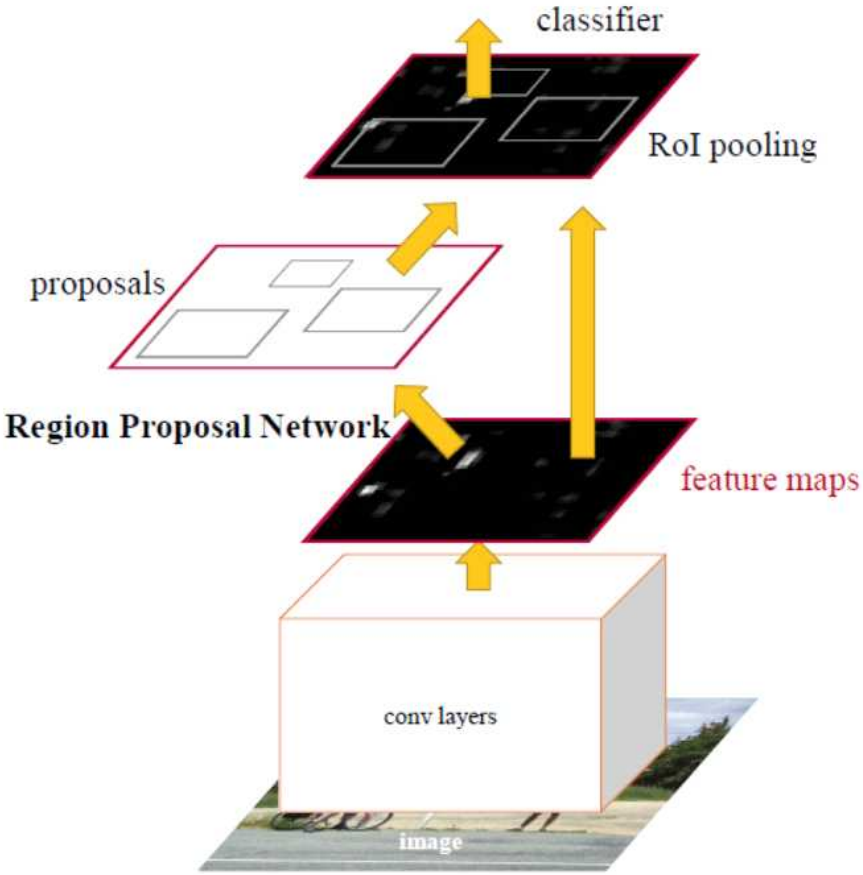
최첨단 od는 의존한다. 물체 위치를 가정하기 위해 지역 제안 알고리즘을
SPP-net Fast R-CNN과 같은 발전으로 bottleneck로부터 지역 제안 계산이 나타나었고
,이러한 탐지 네트워크의 실행 시간이 줄어들었다.
이번 작업을 통해, 우리는 소개한다. RPN을 이것은 공유하다 전체 이미지의 convolution

간단하게 요약



전체적인 shape는 다음과 같다.

1)
R-CNN, Fast R-CNN에서 쓰던 region proposal을 구하는 방식인 selective search는 병목 현상을 일으켜 많은 시간을 사용한다. 따라서 위 논문에서는 RPN(Region Proposal Network) 방식을 이용하여 region proposal을 출력한다. RPN은 CNN을 기반으로 둔 모형이다. 위 CNN은 detection network와 함께 사용된다. 하나의 CNN으로 region proposal을 생성하고 object detection을 수행하는 것이다. 논문에서 ZFnet(AlexNet과 비슷한 구조를 취하지만 첫 번째와 두 번째 layer의 구조가 다른 모형) 또는 VGGnet을 사용하였다.

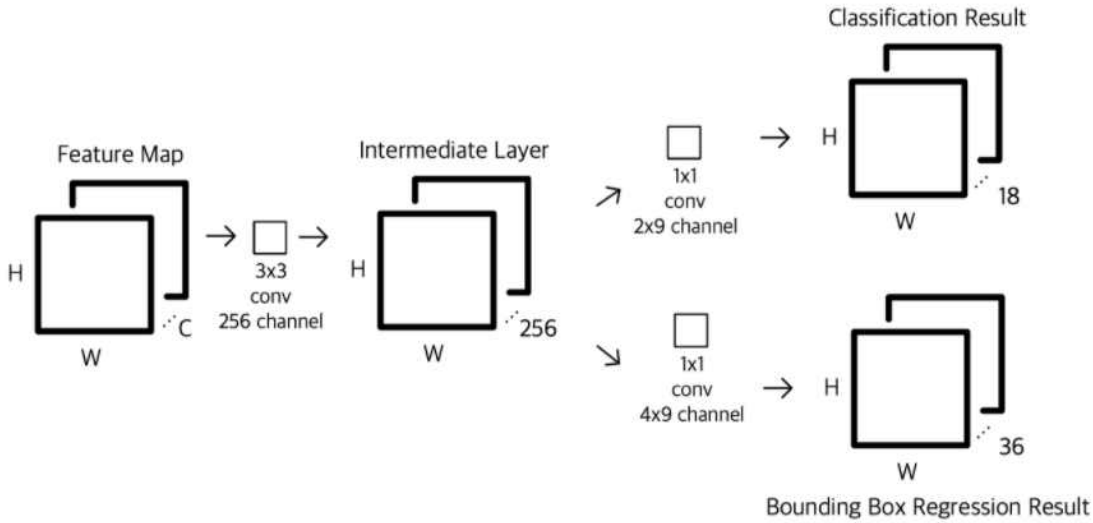


Faster R-CNN

region proposal을 생성하는 과정은 다음과 같다.

1. image를 conv층을 통과시켜 feature map을 얻는다.
2. feature map의 각 위치에서 sliding window를 사용한다.
3. 각 위치에 대하여 region proposal을 생성하기 위해 k개의 anchor box(sliding window처럼 크기가 정형화 된 것이 아닌 scale 및 aspect등을 효용할 수 있는 box)가 사용된다.(만약 feature map의 size가 W*H라면 W*H*k개의 anchor box가 생성됨을 알 수 있다.)
4. cls layer는 k개의 박스에 대해 객체가 있는지 없는지 (0 or 1) score를 출력한다.
5. reg layer는 k개의 박스의 좌표 정보인 4k를 출력한다.

위 과정을 조금 더 자세히 풀면 다음과 같다.

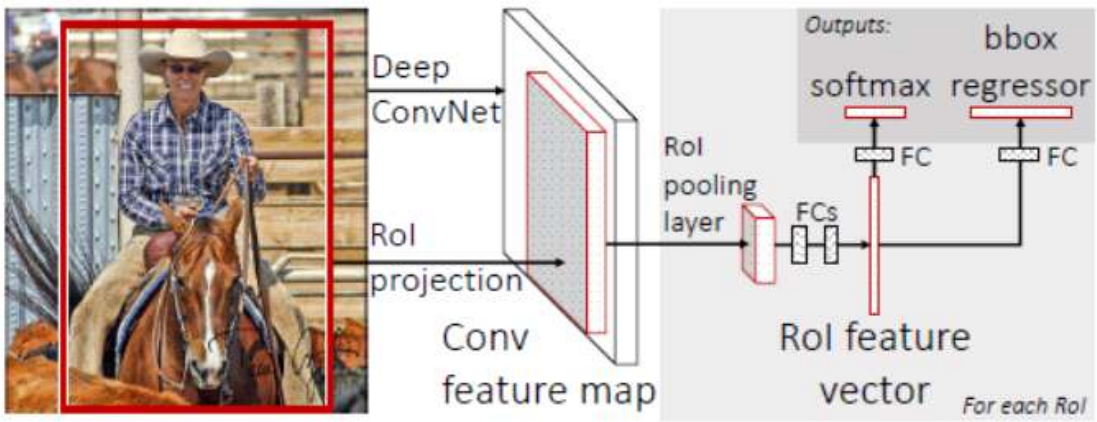


feature map에서 sliding window를 사용하여 intermediate layer를 생성한다. 그 후 1x1 conv 층을 적용하여 cls layer와 reg layer로 나누어 계산한다.

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

loss function은 fast R-CNN처럼 multi-task loss function을 적용하면 된다.

그 후 detection을 위한 network는 다음과 같다.



RPN을 통해 생성된 region proposal을 입력으로 받고 ROI pooling이 수행되어

일정한 크기의 vector를 생성한다. 이 vector는 2개의 fc layer로 입력된다.

이를 요약하여 크게 4가지 스텝으로 학습 시키면 된다.

1. Imagenet pretrained model을 불러와 RPN을 학습한다.
2. 학습된 RPN에서 생성된 proposal을 사용하여 뒤에 있는 detection network(Fast R-CNN)을 학습한다.
3. Fast R-CNN과 RPN을 불러와 CNN 가중치를 고정시킨 후 RPN을 학습한다.
4. CNN의 가중치를 고정시킨 채로 Fast R-CNN을 학습시킨다.

이렇게 계속 반복적으로 학습 시키면서 loss를 줄여나가주는 느낌이다.

Reference

논문 : https://arxiv.org/pdf/1506.01497.pdf

https://deep-learning-study.tistory.com/464

https://velog.io/@skhim520/Faster-R-CNN-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-%EB%B0%8F-%EC%BD%94%EB%93%9C-%EA%B5%AC%ED%98%84