Faster R-CNN

↳ true end-to-end ?

Will Cover

1. What's improved ?
   - RPN , Anchor

2. Architecture + Forward Pass

3. All about RPN
   - Anchor box : translation-invariant , pyramids
   - Loss
   - How to train ?

4. How RPN and Detector share feature map ?
   - alternating training

5. Implementation details

1. What's improved ??

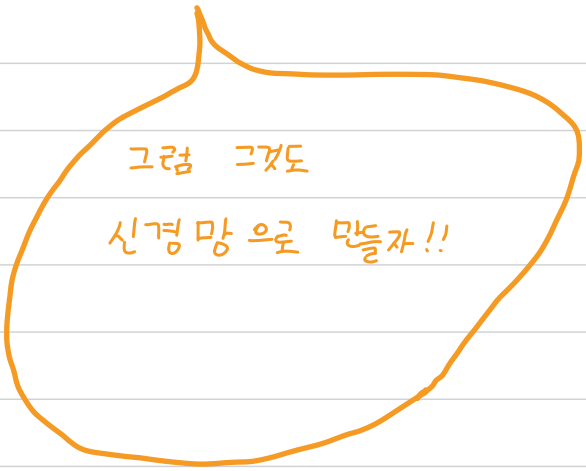
Fast R-CNN 의 한계.

↳ Regional proposal done by Selective - search

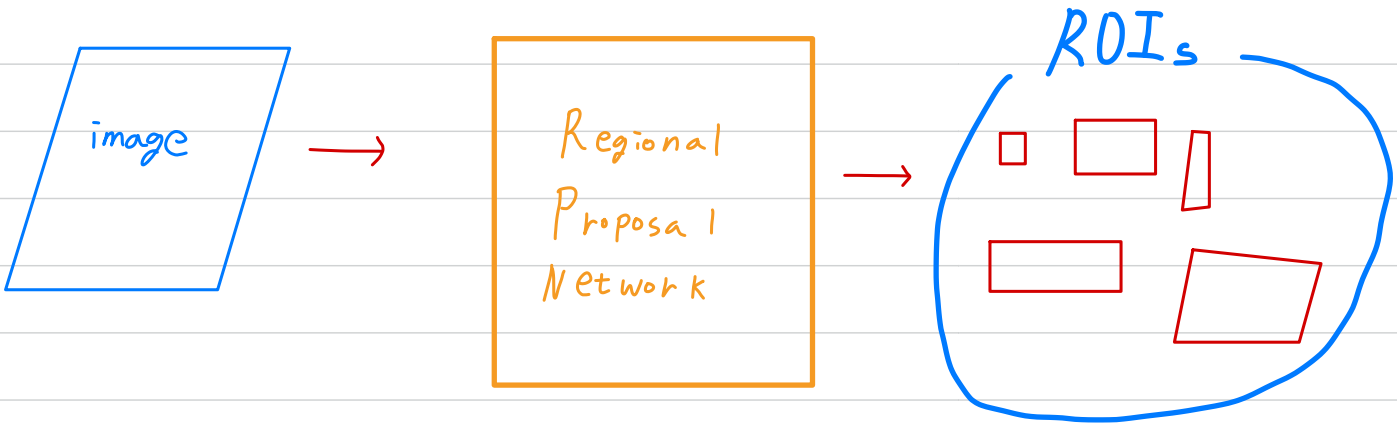
실제 inference-time 2.3초 중 2초가 S.S 에 소요될 만큼

큰 비병목현상을 가져오는 방법 (∵ S.S 는 CPU 환경에서만 가능하기에 느릴수밖에 없다.)

Faster R-CNN 연구팀



RPN



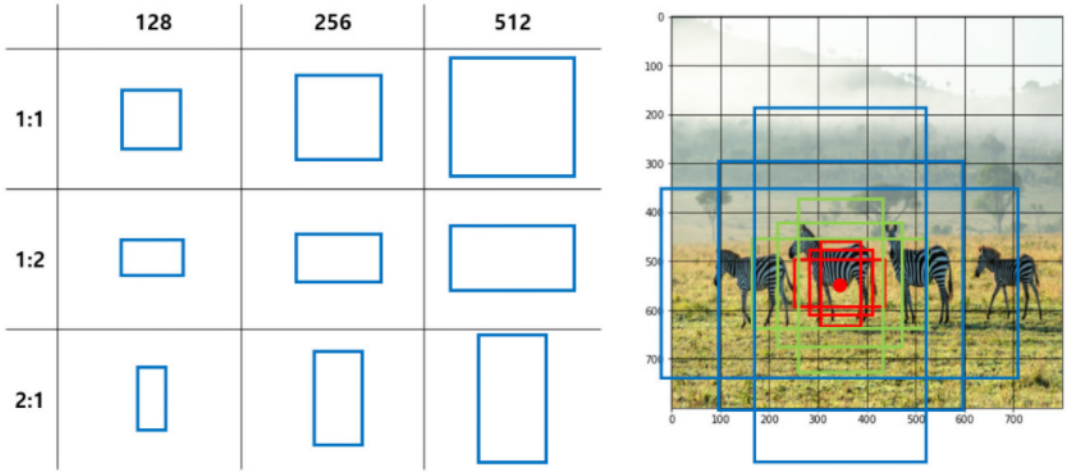
1. What's improved ??

Anchors

↳ ROI 가 될 수 있는 영역의 규격을 사전 정의 해 놓은 것.

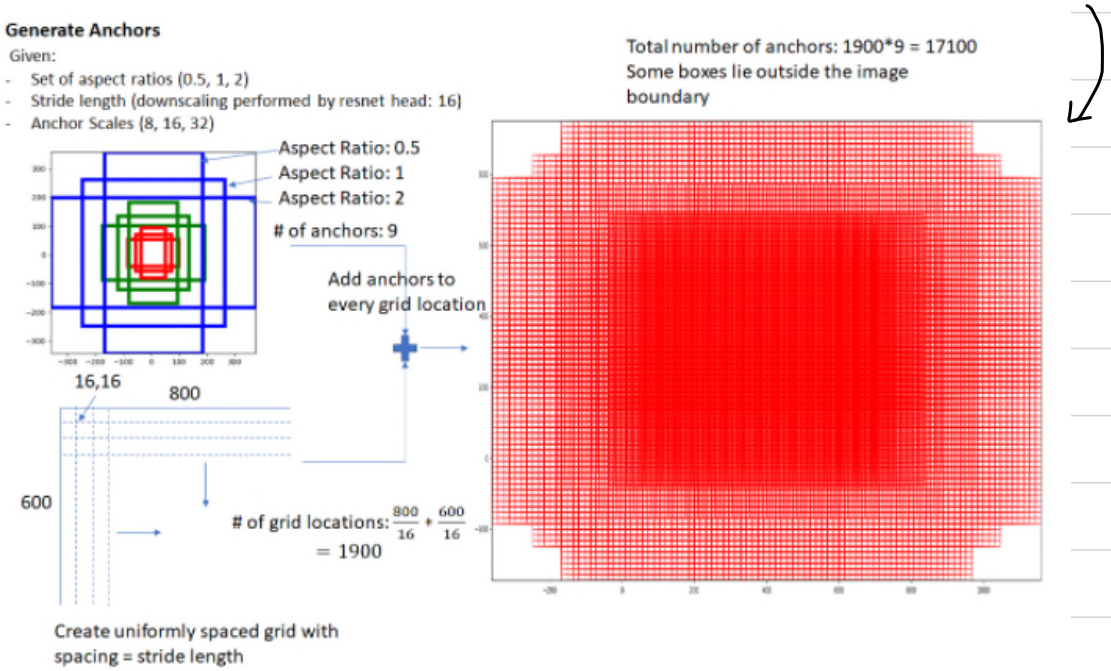
논문 예시

- Scale [128 , 256 , 512]
- Ratio [1:1 , 2:1 , 1:2]

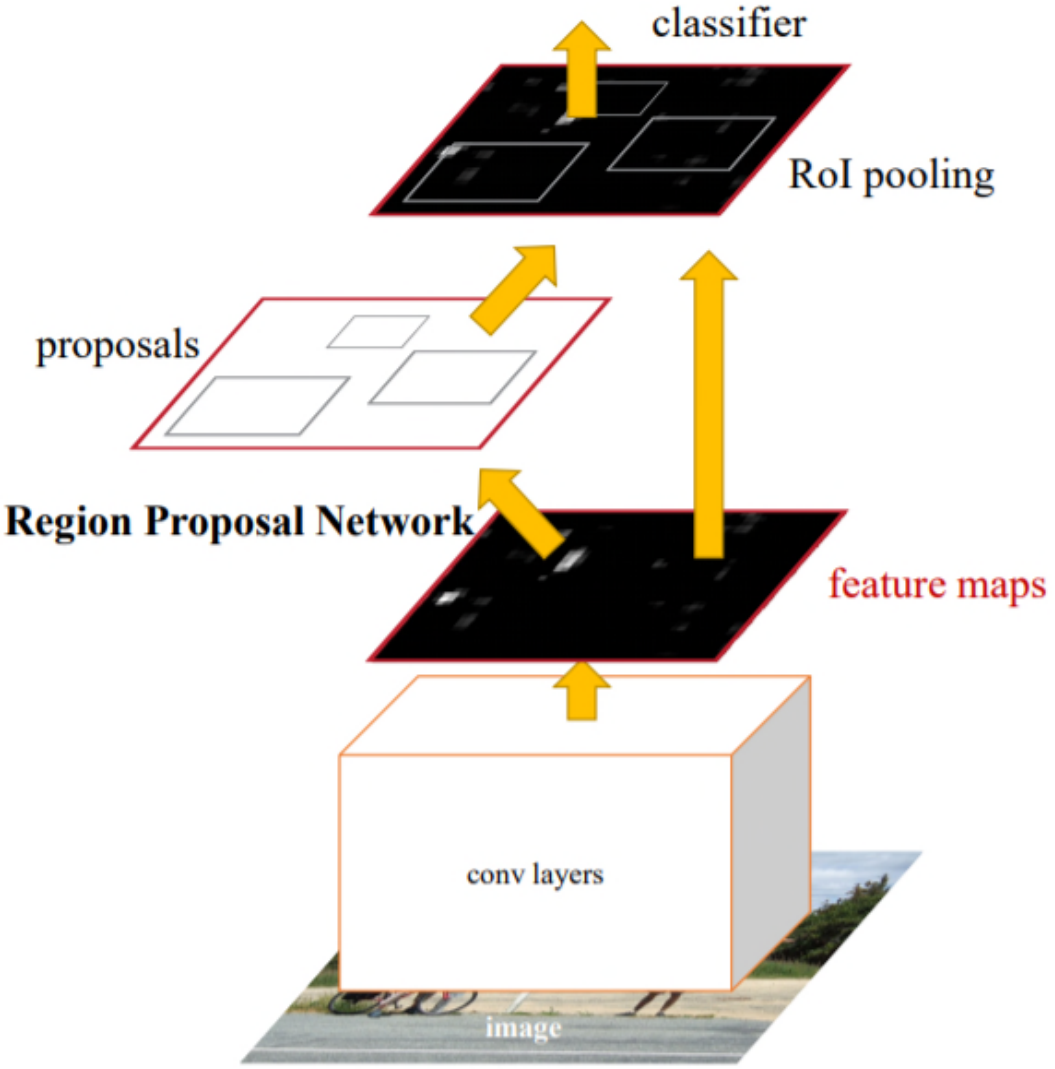


Anchor boxes

수 만은 Anchor Box 중 RPN으로 부터 높은 score를 얻은 것만 Detector로 전달.

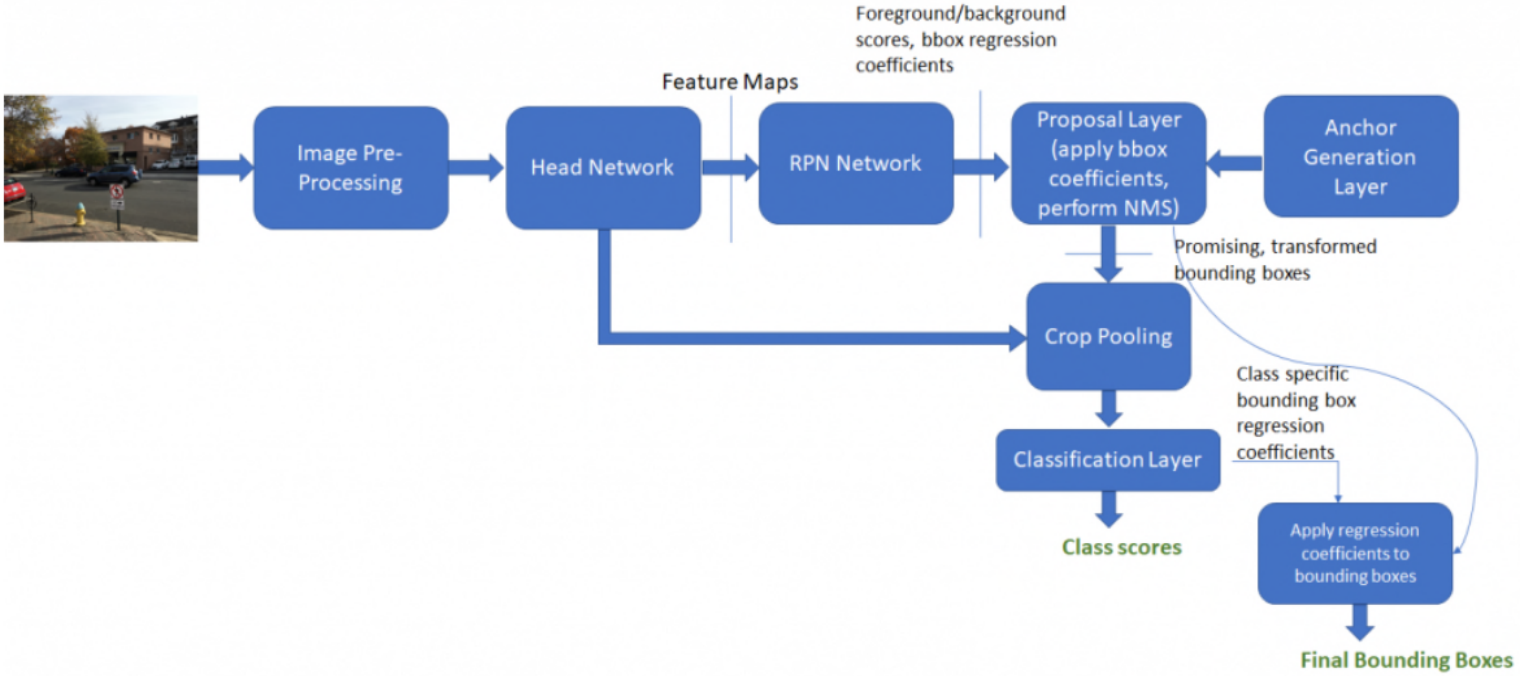


2. Architecture + Forward pass



↳ 그림에서는 간단한 네트워크가 하나 추가된 정도로 보이지만 ....

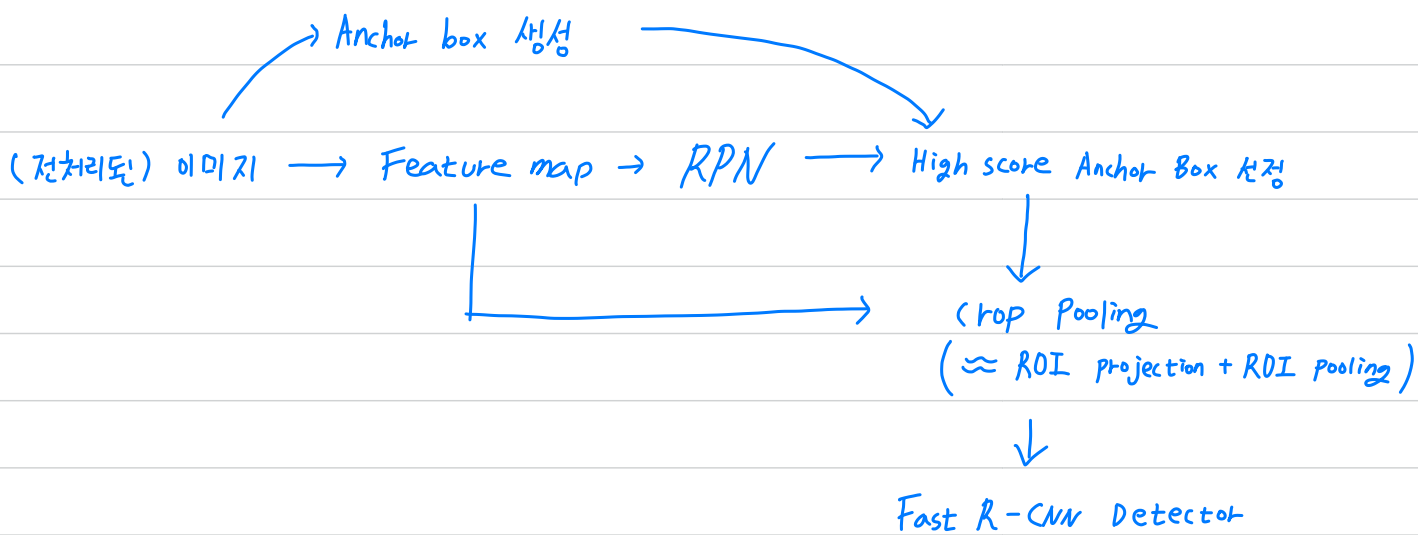
The steps carried out during inference are shown below



- 비슷한 이름의 layer가 여럿 존재. (혼동 주의)

Anchor Generation Layer
Anchor target Layer
Proposal Layer
Proposal target Layer

- 추론 흐름



3. All about RPN.

regular grid. The RPN is thus a kind of fully convolutional network (FCN) [7] and can be trained end-to-end specifically for the task for generating detection proposals.

What is FCN? → Segmentation 을 알아야 함.

Segmentation ? (Semantic segmentation , instance segmentation)

↳ pixel-level classification !!

입력

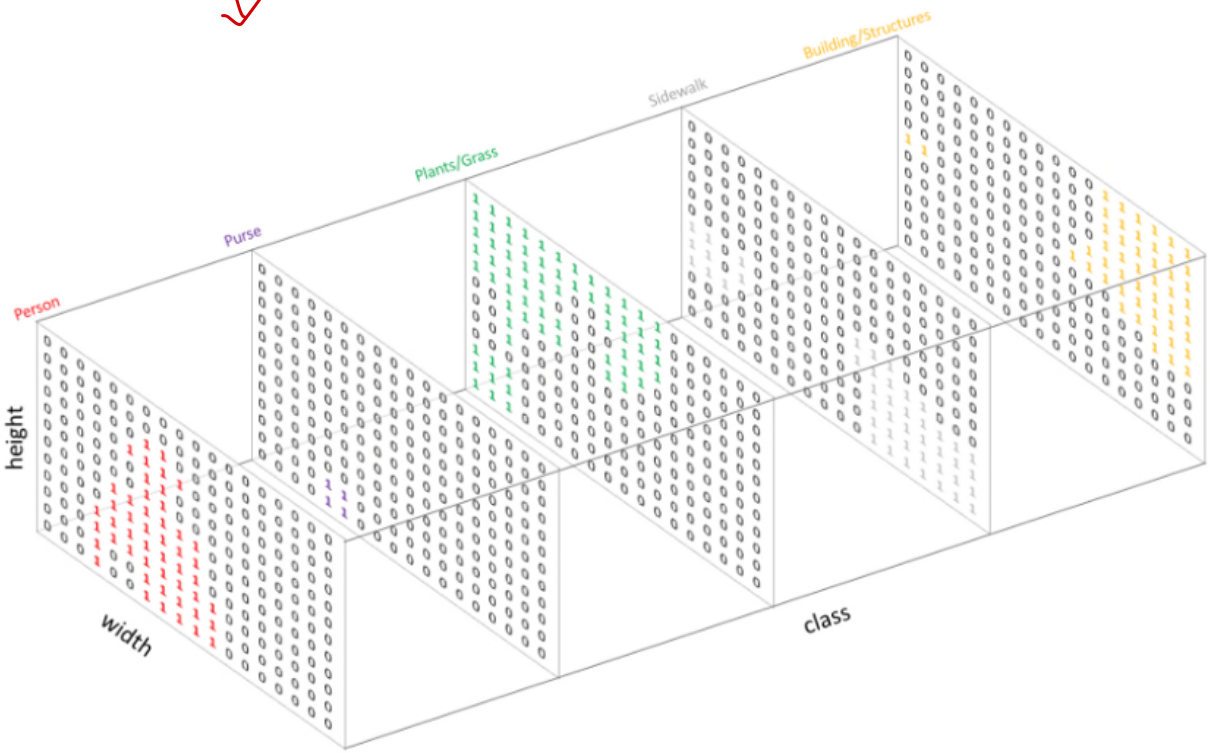
segmented

1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

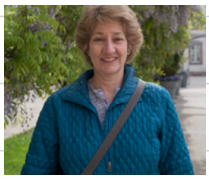
출력

Segmentation map

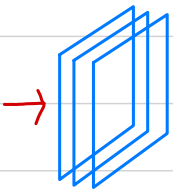
(정확히는)



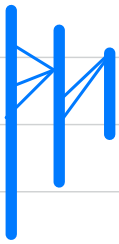
기존 image classification 에서는



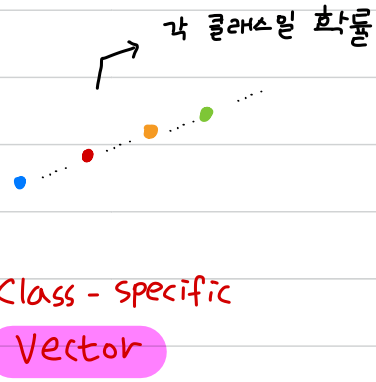
'분류'의 목적만 있는 사진. 위치 관심 X



feature extractor



FC

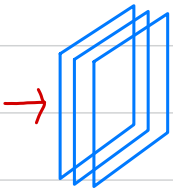


Class - specific Vector

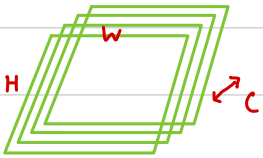
이걸 픽셀단위 분류 작업에 적용할 수 없다.
일단 입력 부터 위치정보가 끝까지 살아 있어야 하는 전체 이미지이며,
출력은 텐서 (혹은 배열) 형태 여야 한다.

일맥상 통하면서 다른 형태의 Fully - Convolutional Network 를 설계하게 된다.

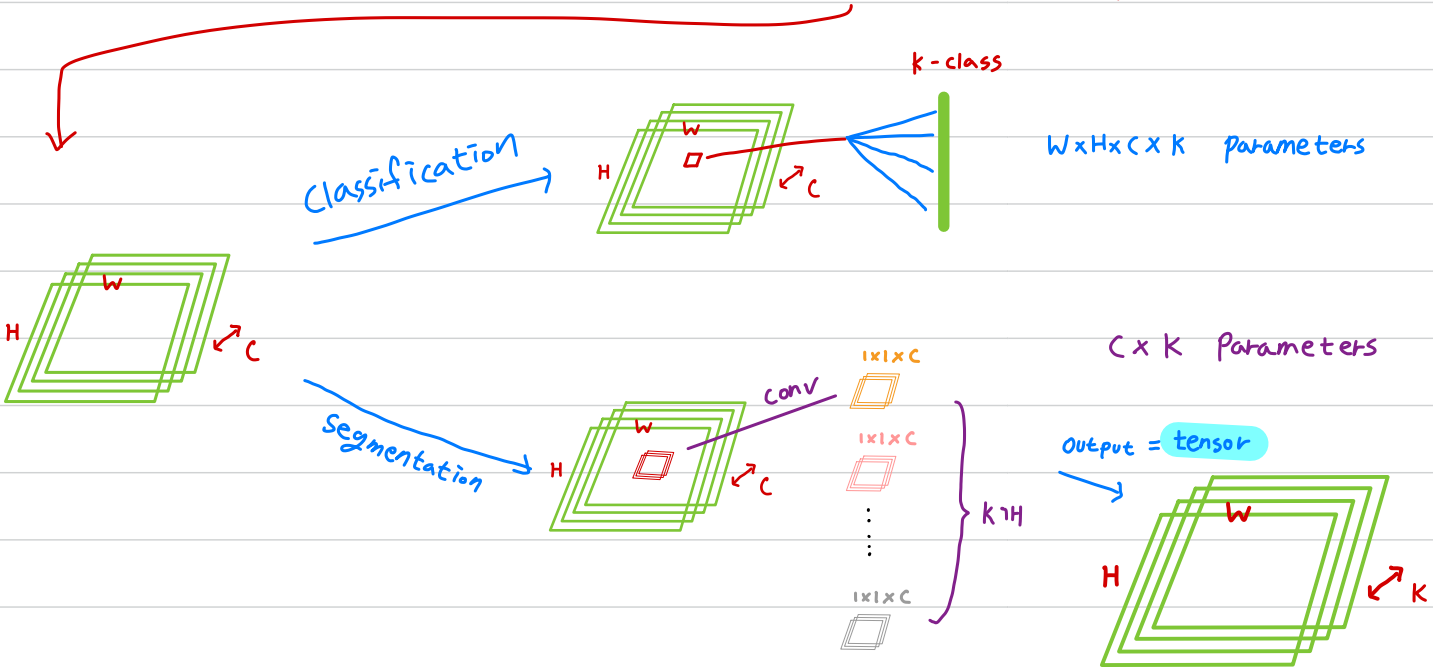
전체 이미지.



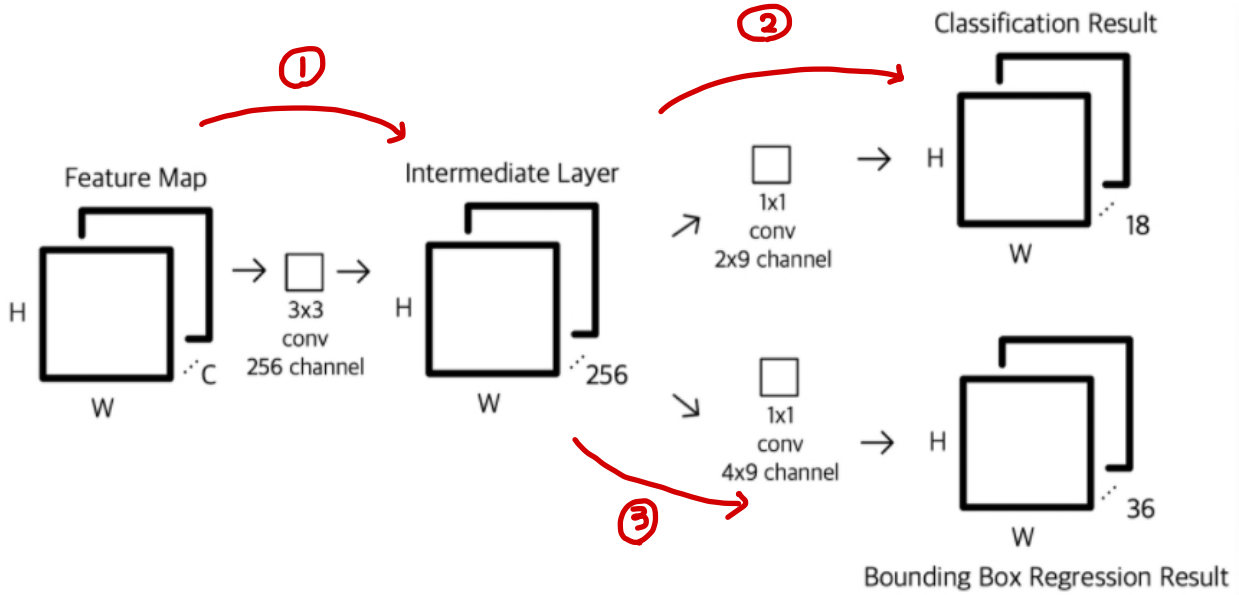
feature extractor



추출된 feature map
(W x H x C 라고 하자)



RPN = kind of FCN ?

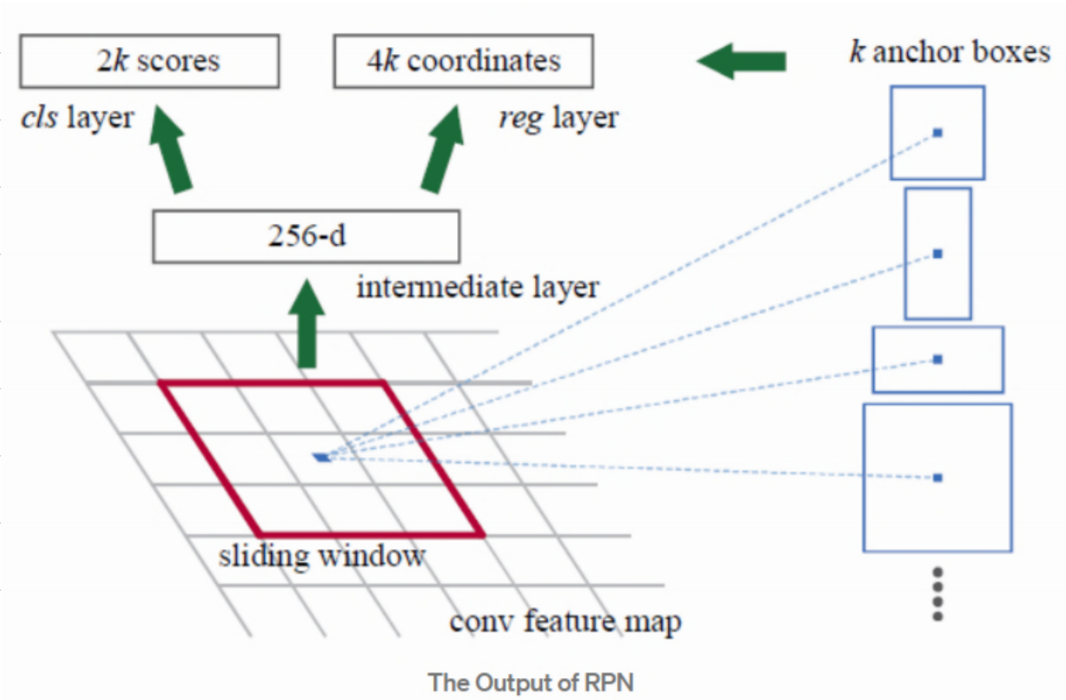


1. Normal conv. operation
   less dimension, higher-level features

2. 각 픽셀을 중심으로 생성된 Anchor box (논문에서 9개) 영역이 객체일/객체가 아닐 확률(2가지 scalar)을 출력하는 FCN

3. 각 픽셀을 중심으로 생성된 Anchor box (논문에서 9개) 영역이 Ground-truth box 가 될 수 있도록 변환에 쓰일 회귀계수 (t_x, t_y, t_w, t_h 4개)

다른버전의 설명 그림.



The Output of RPN

Properties of RPN.

Translation-invariant Anchors

↳ 이상적인 regional-proposal 이라면 이미지에 translation (이동) 이 적용되었을때 이와 상관 없이 같은 proposal 이 이루어져야한다.
(이미지의 특정 위치에서는 객체인지 아닌지 판별이 우수하나 다른 영역에서는 잘 되지 않는 다거나 하는 경우)

↳ Faster R-CNN 에서 제안하는 RPN 의 경우 FCN 기반의 네트워크로써 이러한 translation-invariant 성질이 보장된다.
↖ 정확히 이해 못함. 위치 정보가 끝까지 살아서 그런건가 추정 중

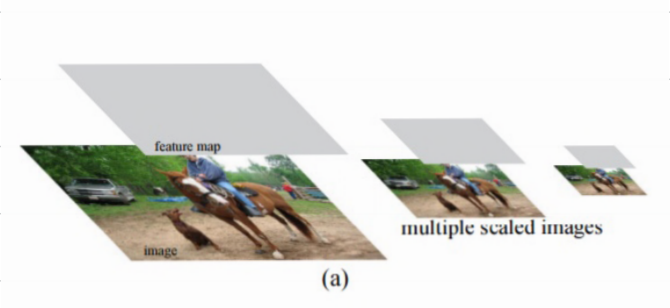
↳ 반대로 말하면 Faster R-CNN 이전에도 R.P. 를 수행하는 네트워크 설계에 대한 시도가 있었으며 translation-invariant 성질이 보장되지 않았음.
(Inception 팀에서 Multibox 라는 이름으로 시도하였다고 함)

↳ F.Y.I translation-variant 해야하는 task 도 존재.
= instance segmentation : 같은 범주라 할지라도 위치가 다르면 두 객체는 개별 인스턴스 취급

Benefits of predefined Anchor size

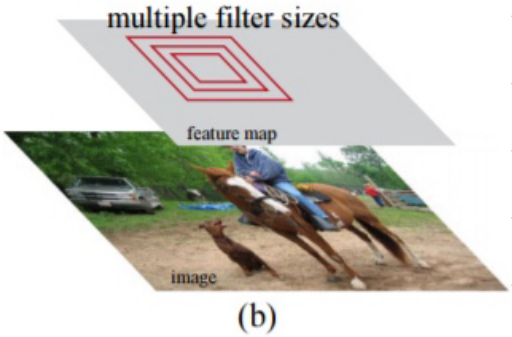
다양한 크기와 비율의 객체에 대해서 학습이 가능하도록 하는 여러 시도들이 있었다.

1. Pyramids of images



R-CNN, Fast R-CNN 등에서 시도.
이미지 자체를 변형하여 S.S 가 이루어지도록 하여 각 스케일에 따라 모델들이 따로 학습.
↳ 나쁜 방법은 아니나 시간 소모가 상당.

2. Pyramids of filters

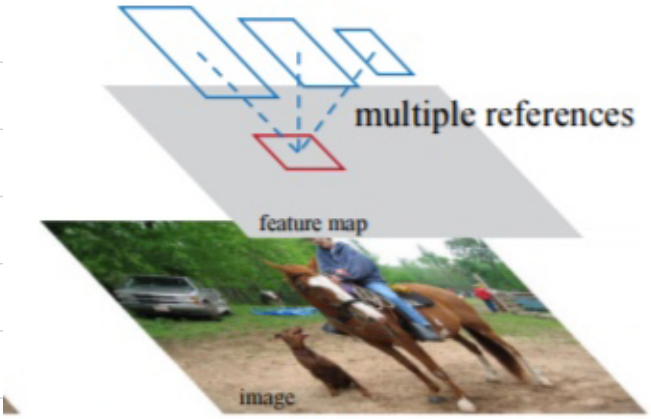


(b)

Inception 팀이 시도한 Multibox 방식.
이미지는 그대로 사용하여, 추출된 feature map 으로부터
다양한 (아마도 사전 정의되지 않은?) 필터 사이즈로 R.P 를
진행하는 방식

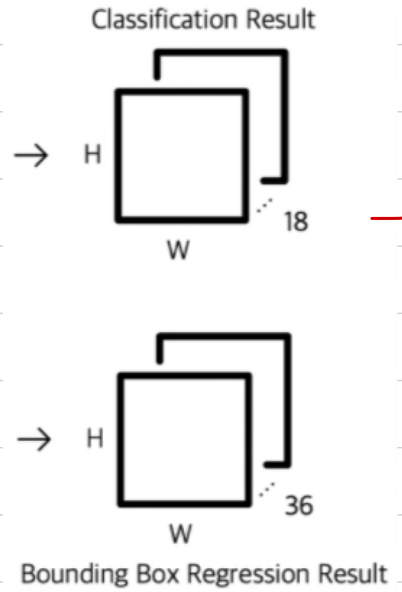
↳ 마찬가지로

★ 3. Pyramids of Anchors



(c)

Faster R-CNN 에서 추구한
방식으로 논문의 말을 번역하면
"Multibox 방식과 달리 각
cls layer, reg layer 는 하나의 스케일/비율 을
담당하기 때문에 더 미세한 차이에 대해 개별 학습할
수 있고, 개별 모델을 생성하는 것이 아니기 때문에
cost-free 에 가깝다."



RPN 의 출력에서 한 채널이 각각
특정 scale / ratio 의 Anchor box 에 대한
정보를 담당하므로 개별 학습이
가능함을 얘기하는 것 같다.

Loss function

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

Fast R-CNN 의 multi-task loss 와 거의 흡사.

L_{cls} : class 별 log loss (binary crossentropy)

L_{reg} : smooth L_1 regressing parameterized 4 coord.

↳ 다른점 파라미터화(?)가 한 단계 거쳐서 진행

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a,$$
$$t_w = \log(w/w_a), \quad t_h = \log(h/h_a),$$
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a,$$
$$t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a),$$

x : 예측

x_a : anchor box 꺼

x^* : Ground-truth 꺼

Positive / Negative label of RPN and Batch

Label

생성 직후 전체 Anchor box 의 수는 아주 많다.
(feature map 이 50x50xC 형태일 때 9 종류의 Anchor box 로 생성하면 50x50x9 = 22500 개)

이 박스 모두를 학습용으로 활용하고자 한다면 시간적, 성능적 손실이 크다.

따라서 일정 기준을 만족하는 박스만 P/N 라벨링을 하여 학습에 활용한다.

Positive : (아무) Ground truth 와 0.7 이상의 IOU 를 보이는 박스
Negative : (그 어떤) Ground truth 와도 0.3 이상의 IOU 를 보이지 않는 박스
NULL : 0.3 < IOU < 0.7 의 박스로 학습에 쓰지 않음.

Batch

Fast R-CNN 처럼 image-centric 방식으로 한 이미지의 여러 ROI 로 배치를 구성하며 Negative sample 의 dominant 한 성격을 보완하기 위해 P/N 의 비율을 1:1 로 하여 256 anchor box 를 한 배치로 설정한다.

↳ 128 개의 Positive sample 이 확보되지 않는 이미지의 경우 Negative sample 을 보충하기도 함.