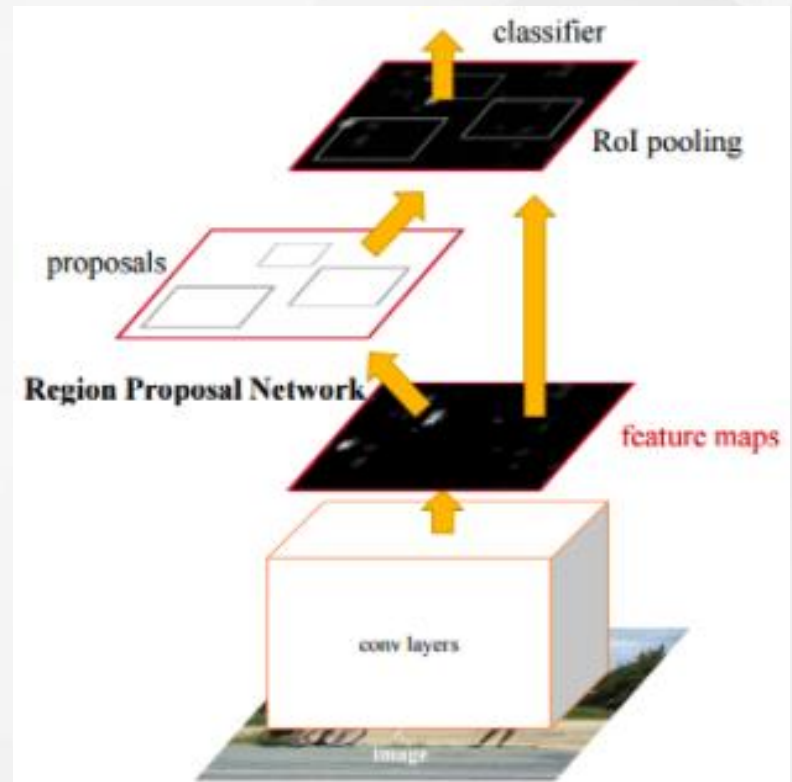


Chapter 02. 이미지 처리 분야 딥러닝 모델 (Object Detection)

Fast, Faster RCNN



RCNN

R-CNN: *Regions with CNN features*

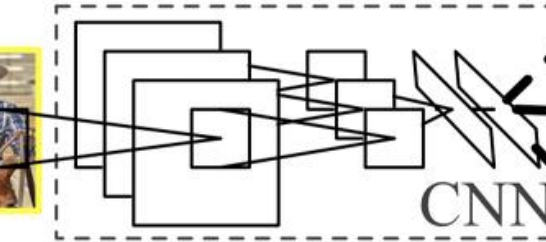


1. Input image



2. Extract region proposals (~2k)

warped region



3. Compute CNN features

aeroplane? no.

⋮

person? yes.

⋮

tvmonitor? no.

4. Classify regions

RCNN

1. 오래걸린다

Selective Search에서 뽑아낸 2000개의 영역 이미지들에 대해서 모두 CNN모델에 classification

Training Time: 무려 **84시간**

Testing Time은 GPU K40 사용 기준으로 frame당 **13초**

CPU를 사용하였을 때 frame당 **53초**가 걸립니다.

2. 복잡하다

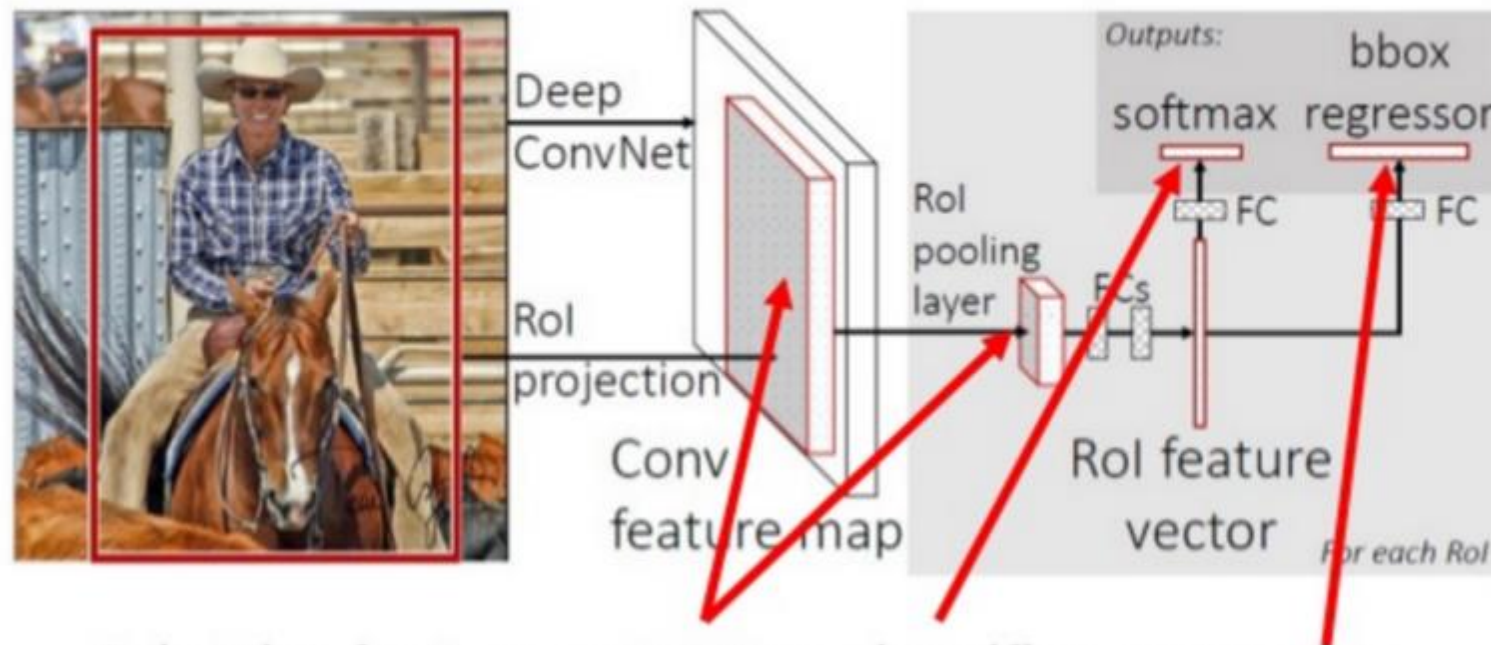
R-CNN은 Multi-Stage Training을 수행하며,
CNN, SVM, 그리고 Bounding Box Regression까지
총 세 가지의 모델을 필요로 하는 **복잡한 구조**

3. Back Propagation이 안된다.

SVM, Bounding Box Regression에서 학습한 결과가 CNN을 **업데이트 시키지 못합니다.**

Fast RCNN

Fast R-CNN: Joint Training Framework



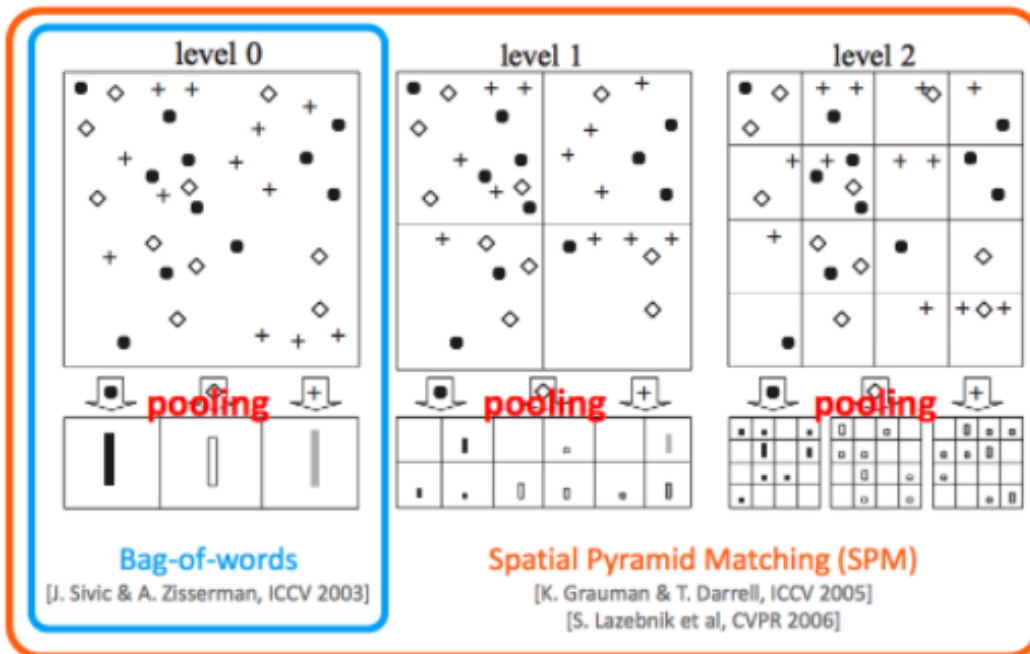
Joint the feature extractor, classifier, regressor together in a unified framework

Unified Framework

1. Feature Extractor
2. Classifier
3. Regressor

Spatial Pyramid Pooling

Warping 에서 일어나는
정보손실을 막으며,
이미지의 차원을 맞추기 위한 방법
일정 개수의 지역으로 나눈 뒤, 각 지역에 BoW
를 적용



<https://blog.lunit.io/2017/06/01/r-cnns-tutorial/>

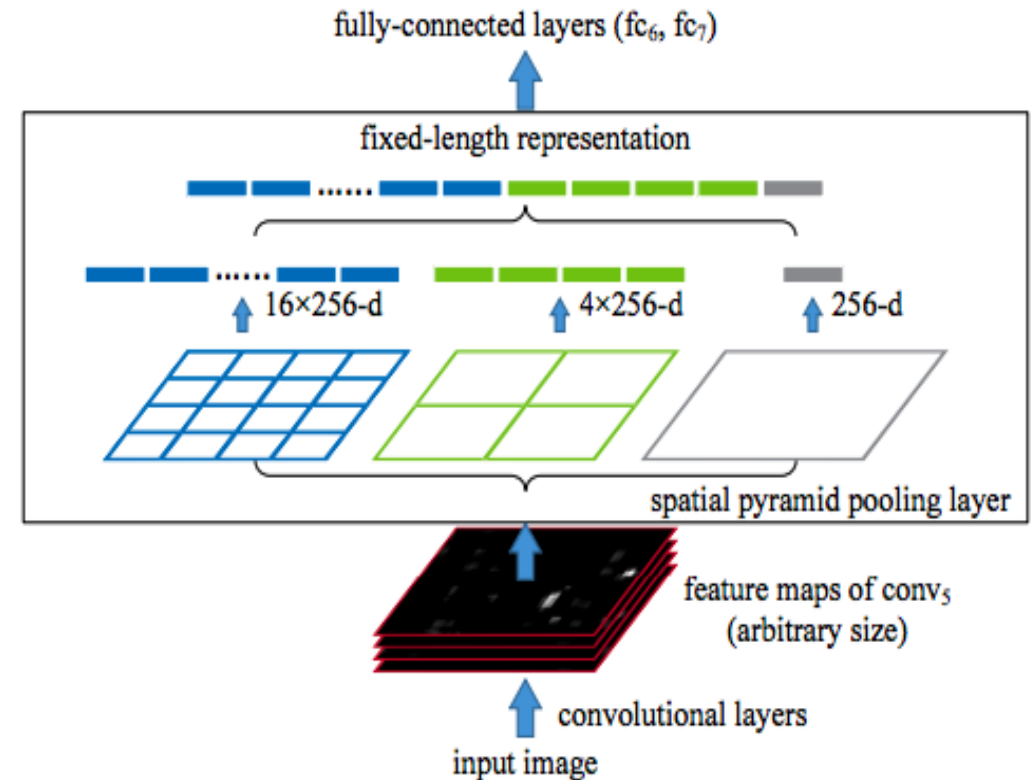
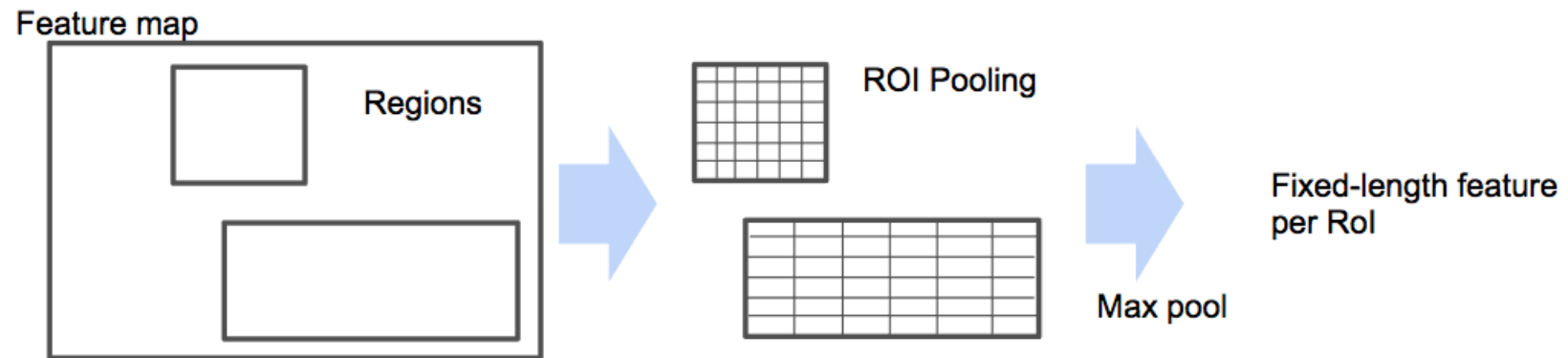


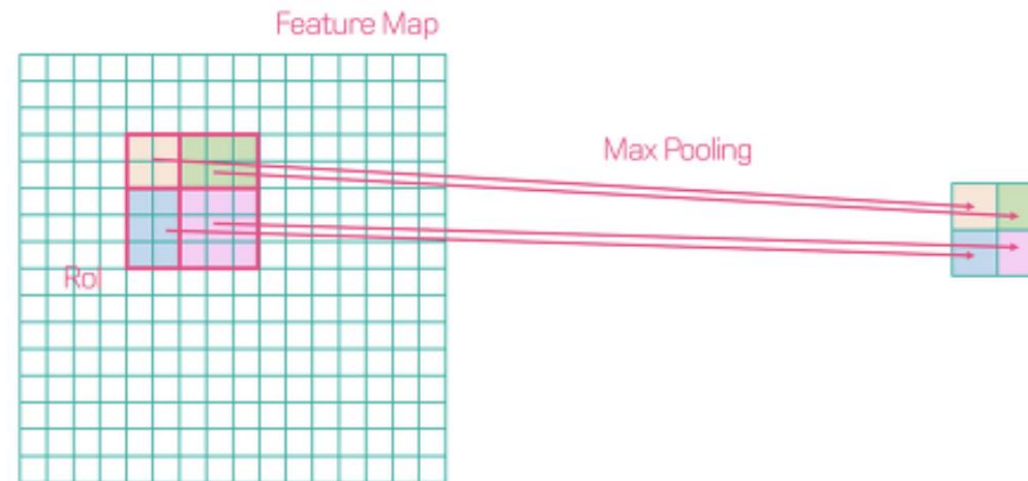
Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the conv₅ layer, and conv₅ is the last convolutional layer.

Region of Interest Pooling

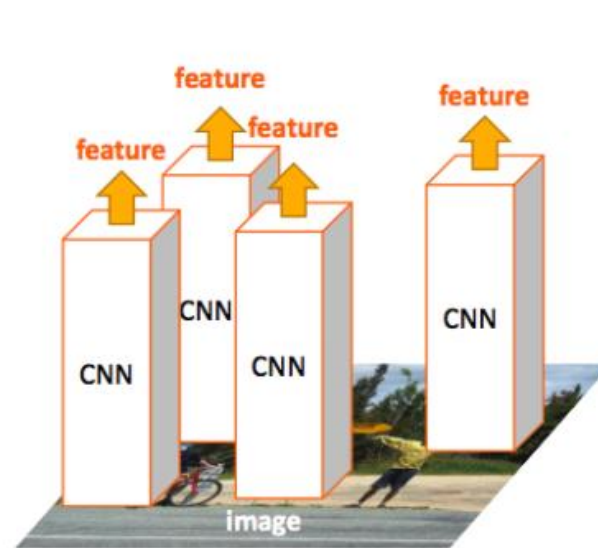
<https://blog.lunit.io/2017/06/01/r-cnns-tutorial/>



RoI Pooling



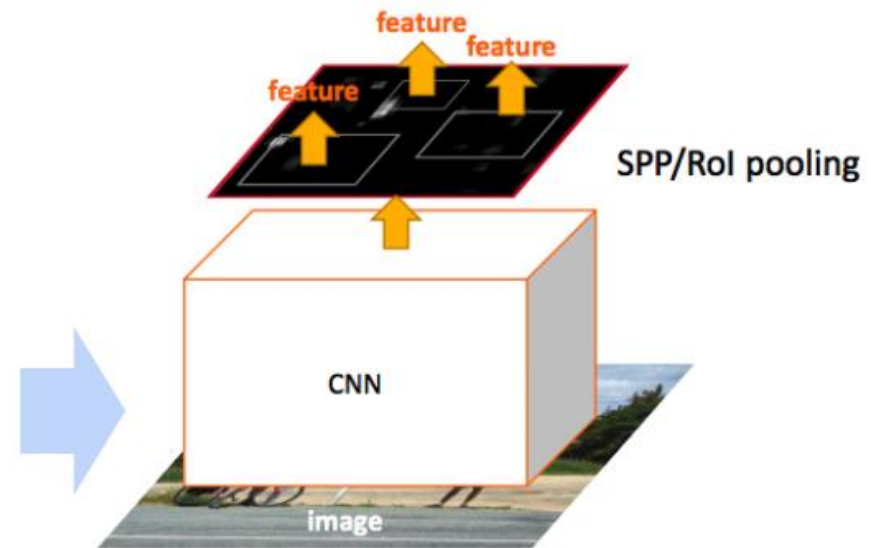
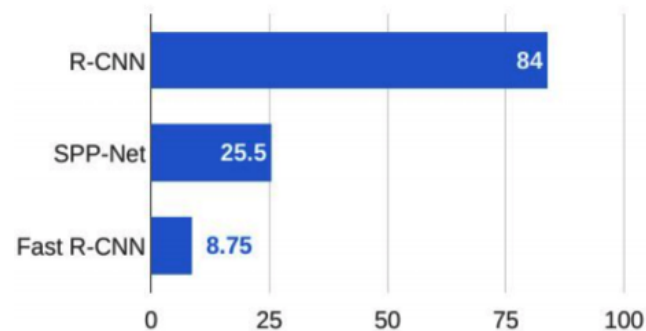
각 bin에 대해 max pooling
또는 average pooling을 취함



R-CNN

- Extract image regions
- 1 CNN per region (2000 CNNs)
- Classify region-based features
- Complexity: $\sim 224 \times 224 \times 2000$

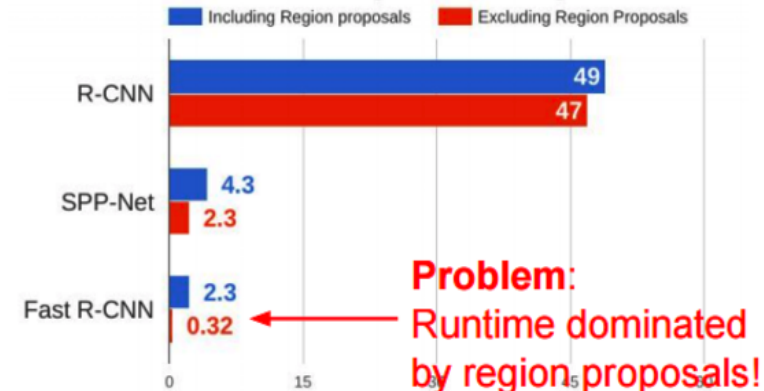
Training time (Hours)



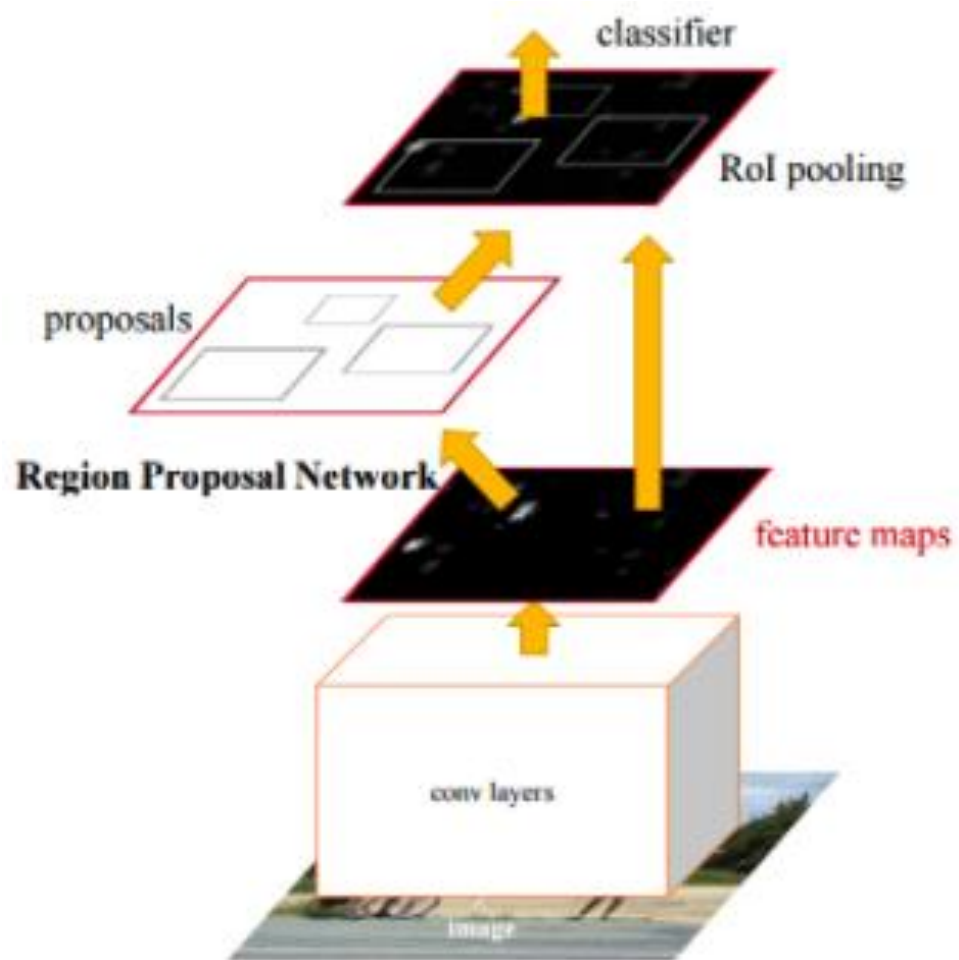
SPP-net & Fast R-CNN (the same forward pipeline)

- 1 CNN on the entire image
- Extract features from **feature map regions**
- Classify region-based features
- Complexity: $\sim 600 \times 1000 \times 1$
- **$\sim 160\times$ faster than R-CNN**

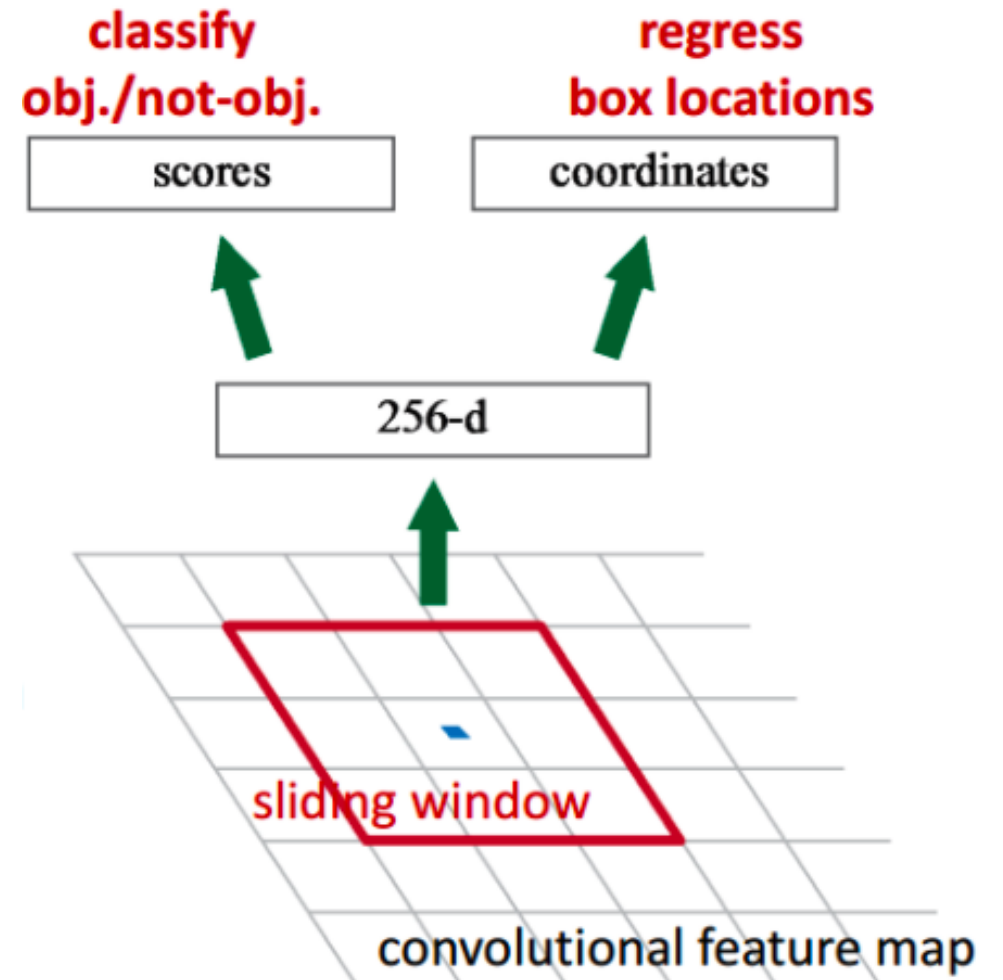
Test time (seconds)



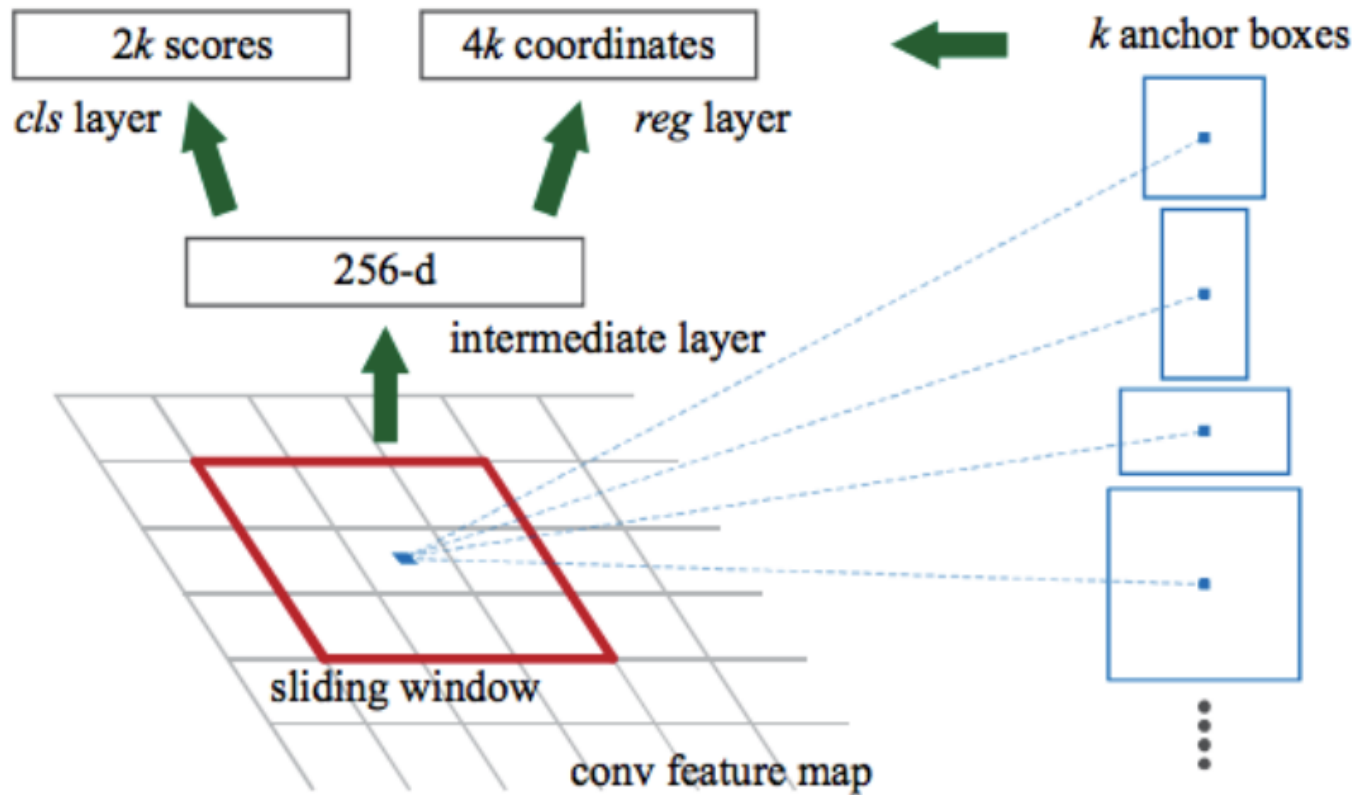
Faster RCNN



region proposal network(RPN)를 학습해보자



Faster RCNN



RPN에 사용된 anchor

논문에서는 3가지의 크기
(128, 256, 512)와 3가지의 비
율(2:1, 1:1, 1:2)을 사용

Faster RCNN

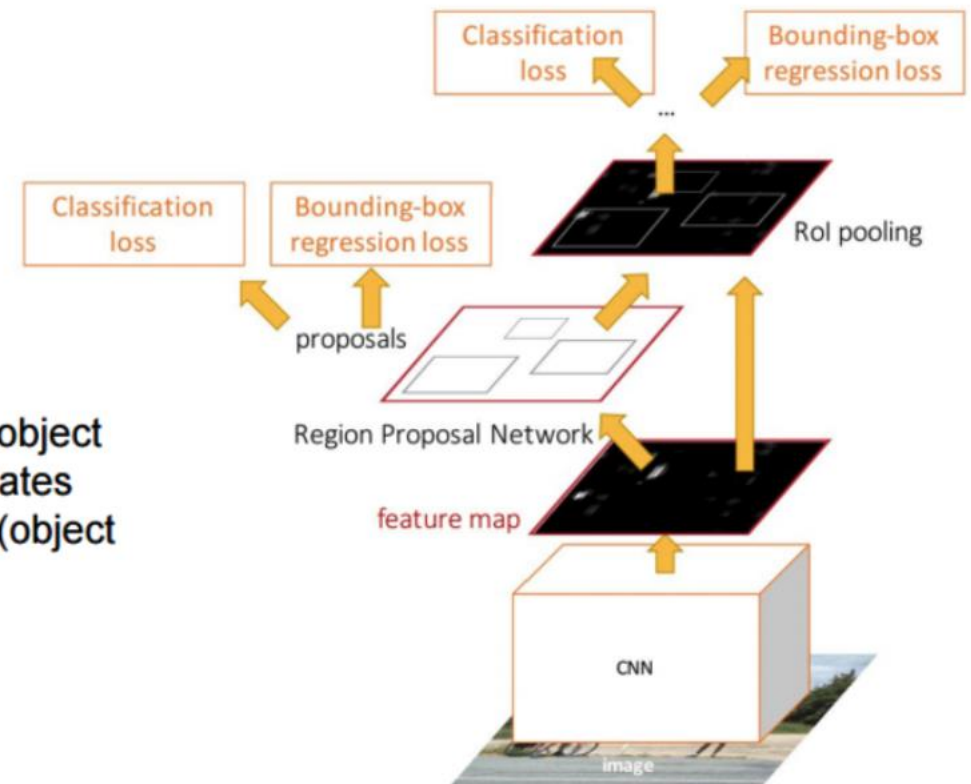
per image

system	time	07 data	07+12 data
R-CNN	~50s	66.0	-
Fast R-CNN	~2s	66.9	70.0
Faster R-CNN	198ms	69.9	73.2

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

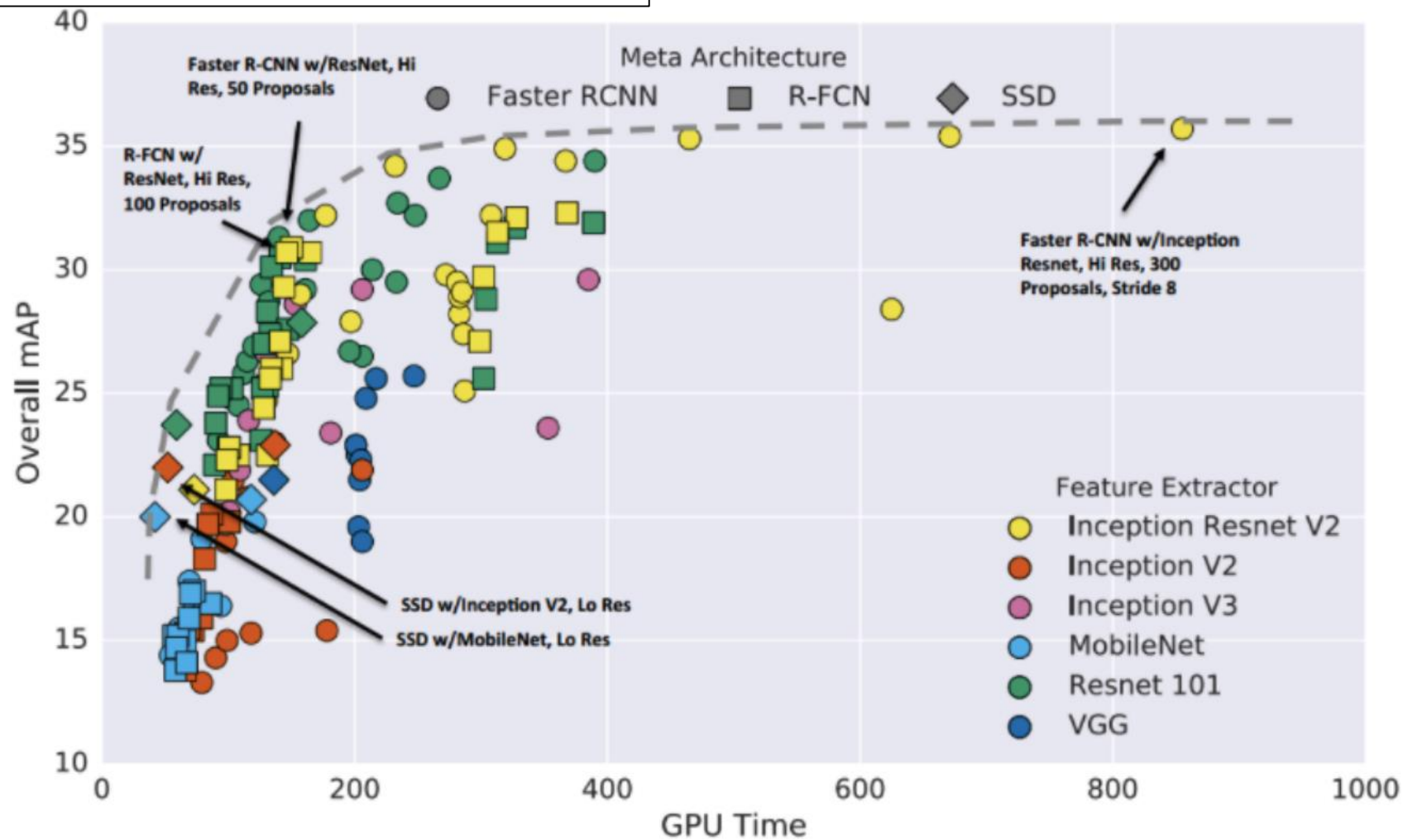
Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



Faster R-CNN의 Approximate joint optimization 학습 방법

Faster RCNN



CNN 기반 object detector 성능 비교 [10]

Thank You