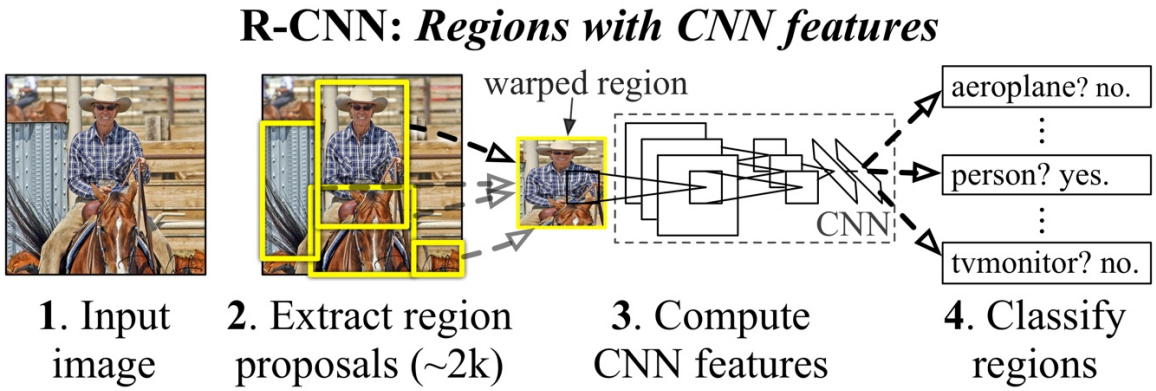
Paper review on Faster R-CNN

Yujin Cheon 2022.11.18

0. Background

R-CNN



R-CNN은 세가지 모듈(Extract region proposal, compute CNN features, classify regions)로 구성되어 있다.

- Extract Region Proposal: 입력 이미지가 들어오면 이미지 내에서 bounding box가 될만한 후보, 즉 candidate bounding box를 선별. candidate bounding box(region)을 선별(proposal)해주는 것을 region proposal이라고 한다.
- Compute CNN features: (227x227 warped image를 input으로 가지는) ConvNet(AlexNet)을 사용해 feature extraction을 함.
    1. region proposals와 ground-truth box의 ***IoU(Intersection of Union)***을 비교하여 IoU가 0.5보다 큰 경우 positive samples, 0.5보다 작은 경우 negative samples로 나눈다.
    2. 이렇게 나눈 후 positive sample 32개 + negative sample 96개 = 128개의 이미지로 이루어진 하나의 미니 배치를 만든다.

3. 이렇게 생성된 배치들을 ConvNet에 넣어 fine-tuning을 진행한다
4. 128개의 미니배치를 구성한 후 fine-tuning된 ConvNet에 입력하여 4096 dimensional feature vector를 추출

- Training: SVM을 사용한 classification과 bounding box를 사용한 localization으로 나뉨.
  - Classification: SVM은 2진 분류를 수행하므로 분류하려는 객체의 종류만큼 SVM이 필요하다
  - bounding box regressor: selective search 알고리즘을 통해 얻은 객체의 위치는 부정확할 수 있다. 이런 문제를 해결하기 위해 객체의 위치를 조절해주는 Bounding box regressor가 있다.

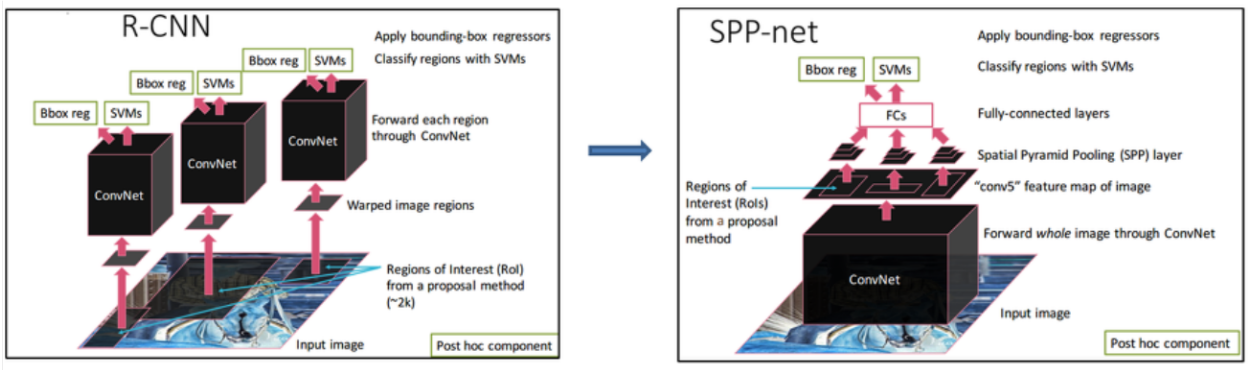
SPP Net

R-CNN은

1) 선별된 모든 candidate bounding box를 ConvNet에 입력해 그 숫자만큼 cnn operation을 하므로 하나의 이미지를 detection하는 데 시간이 많은 소요되어 실시간 성능을 보장하기 어려우며,

2) warping과정에서는 이미지 왜곡이 일어날 수 있다.

이 두 문제를 해결하기 위해 나온 것이 SPP Net이다.



- Reduce CNN operation

선별한 모든 candidate bounding box에서 cnn operation을 수행하는 대신, 전체 입력 이미지에 대해 cnn 작업을 수행하고 다섯번째 conv layer에 도달했을때 conv5 feature map을 가지

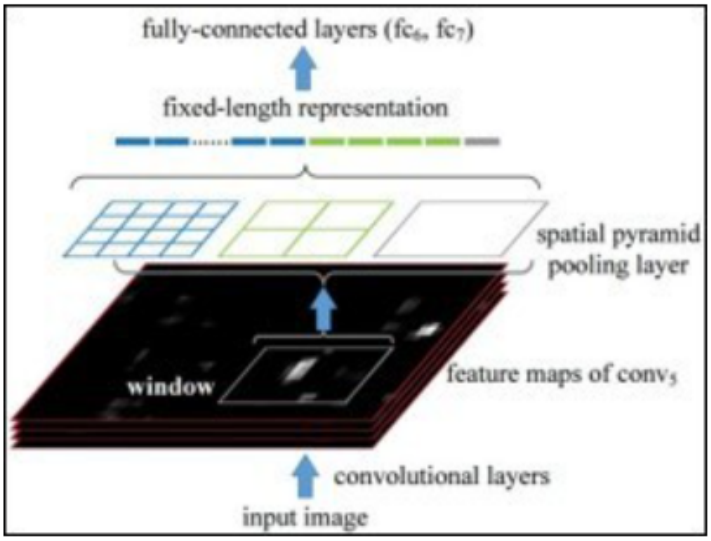
고 region proposal 방식으로 candidate bounding box를 선별한다. 이렇게 cnn operation은 1번으로 단축된다.

candidate bounding box = region of interest (RoI)

- Remove warping process and avoid distortion

SPP-Net은 warping으로 인한 왜곡을 없애기 위해 spatial pyramid pooling을 사용한다.

- Spatial Pyramid pooling(in SPP-Net)



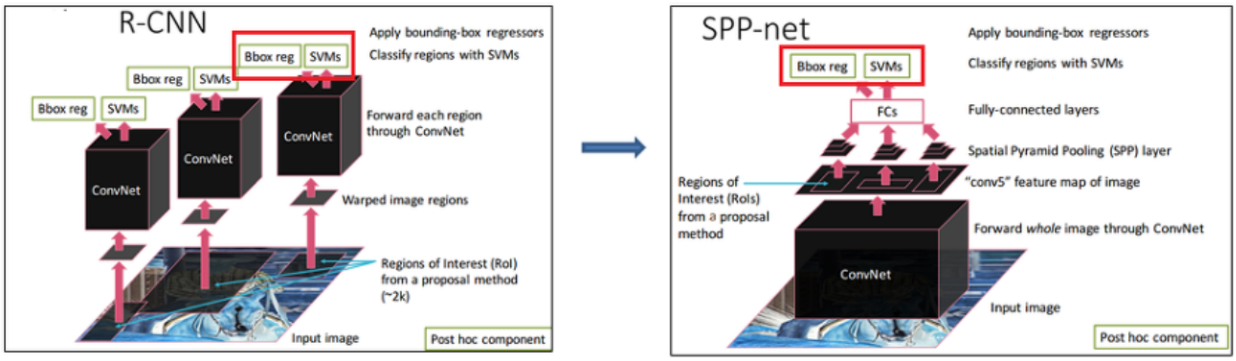
conv5 layer feature map에서 선별된 RoI(Region of Interest)가 4x4, 2x2, 1x1 사이즈가 되게끔(window size와 stride를 설정해) max pooling 해준다. ex) 13x13 size RoI에서 2x2 spatial bin을 얻기 위해서는 7x7 window에 stride 6로 pooling을 해야한다.

4x4, 2x2, 1x1 spatial bin을 얻으면 이것들을 flatten 하여 16+4+1=21개의 feature를 fc layer로 넘긴다.

Recap on Fast R-CNN

Cons of RCNN & SPP-Net

1) **Three stage pipeline**: 학습을 세 번 나눠서 하기 때문에 시간 소모가 큼.

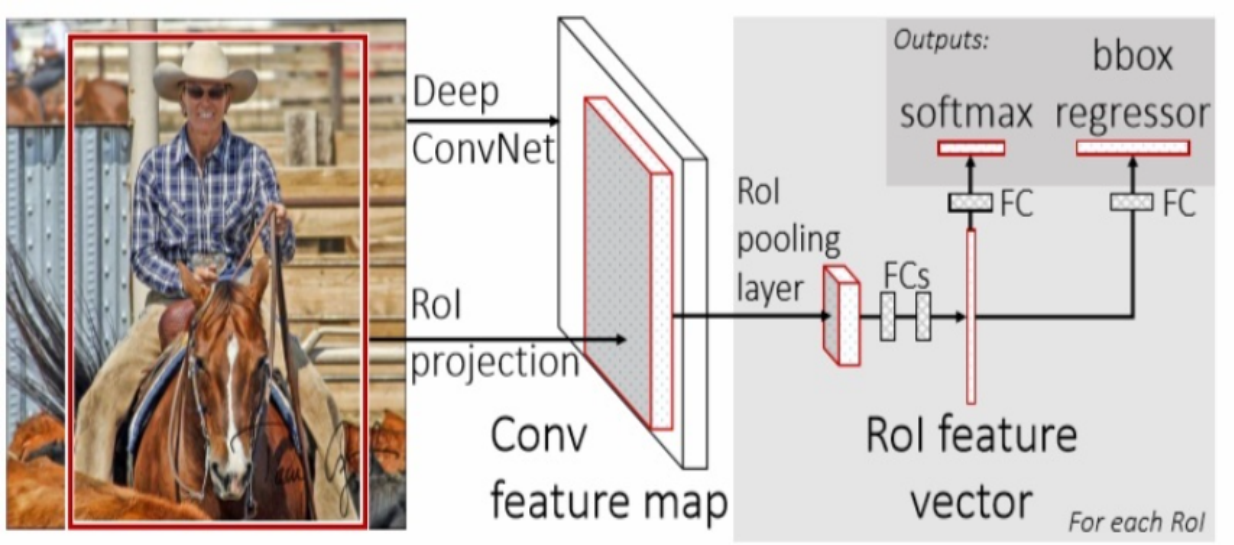


<R-CNN과 SPP-net의 학습 방식의 공통점>

1. CNN 모델을 사용할 때 ImageNet에 학습된 pre-trained 모델을 쓴다. 그리고 이 pre-trained모델을 Pascal VOC 이미지 데이터셋에 finetuning 한다.
2. finetuning 후 CNN끝의 softmax layer를 제거하고 SVM을 설치한다. SVM에 Pascal VOC 데이터셋을 학습시킨다.
3. classification이 끝난 후 bounding box regression을 한다.

2) **Feature vectors stored in hard disk**: ConvNet을 통해 얻은 feature vector들은 하드디스크에 저장된다. SVM과 bounding box regression학습을 하기 위해서는 CPU에서 하드디스크에 접근해 feature vector를 받아와야 하는데 물리적 거리로 인해 많은 시간이 소요된다.

Fast RCNN Architecture



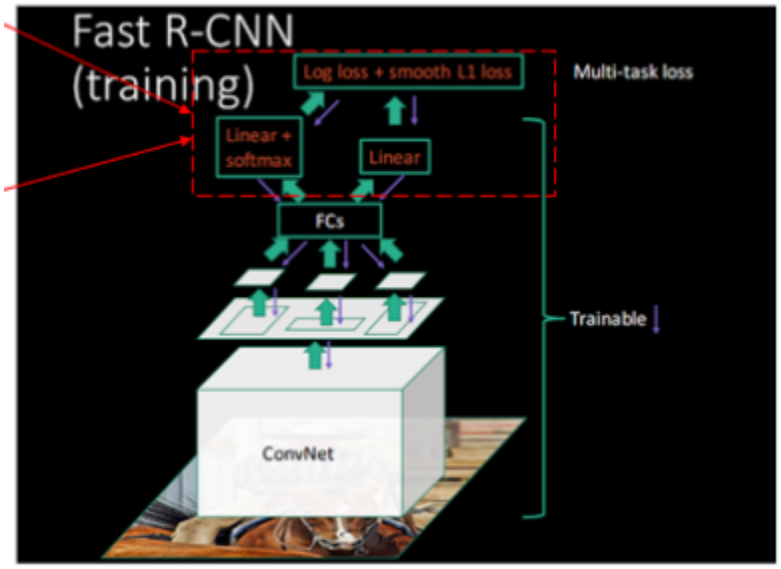
1) **RoI pooling layer**

- Conv layer을 통해 추출된 feature map에서 RoI를 선별한 후, 그 RoI를 모두 7x7 size로 만들어준다. (SPP-net의 Spatial pyramid pooling layer에서 7x7 spatial bin을 만드는 것과 같다.)
- 생성된 7x7 spatial bin은 flatten되어 2개의 FC layer을 통과하여 softmax, bounding box regressor에 사용된다.
- SPP-net에서 1x1,2x2,4x4의 3가지 spatial bin을 이용하는 것은 하나의 객체를 3가지의 resolution으로 학습시키는 것이기 때문에, **7x7의 single spatial bin**을 사용하는 것이 SPP에서 발생하는 overfitting 문제를 피할 수 있다.

2) **Truncated SVD**

- Fast RCNN은 한 번의 cnn operation 후 대략 2000개의 RoI를 선별해 FC layer에 넣기 때문에 FC layer에서의 연산이 많다. 무수히 많은 FC layer의 연산들이 truncated SVD기법으로 compress되어 파라미터 수와 테스트 시간을 줄일 수 있다.
- 네트워크를 compress하기위해 하나의 fully connected layer W를 두 개의 fully connected layer로 대체한다.(→matrix factorization)
- 이렇게 대체하는 것을 통해 uxv사이즈인 weight matrix W의 parameter count를 u x v → t(u+v)로 줄일 수 있다.

Training Fast RCNN



1) Single Training Stage

1-1) One loss function with two multi-task

Fast R-CNN은 bounding box와 classification을 동시에 학습한다. 즉 하나의 loss function을 통해서 multi-task(classification & localization)를 수행하도록 만들었다.

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

$L_{cls}(p,u)$: $classification\ \ loss\ \ function\ =\ log\ \ loss$

$L_{loc}(t^u,v)$: $localization\ \ loss\ \ function$

-classification loss 는 특정 클래스 u에 대한 log loss이다.

-localization loss는 특정 클래스 u에 대한 true bounding box v와 predicted bounding box t의 차로 정의되는 loss이다.

1-2) Fine-tuning

기존의 R-CNN은 ConvNet, SVM, bb regression에 대해 각각 Pascal VOC로 finetuning을 한다. 하지만 fast R-CNN은 SVM을 따로 classifier로 붙여서 쓰지 않고 기존의 cnn classifier

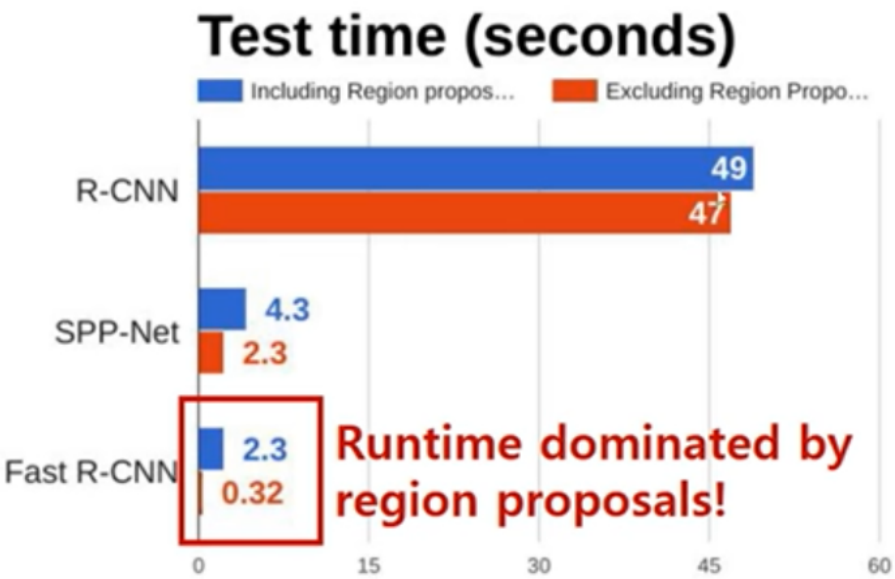
softmax를 그대로 이용하여 학습하기 때문에 Pascal VOC데이터로 finetuning을 한 번만 하게 된다. →single training stage

2) Eliminate disk storage

Fast RCNN은 feature를 하드디스크에 넘기지 않고 학습을 시키기 때문에 classification & localization연산 시간을 절약할 수 있다.

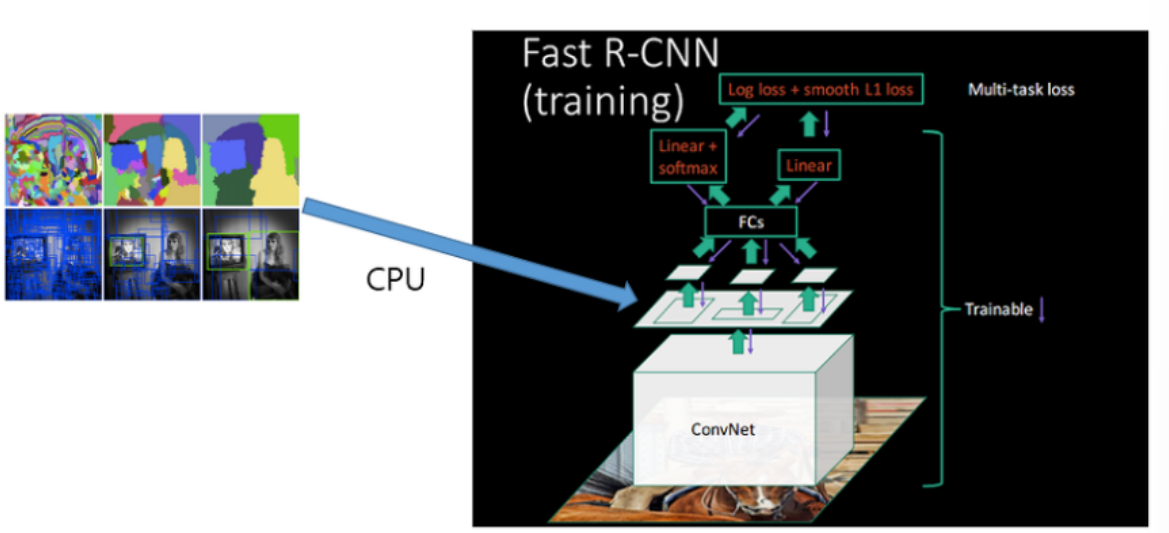
Faster R-CNN

1. Cons of fast R-CNN



- **Hard to perform in real-time(especially with region proposal method)**
  - Fast R-CNN은 R-CNN이나 SPP-Net에 비해 시간이 2.3초로 크게 감소했지만 여전히 **real-time(1/30초)을 수행하는 데에는 무리가 있다**. 무엇보다 2.3초 test time의 대부분을 region proposal이 차지하고 있다.
  - Region proposal은 selective search 방식으로 RoI를 추출하는데, selective search는 외부에서 연산된 것을 Fast RCNN 내부에 있는 feature map에 전달하여 RoI pooling을 하는 것이다. Selective search자체는 cpu에서 연산을 해 시간을 많이 소모

하므로 selective search를 사용하지 않는 region proposal 방식을 고안할 필요성이 제기되었다.



2. Architecture of Faster R-CNN

Fast R-CNN + RPN(region proposal network)

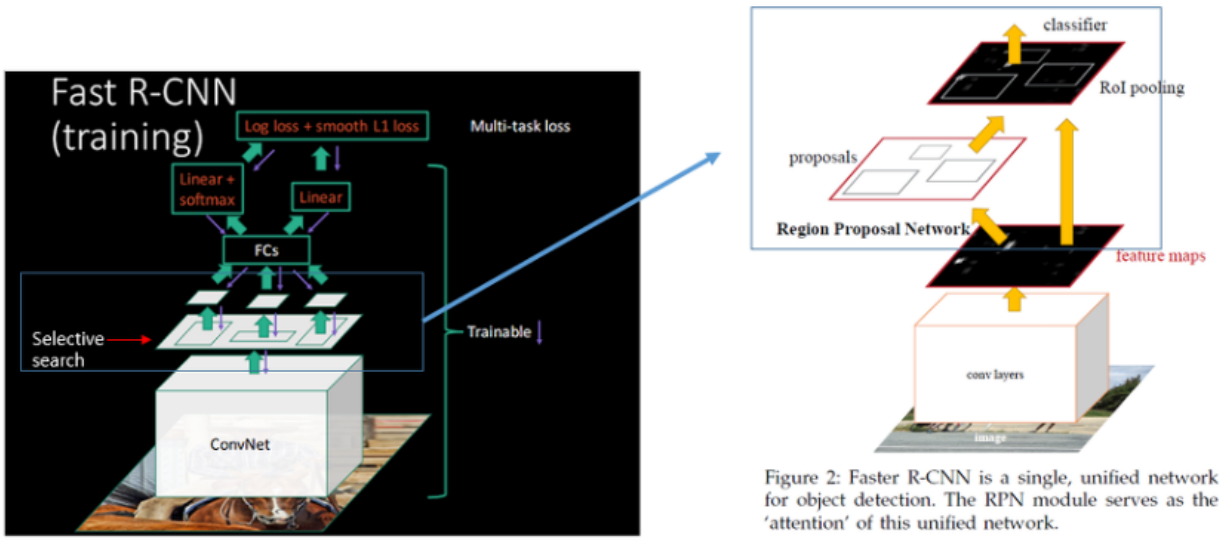
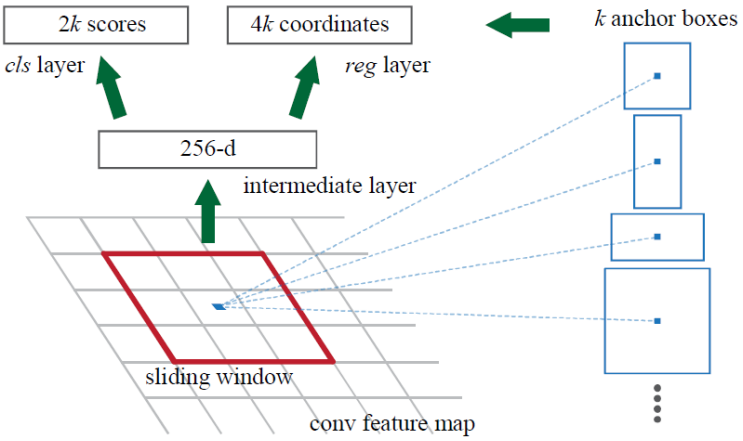


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

Faster RCNN의 구조는 Fast RCNN과 대체로 유사하며, region proposal 방식이 selective search에서 regional proposal network로 바뀌었다.

1) Region Proposal Network(RPN)

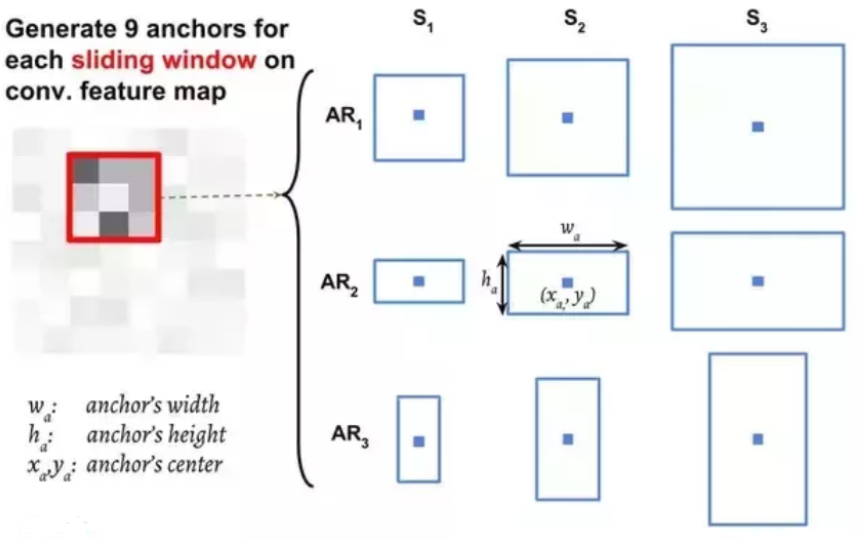


1. CNN을 통해 추출한 13x13x256 feature map에 3x3x256 conv filter를 적용.
2. 3x3 filter의 center지점에서 9개의 anchor box를 생성한다. feature map의 너비와 높이인 13x13부분이 전부다 filter의 center지점이 될 수 있고 center 지점마다 9개의 anchor를 생성한다. → 13x13 feature map에서는 13x13x9의 anchor가 생성된다. 따라서 output 연산량은 256x9이다.

   <9개 anchor box 특징>

   3 different size, 3 different aspect ratio를 적용하여 서로 다른 9개의 anchor box가 생성된다. 이 multi-scale anchor들을 통해 더 정확한 localization을 할 수 있다.

   <Multi-scale anchors>

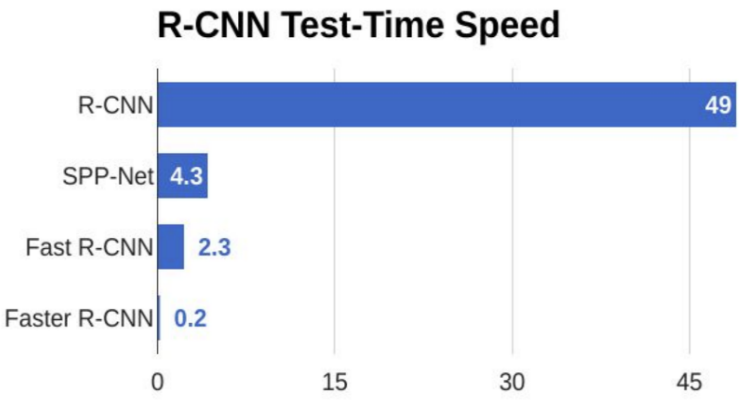


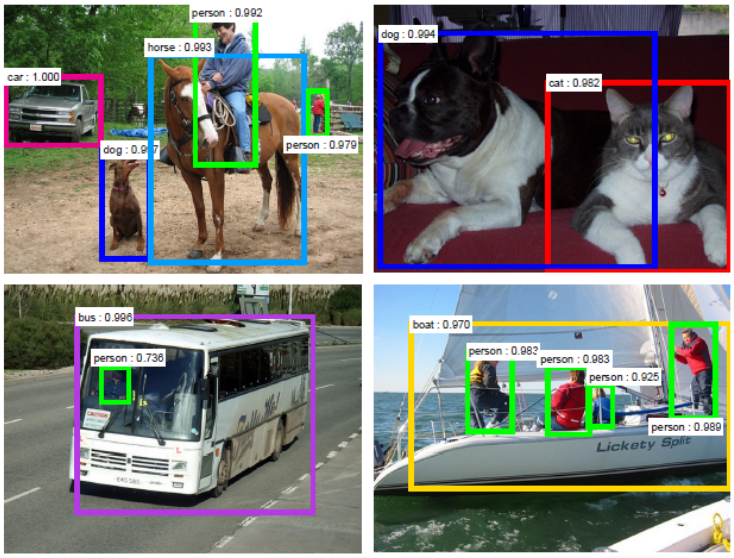
3. 9개의 anchor 각각에 대해서 객체가 포함되어 있는지 아닌 지에 대한 **classification** 작업이 진행된다. 이 작업을 마치면 나오는 연산량은 **256x9x2**이다.
4. 9개의 anchor 각각에 대해서 **bounding box regression**을 한다. 이때 좌표정보로 4개의 인자 (x, y, 높이, 넓이)가 사용되기 때문에 이 작업을 마치면 나오는 연산량은 **256x9x4**가 된다.
5. classfication과 bounding box regression은 동시에 수행되므로 RPN의 loss function은 각각의 loss를 더해주어 구해진다. 그렇기 때문에 하나의 3x3 conv filter 영역으로부터 생성되는 **최종 output** 파라미터의 수는 **256x9x2 + 256x9x4** 이다.

2) Translation invariant anchor

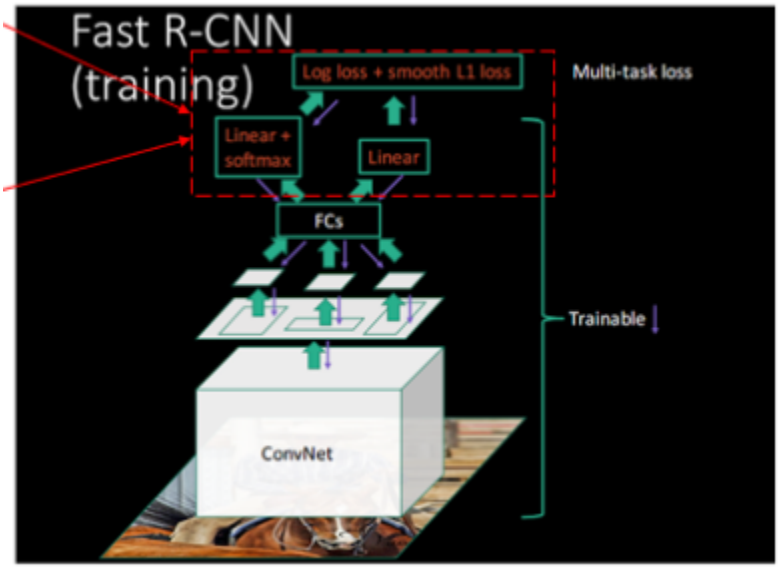
객체의 translation, reflection, rotation 등의 이동이 있더라도, anchor box가 이동한 객체를 찾는데에는 영향을 미치지 않는다.

RPN방식이 selective search(ss)방식보다 더 적은 RoI를 추출할 수 있게 되어 시간도 굉장히 절약되고 mAP도 더 높게 나왔다.





3. Training of Faster R-CNN



ConvNet을 통과한 feature map 위에는 객체 정보가 담긴 여러 ground truth가 있다. RPN의 동작 방식은 이 feature map위에서 3x3 conv filter로 이동하면서 여러 anchor들이 생성하는 것이다. (보통 하나의 ground truth위에 여러 anchor들이 생성됨.)

생성된 anchor들을 IoU가 높은 순서대로 positive example에 넣고 IoU가 0.3이하이면 negative example이 된다. 그렇게 FC layer를 통과하고 loss를 거쳐 학습이 진행이 된다.

1) loss function

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

$$t_{\mathrm{x}} = (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}},$$
$$t_{\mathrm{w}} = \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}),$$
$$t_{\mathrm{x}}^* = (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}}^* = (y^* - y_{\mathrm{a}})/h_{\mathrm{a}},$$
$$t_{\mathrm{w}}^* = \log(w^*/w_{\mathrm{a}}), \quad t_{\mathrm{h}}^* = \log(h^*/h_{\mathrm{a}}),$$

pi: anchor i가 객체일 예측 확률(predicted probability)

ti: 예측된 bounding box의 좌표 4개 → (tx, ty, tw, th)

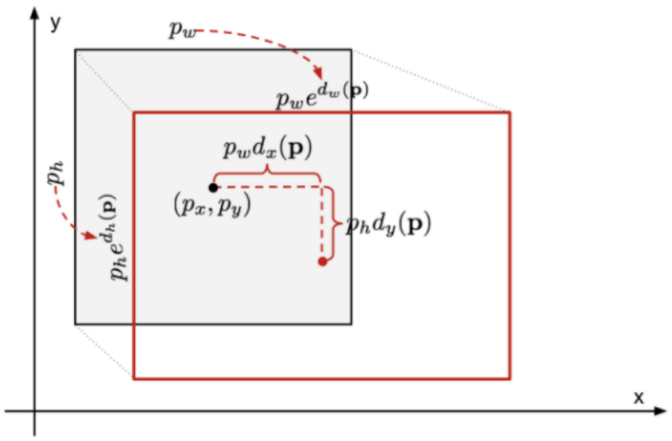
*: 예측이 아닌 ground truth를 의미함.

Lcls: 객체인지 아닌지에 대한 log loss

Lreg: smooth function same as fast R-CNN

Ncls: mini-batch size

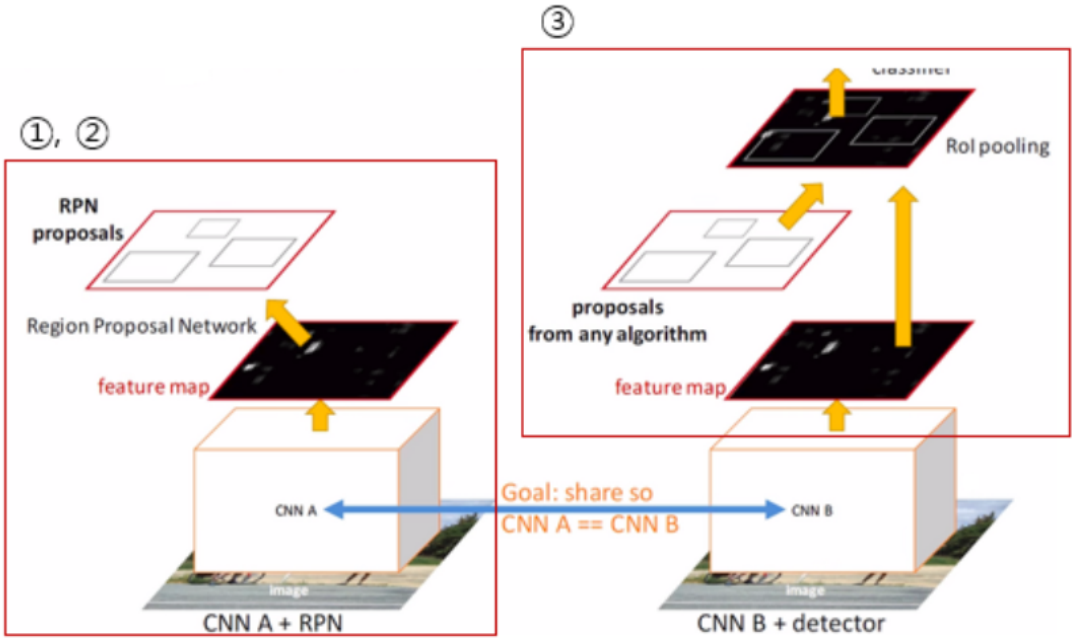
Nreg: the number of anchor locations



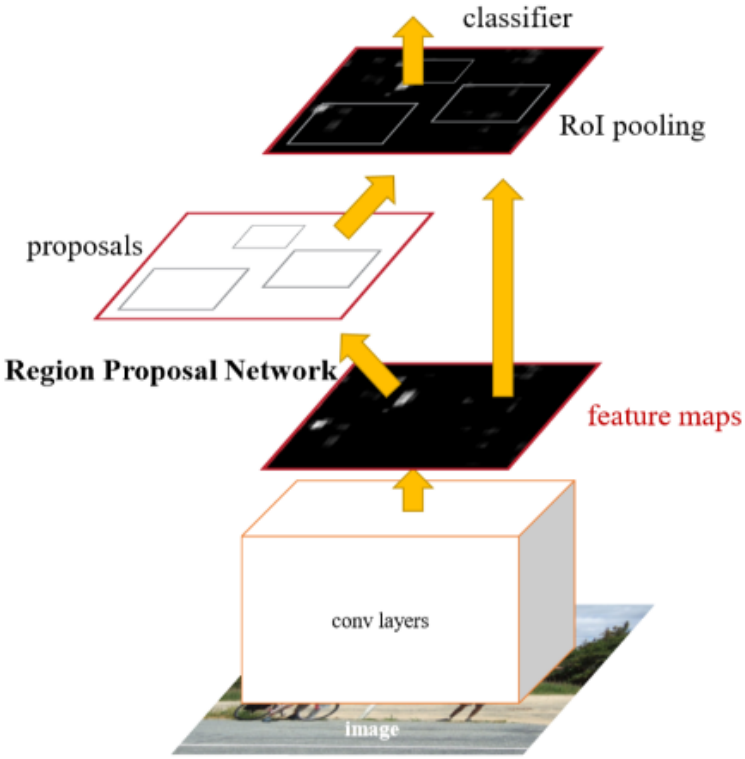
2) sharing features for RPN and fast RCNN

Faster RCNN은 RPN과 Fast RCNN의 조합이라고 볼 수 있다. 하지만 처음부터 RPN을 Fast RCNN에 내장시켜 학습하는 것이 아니라 분리해서 학습을 한다.

1. ImageNet pre-trained Network(M0)를 가져와서 RPN을 ImageNet 데이터로 학습시킨다. RPN=M1
2. pre-trained model M0와 RPN에서 RoI(P1)을 추출한다.
3. pre-trained model M0와 RPN에서 추출한 RoI(P1)을 적용시켜 Fast RCNN을 학습시킨다. 학습된 fast RCNN=M2



4. 학습된 fast RCNN(M2)를 이용해 다시 RPN을 학습시킨다.(RPN=M3) 이때 CNN은 따로 학습하지 않는다. 즉 fast rcnn에서 사용하는 conv layers를 그대로 가져와 적용하고 그 위에서 RPN만 학습시킨다. feature map에서 region proposal network로 가는 filter들만 학습하면 fast RCNN과 cnn을 share하게 된다.



References

논문: https://arxiv.org/abs/1506.01497

https://velog.io/@skhim520/R-CNN-논문-리뷰

https://89douner.tistory.com/89

https://89douner.tistory.com/90

https://89douner.tistory.com/91