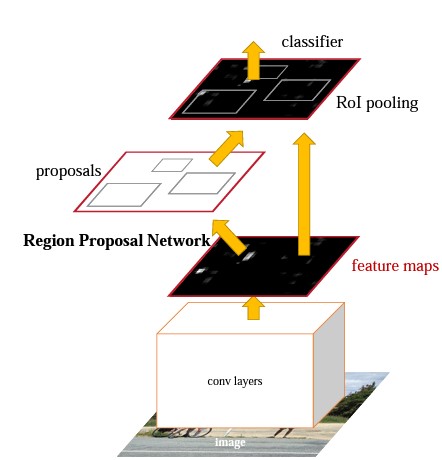
Faster R-CNN 논문 리뷰

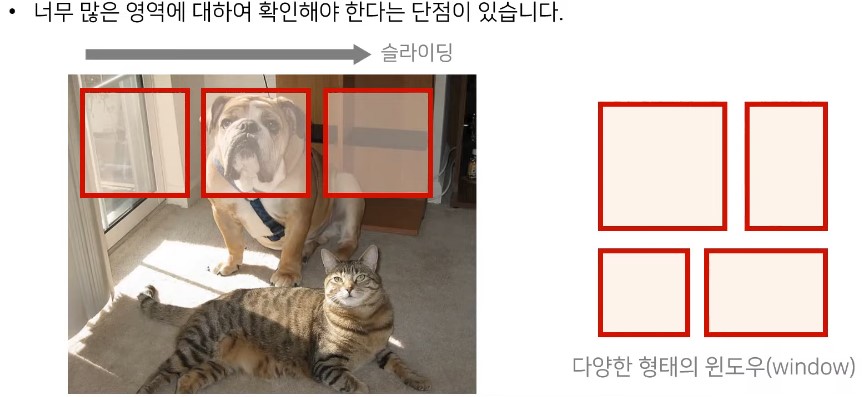
기존에 컴퓨터 비전에서는 CNN을 많이 쓰고 있었는데 이는 한 객체만 있는 이미지를 판별하는 용도로 만들어졌다. 그래서 여러 개의 객체를 판별하기 위해 만들어진 것이 R-CNN이다. 이 모델은 yolo와는 다르게 2 stage detector인데. 이를 간단하게 설명하면 먼저 Region of Interest를 선별(객체가 있을 장소를 ‘대략적으로’ 파악)한 후 이로 인해 얻어낸 RoI를 CNN으로 판단하여 object detection(어떤 객체인지, 정확한 위치는 어디인지 등을 판별)을 하며 이렇게 2단계를 걸쳐 object detect하는 것을 2-stage detector라고 한다. 참고로 yolo는 이 과정을 동시에 진행하는 1-stage detector이다.

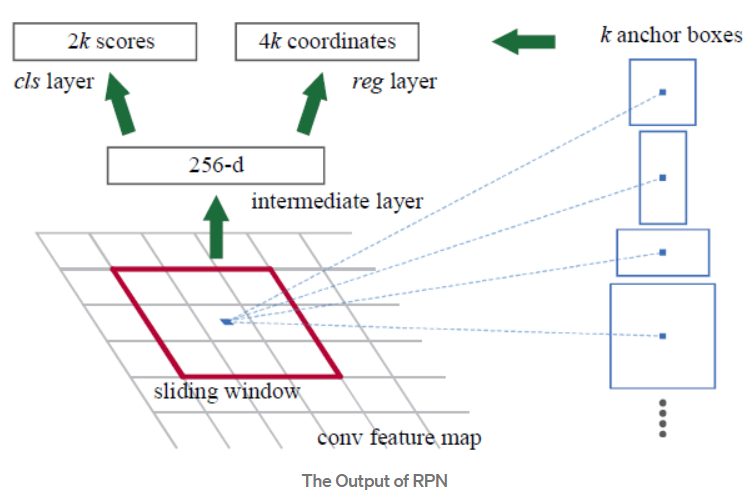
Faster R-CNN의 이 전 버전인 Fast R-CNN은 Selective search라는 기법을 사용하며 이는 CPU에서 돌아가기 때문에 속도가 느린 단점이 있다. 이 후 Region proposal을 찾고 feature map을 뽑기 위해서 CNN을 한 번만 거치는 것이 장점이다. 이 후엔 기본적인 CNN을 이용하여 Softmax layer를 거쳐 각각의 probability를 구함

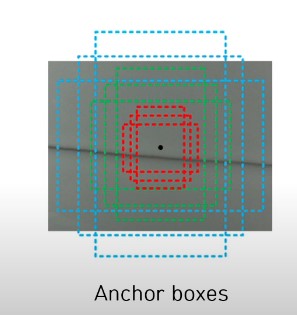
Faster R-CNN은 연산을 GPU상에서 할 수 있는 딥러닝 모델인 Region Proposal Network(RPN)을 이용한다. 또한 이 논문에서는 CNN의 모델은 ZFNet 또는 VGGNet을 사용하였다. RPN이 region proposal을 생성하는 과정은 다음과 같다.



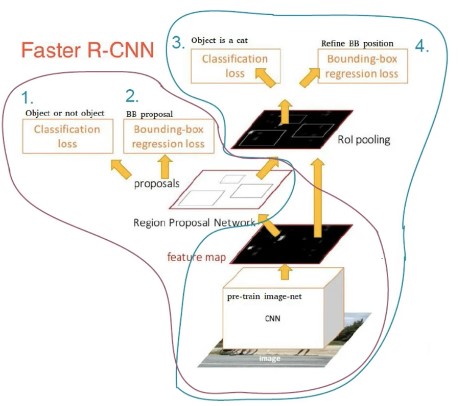
1. 이미지가 convolution layers를 통과하여 feature map이 추출된다. 여기서 추출된 feature map은 공유가 되는 것.
2. Feature map의 각 위치에서 sliding window(3x3 filter)를 사용한다



1. 각 위치에 대하여 region proposal을 생성하기 위해 k개의 anchor box가 사용됨. 128, 256, 512 3개의 scale과 1:1, 1:2, 2:1 3개의 aspect ratio를 사용하여 총 9개의 anchor box를 사용하였다. 결국 proposals의 단계는 물체의 존재 유무를 예측하고 만약 있다면 정확히 어떤 바운딩 박스안에 존재하는지를 예측하는 것
2. classification layer는 k개의 박스에 대해 객체가 있는지 없는지에 대한 2k score를 출력한다. 물체가 존재하면 1, 존재하지 않으면 0.
3. Regression layer는 k개의 박스의 좌표(박스중앙, x, y 좌표, w, h정보를 출력함



4-step alternating training



 1. 맨처음에 convolution을 통한 feature map을 받는 것이 아닌imagenet pre-trained model을 불러와서 RPN을 학습합니다. 여기서 나온 bound box가 output이 될 것임

 2. 1번에서 나온 output과 아래쪽에 있는 convolution network를 거친 feature map 이 2개를 받아 classification과 regression을 진행하여 output을 냄

 3. detector network와 RPN을 불러온 다음 CNN의 모든 가중치를 고정시킨 채 RPN을 학습합니다.

 4. CNN의 가중치를 고정시킨 채로 detection network(Fast R-CNN)를 학습시킵니다.