5 minute summary:
R-CNN, Fast R-CNN, Faster R-CNN

1. R-CNN

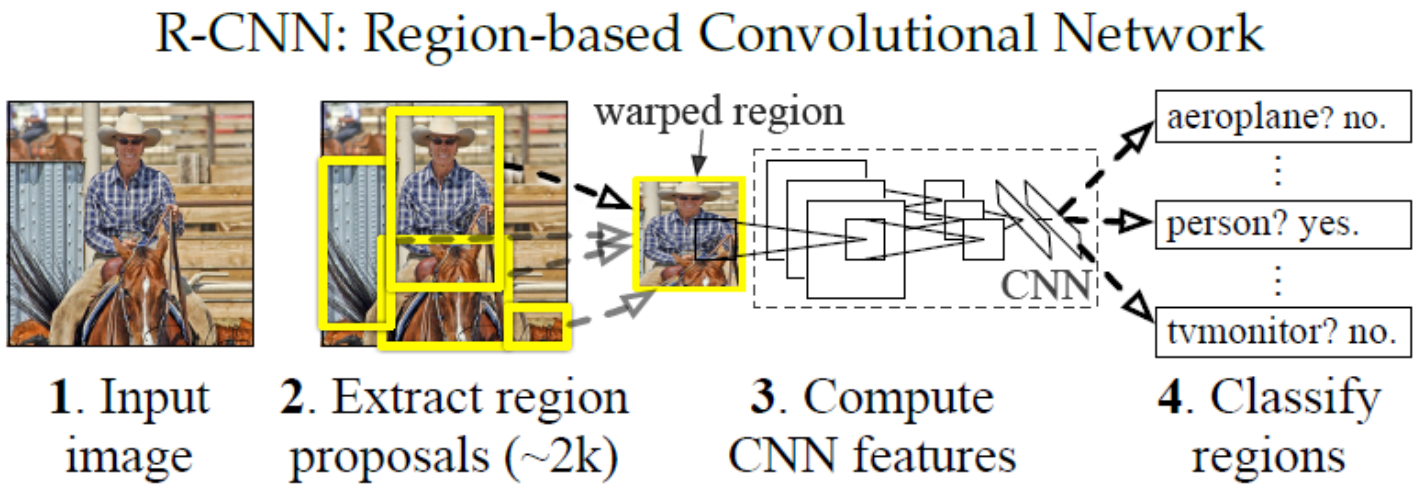


Fig. 1. Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional network (CNN), and then (4) classifies each region using class-specific linear SVMs. We trained an R-CNN that achieves a mean average precision (mAP) of 62.9% on PASCAL VOC 2010. For comparison, [21] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%. On the 200-class ILSVRC2013 detection dataset, we trained an R-CNN with a mAP of 31.4%, a large improvement over OverFeat [19], which had the previous best result at 24.3% mAP.

- 분리된 네트워크
(Region Proposal Network, CNN, Classifier)
- Backpropagation 불가
- Selective Search, SVM
- Selective Search 시간이 너무 오래 걸림.
- CNN의 마지막 레이어 사이즈에 맞추기 위해서 Bounding Box의 크기를 warp 하였다.

2. Fast R-CNN

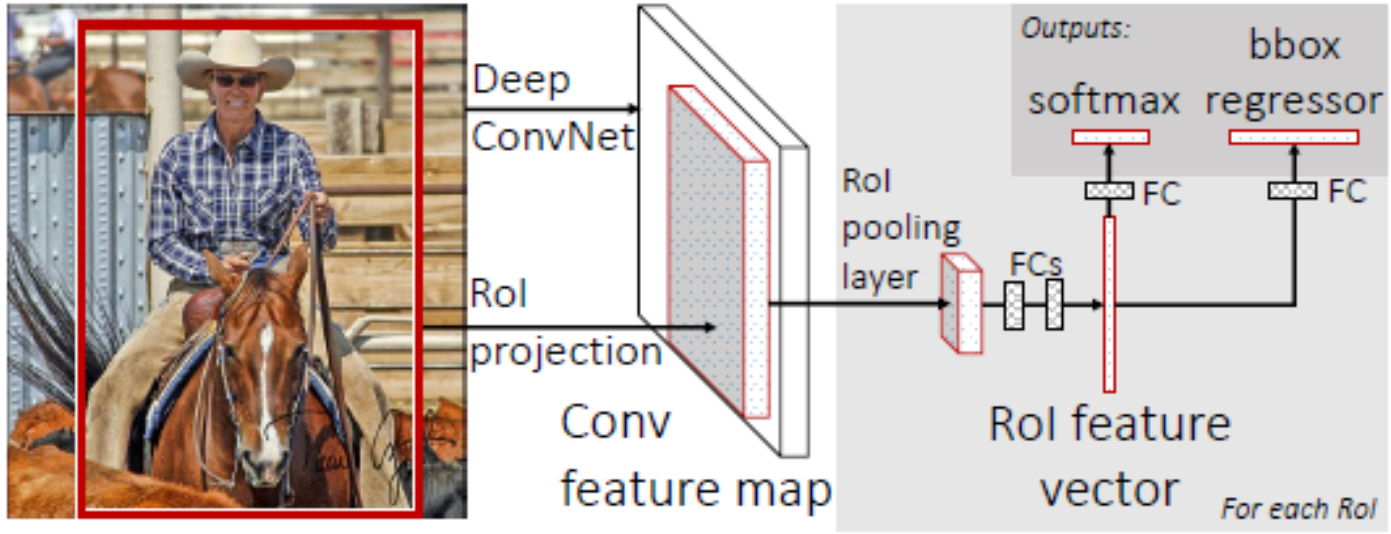


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

- 일부 분리된 네트워크

Region Proposal Network & CNN + Classifier + Regressor

- RoI Pooling을 통해 최종 사이즈를 조절해서 warping 문제를 해결.
- Softmax와 box regressor를 병렬적으로 구성하여서 backpropagation 문제 해결
- Selective Search에 걸리는 시간이 2.3초 중에 2초로 상당한 bottleneck.

3. Faster R-CNN

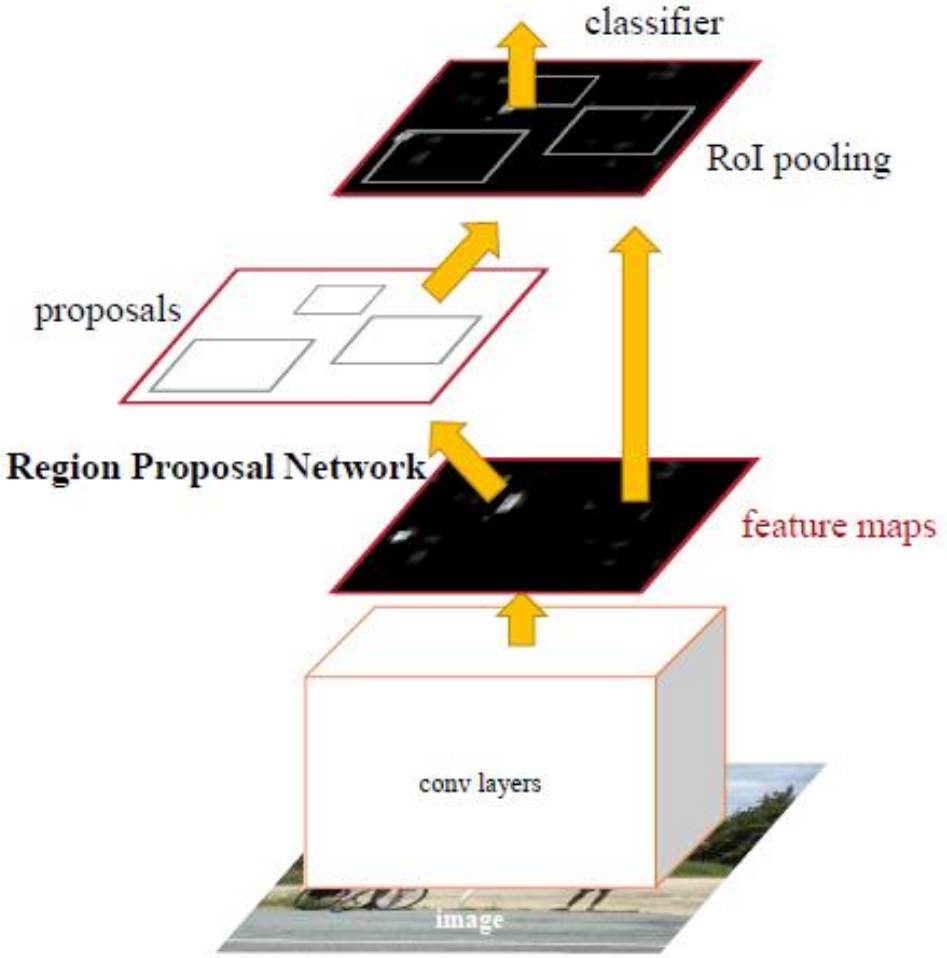
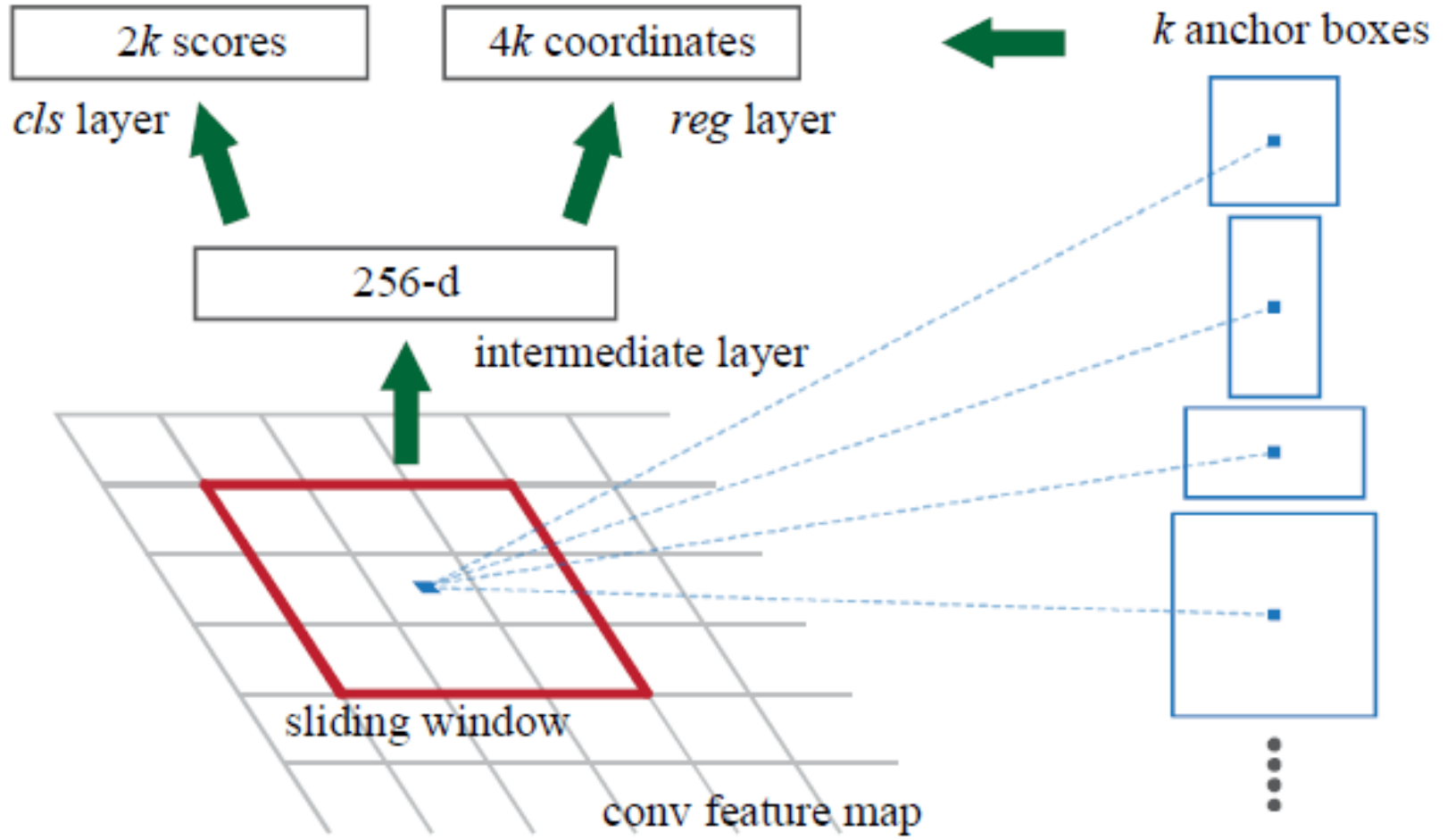


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

- 통합된 네트워크
- Anchors를 도입
- Anchor는 Sliding window의 center
- K개의 anchor boxes를 정해 놓음.
- 최상위 convolutional feature map에서 3 x 3 filter로 256 depth의 feature map을 만들고 이를 1 x 1 convolution을 시행하고 classify(obj./not-obj.), regression(box location) 실시.



최상위 convolutional feature map에서
3 x 3 filter로 256 depth의 feature map을 만든 후, 1 x 1 convolution을 시행,
classify(obj./not-obj.), regression(box location) 실시.