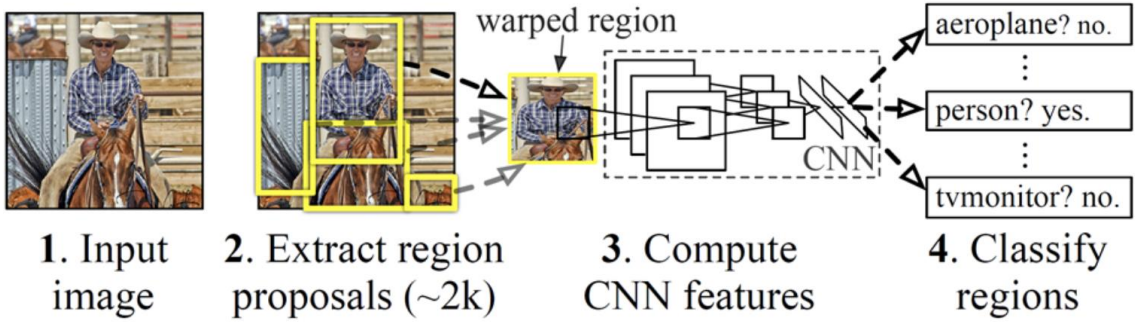
Fast R-CNN

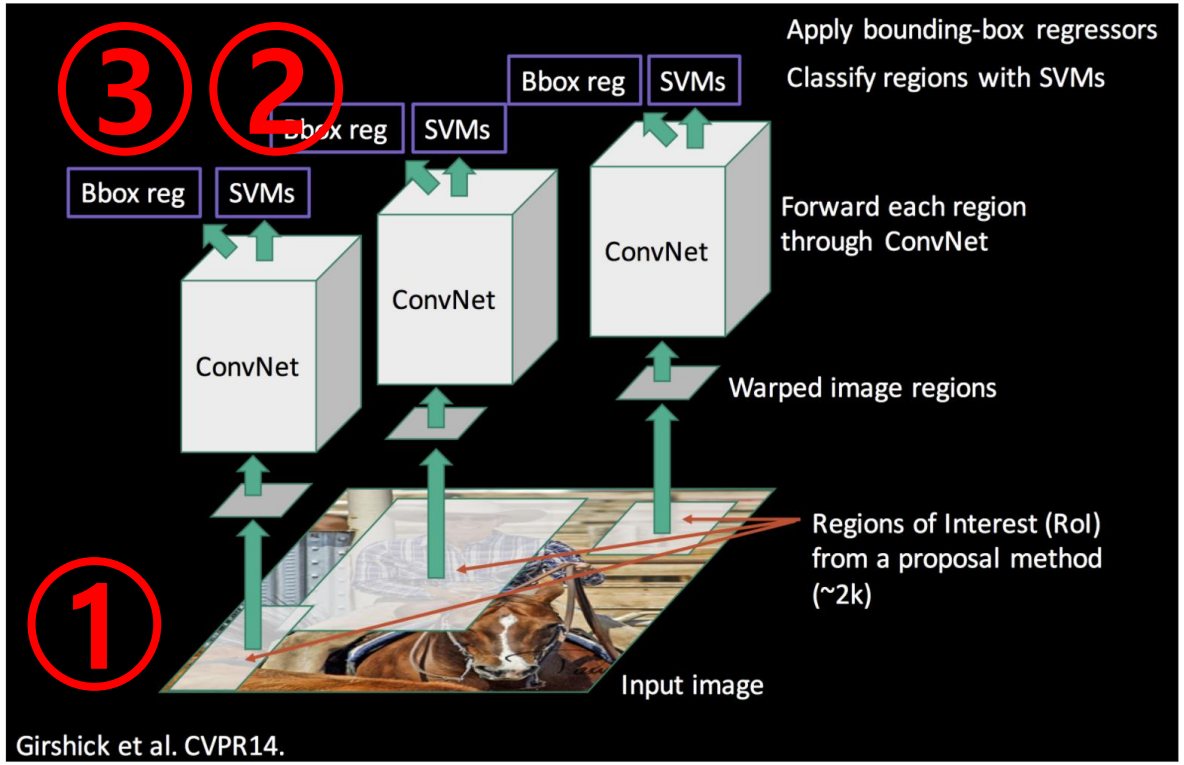
Ross Girshick

박태우

Introduction – R-CNN 알고리즘

1. Selective Search에 의해 region proposal 추출.
   (selective search는 링크 참고.)
2. Region proposal을 모두 CNN에 통과시킴.
   (ImageNet기반 pre-trained 모델)
3. CNN의 결과로 얻은 각각의 feature map을 SVM을 통해 classification, regressor를 통해 bounding box regresion 진행.



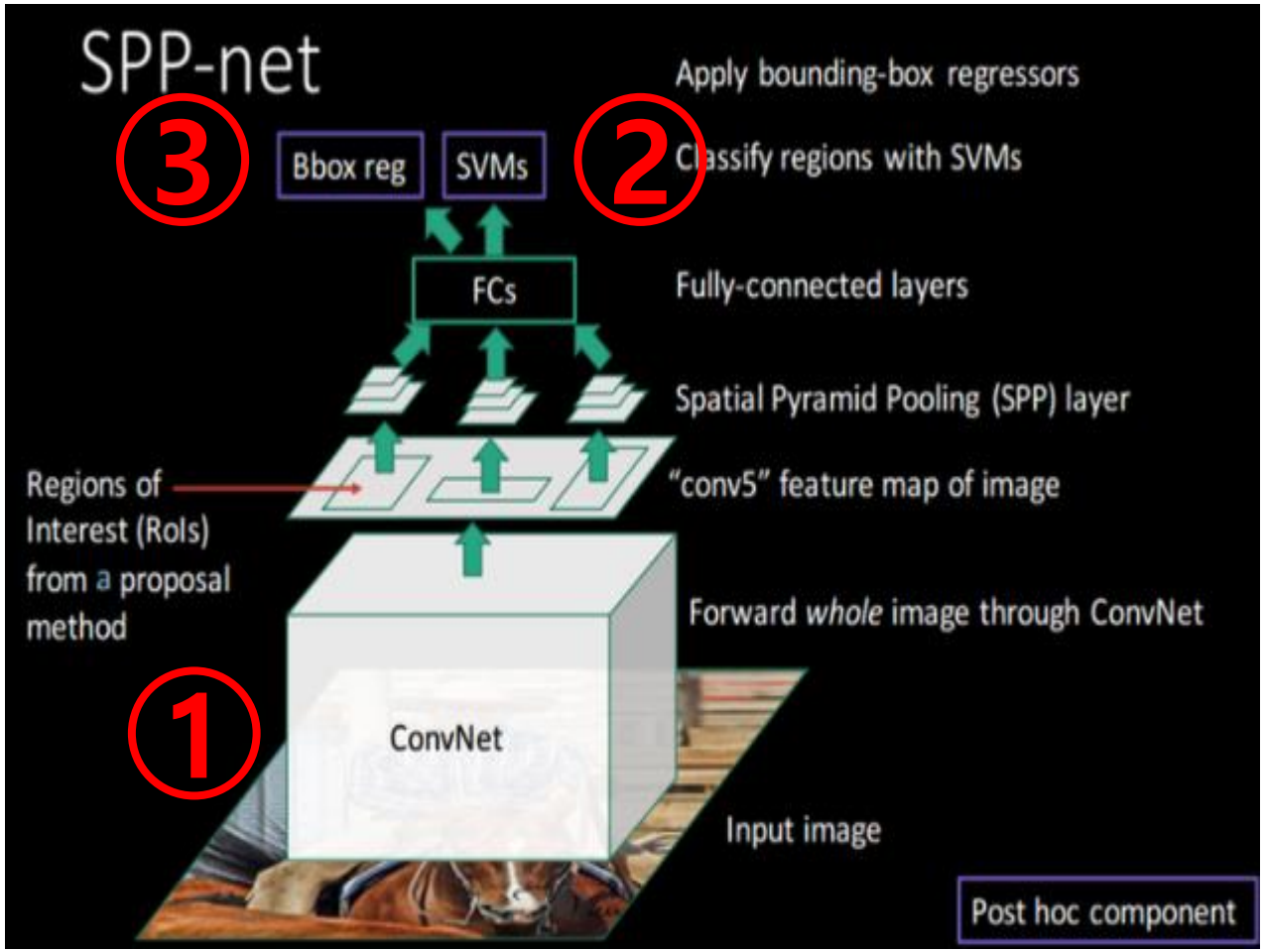


***단점**

- 모든 region proposal을 CNN에 통과시키고, 학습의 단계가 나눠져있기 때문에 속도가 느림.
- region proposal이 CNN에 통과되기 위해선 input 사이즈에 맞게 변형되어야함.

Introduction – SPPnet 알고리즘

1. 전체 이미지를 Covolutional Network에 통과시켜 feature map 추출.
2. Feature map에서 각 RoI 추출한 뒤 SPP layer와 FC에 통과.
3. FC의 결과를 SVM을 통해 classification, regressor를 통해 bounding box regresion 진행.



***단점**

- R-CNN에 비해 속도는 상당히 증가했지만 여전히 학습에 multi-stage pipeline을 적용시킴.
- R-CNN과 달리 SPP 앞단에 있는 ConvNet는 fine tuning을 하지 않기 때문에 정확도에 문제가 있음.

Architecture and Training

1. 전체 이미지를 ConvNet에 통과시켜 feature map 추출

2. Selective Search를 통해 얻은 RoI들을 feature map에 투영 및 RoI Pooling layer에 통과시켜 fixed feature vector를 얻음.

3. Feature vector는 2개의 FC에 각각 통과

4. FC1의 경우 softmax를 적용해 classification 수행, FC2의 경우 regressor로써 class 1개당 4개의 bounding box value를 가짐.

Architecture and Training

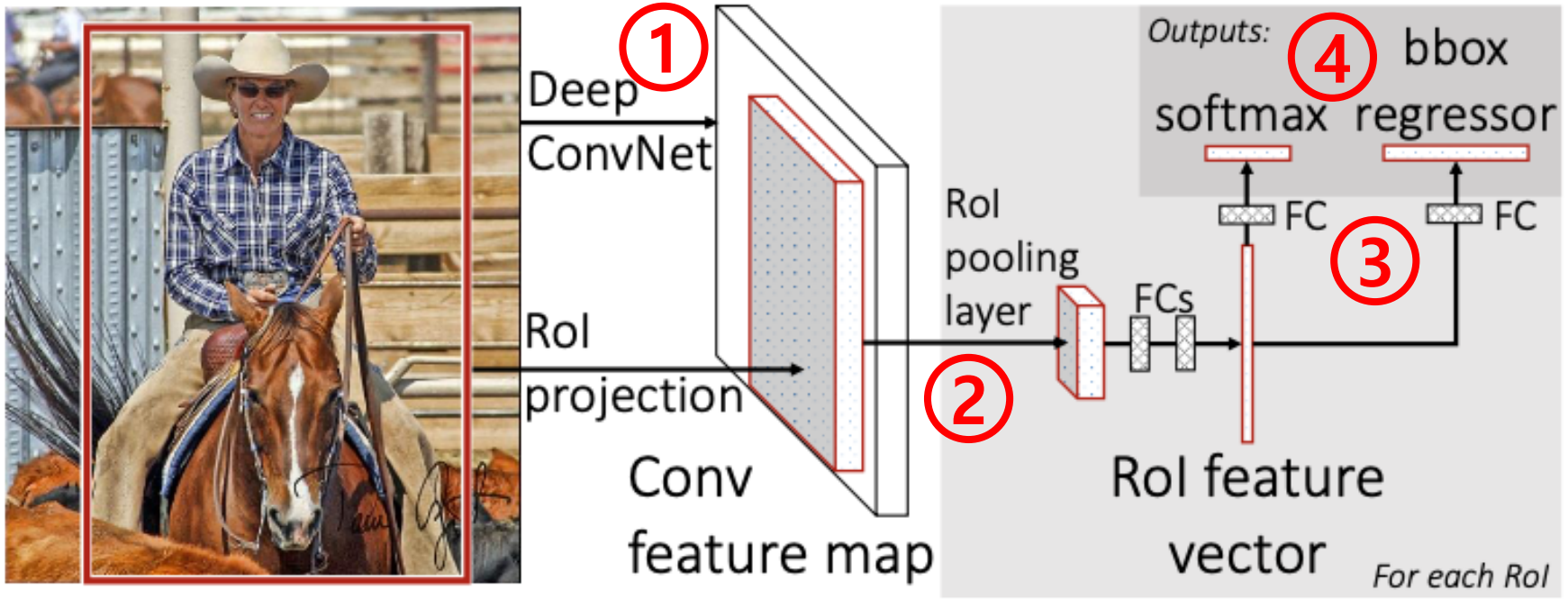
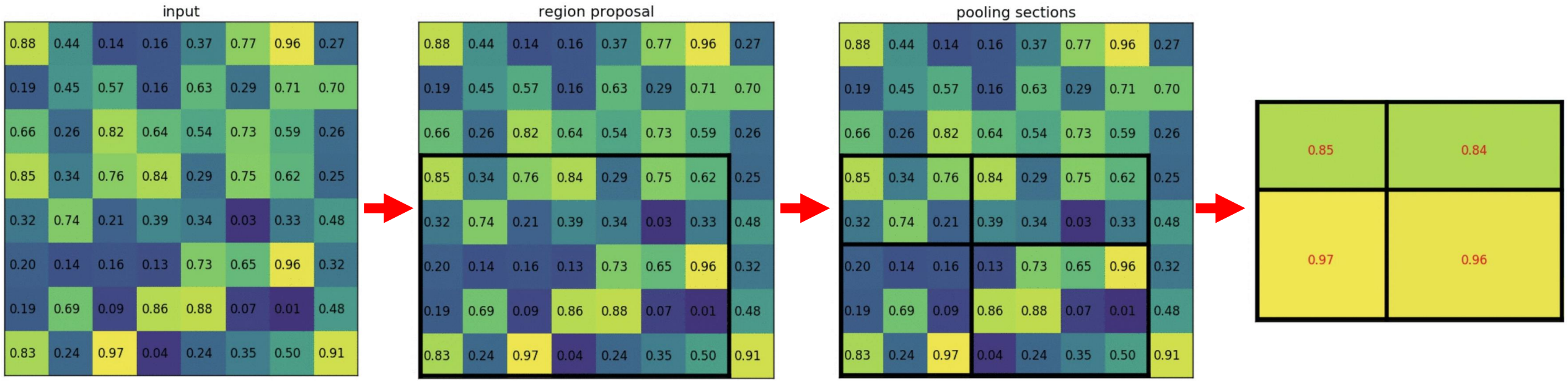


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax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

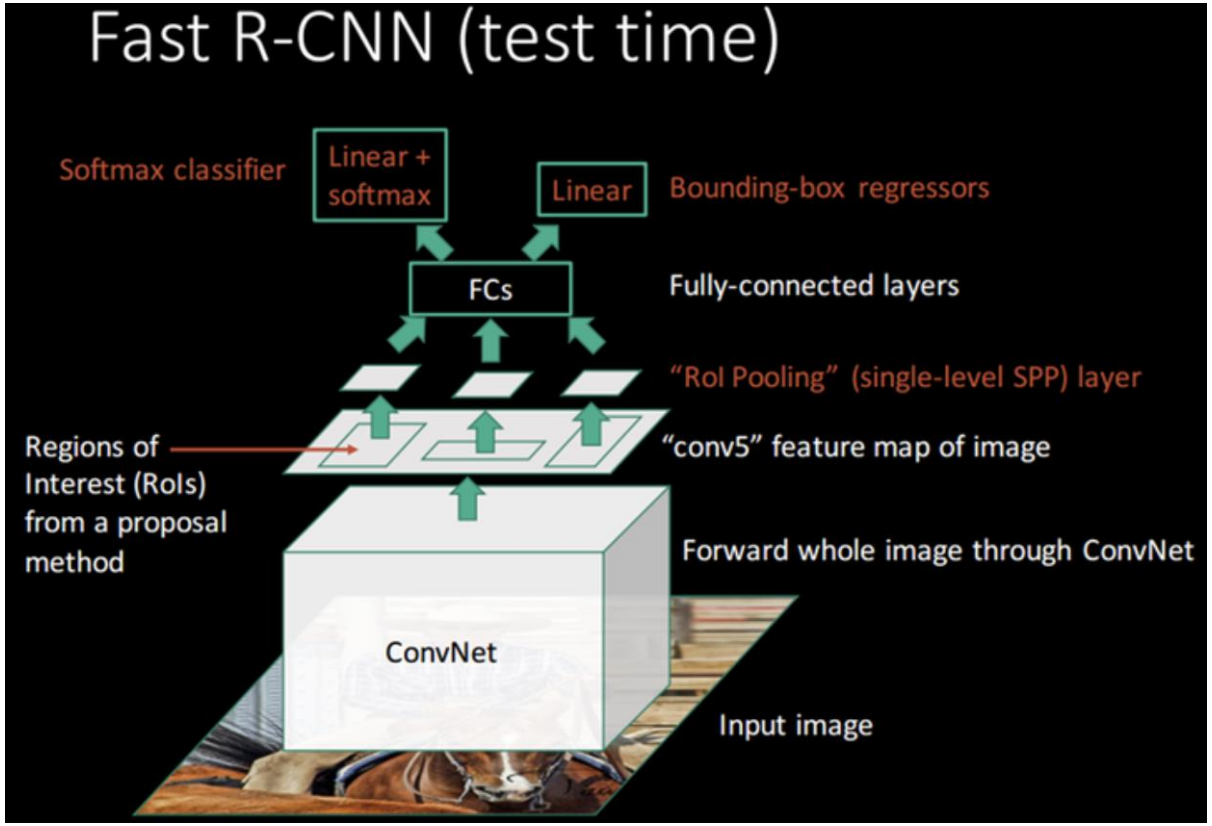
Architecture and Training

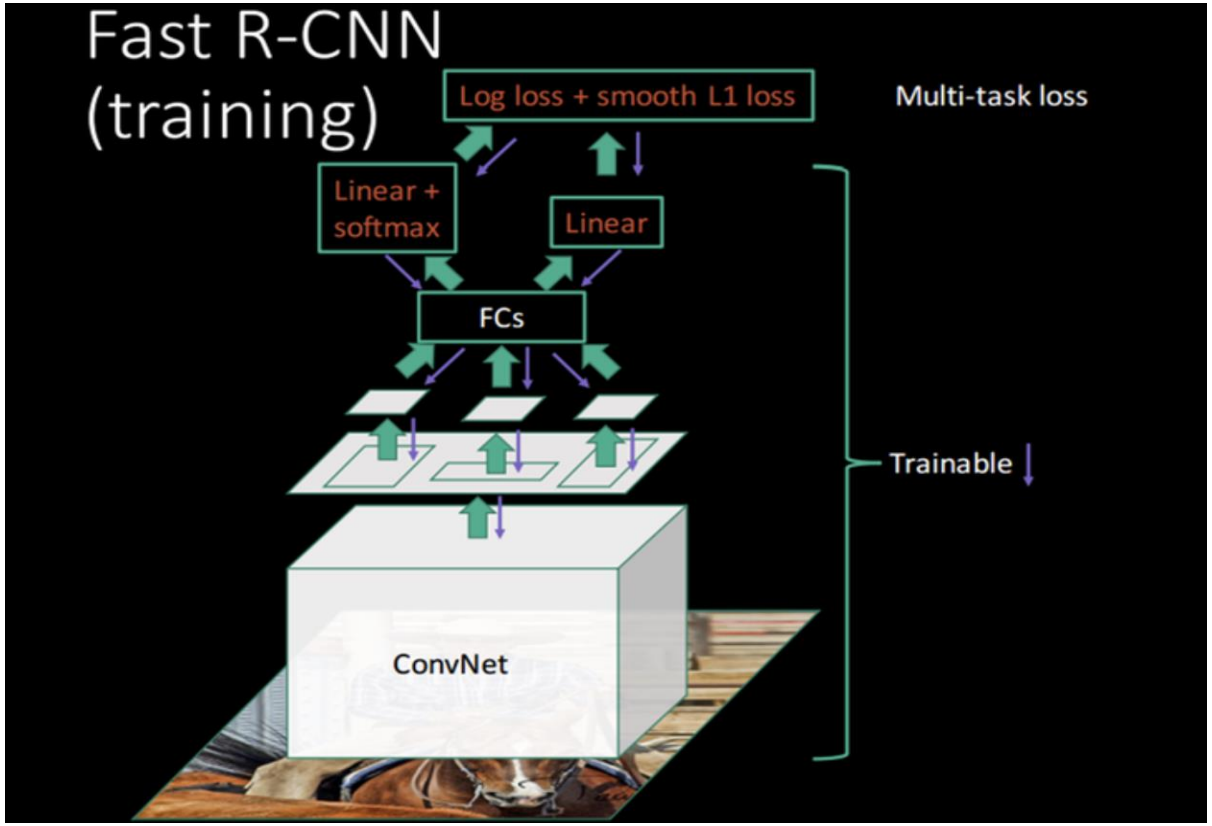
RoI Pooling Layer



Architecture and Training

- Multi-stage가 아니라 Single-stage
- ConvNet 부분까지도 학습이 가능함. (SPPNet의 경우 고정된 파라미터)





Architecture and Training

Multi Task Loss

- 각각의 FC를 통해 classification, regression이 되는됨.

- 이 때 loss는 classification loss와 bounding box regression을 섞은 multi task loss를 구함.

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

p의 경우 softmax의 결과로 k+1개의 확률 값, u는 ground truth.

t는 각 class에 대해서 각각 x, y, w, h 값을 조정하는 tk 값을 가짐.

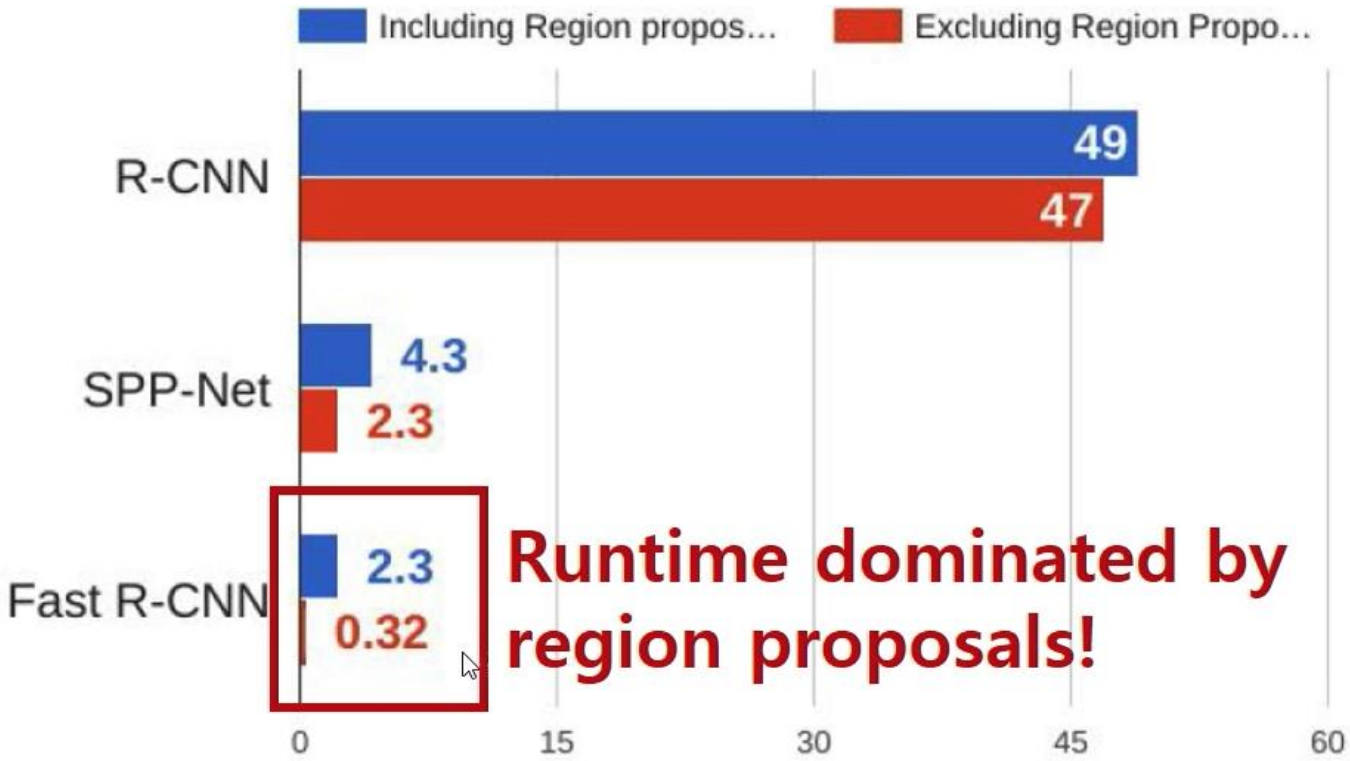
$$L_{\text{cls}}(p, u) = -\log p_u$$

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

Architecture and Training

Test time

- 기존의 R-CNN, SPP-Net에 비해 성능의 향상을 보였지만 region proposal을 위한 연산에서 병목 현상이 발생함.
- 이후 Faster R-CNN에서 Selective Search의 개선이 이루어질 것으로 예상함.



정리

특징

- R-CNN, SPP-Net을 거쳐 single stage로 구현하여 성능의 향상을 이룸.
- Single stage이기 때문에 feature map을 추출하는 convNet까지 학습할 수 있음.
- 하지만 아직 Selective Search를 사용하기 때문에 성능 개선의 여지가 남아있음.

참고 링크

https://yeomko.tistory.com/17

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

https://m.blog.naver.com/PostView.nhn?blogId=laonple&logNo=220776743537&proxyReferer=https:%2F%2Fwww.google.com%2F