Mask R-CNN

Kaiming He | 24 Jan 2018

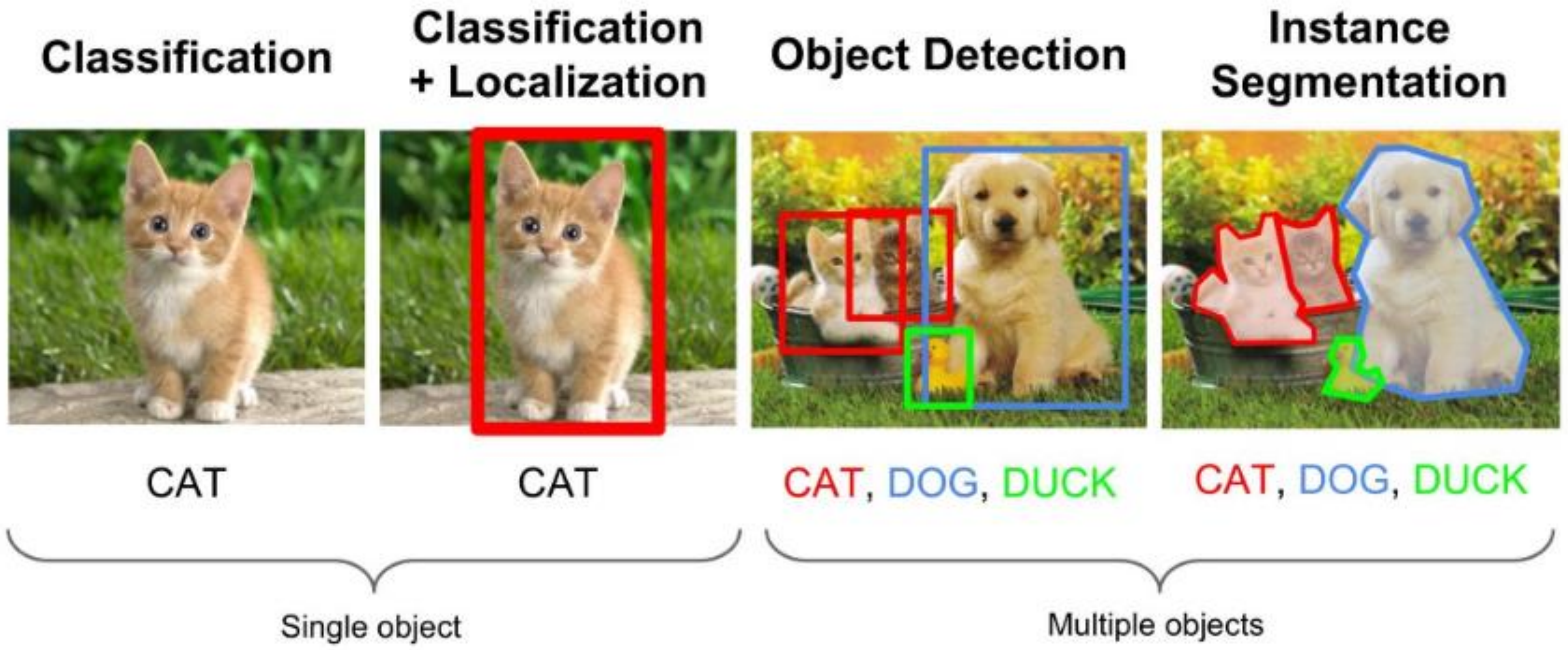
논문 구현

CloseAI팀 이정훈 박준혁 김유철 이상헌

목차

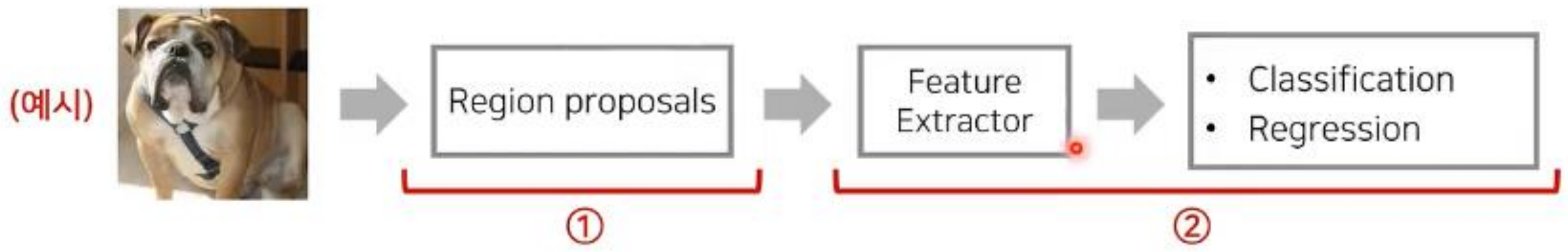
Object Detection

Object Detection



Object Detection 방식

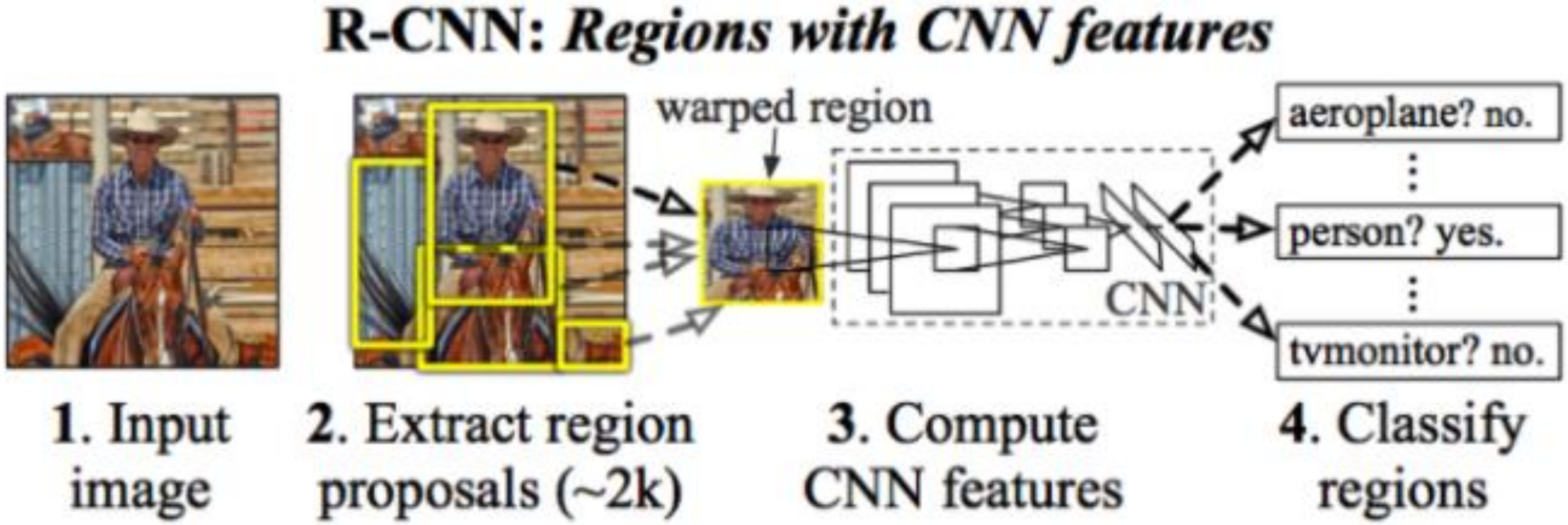
2-Stage

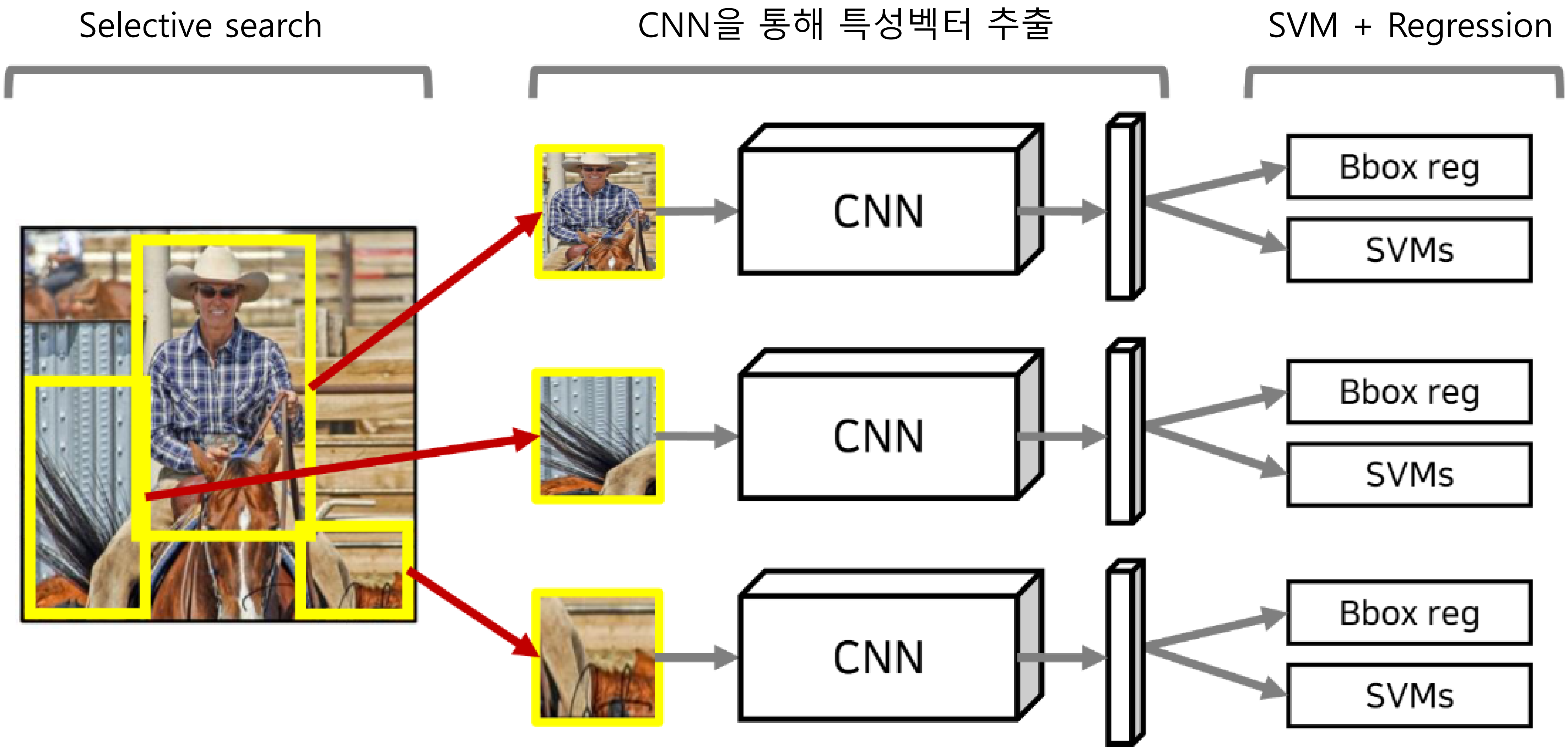


1-Stage

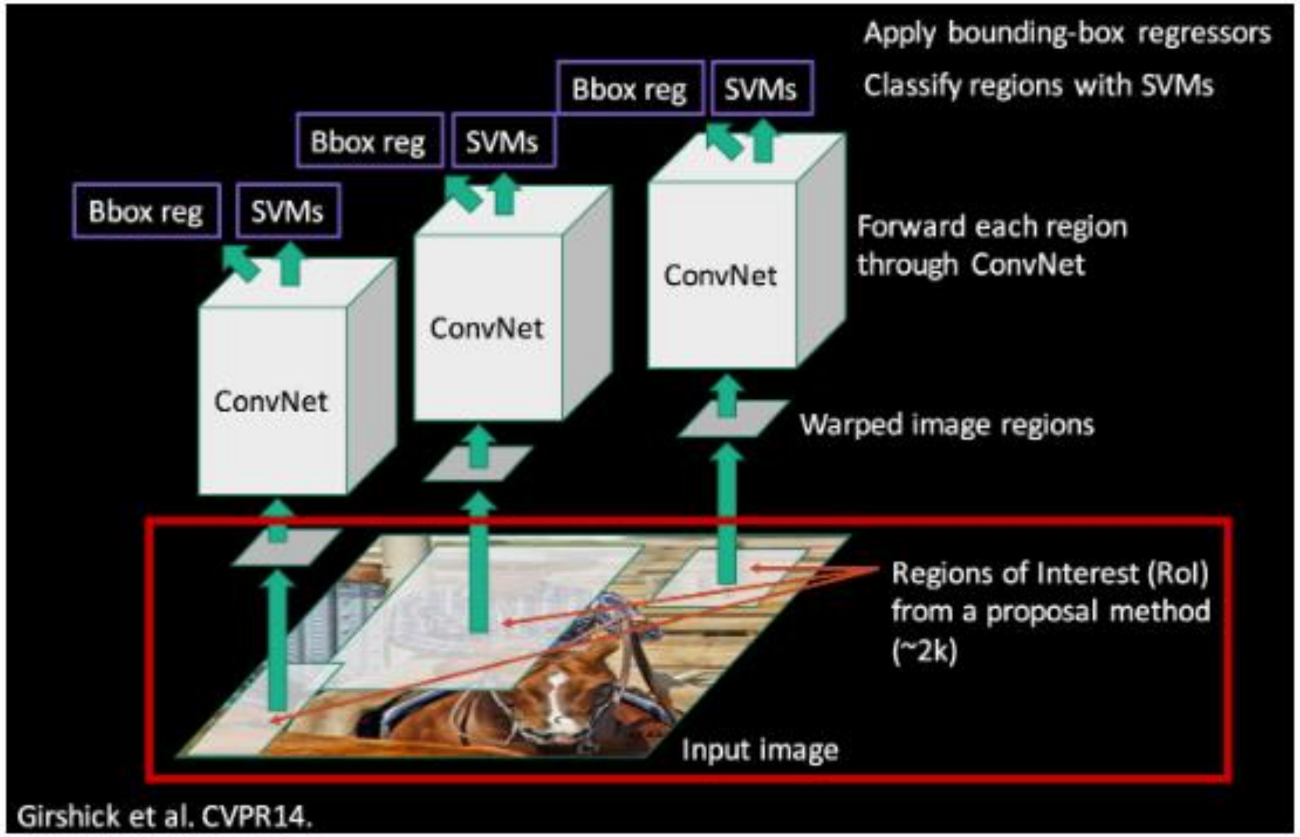
R-CNN

R-CNN Process




Selective search
CNN을 통해 특성벡터 추출
SVM + Regression
CNN
CNN
CNN
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs

Region proposal(영역 제안)



- 물체가 있을 법한 영역
- 기존 모델 sliding window
- Selective Search

Sliding window

- 고정된 크기의 window
- 각 window 위치에서 CNN
- 느리다는 단점

Selective search

- 초기 분할
- 유사 영역 병합
- 지역 제안
- 객체 검출

R-CNN의 한계

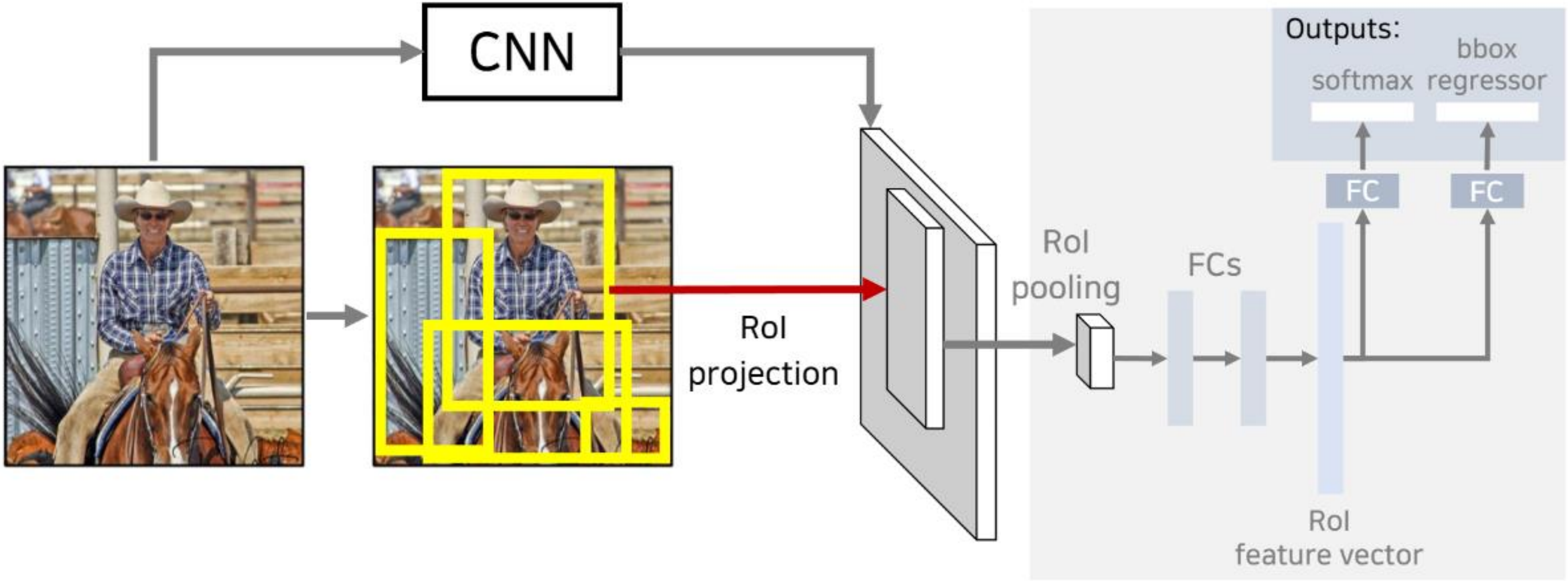
cpu 기반 Selective search 많은 시간 소요

Selective search로 추출한 RoI마다 개별 CNN연산(약 2000 * CNN)

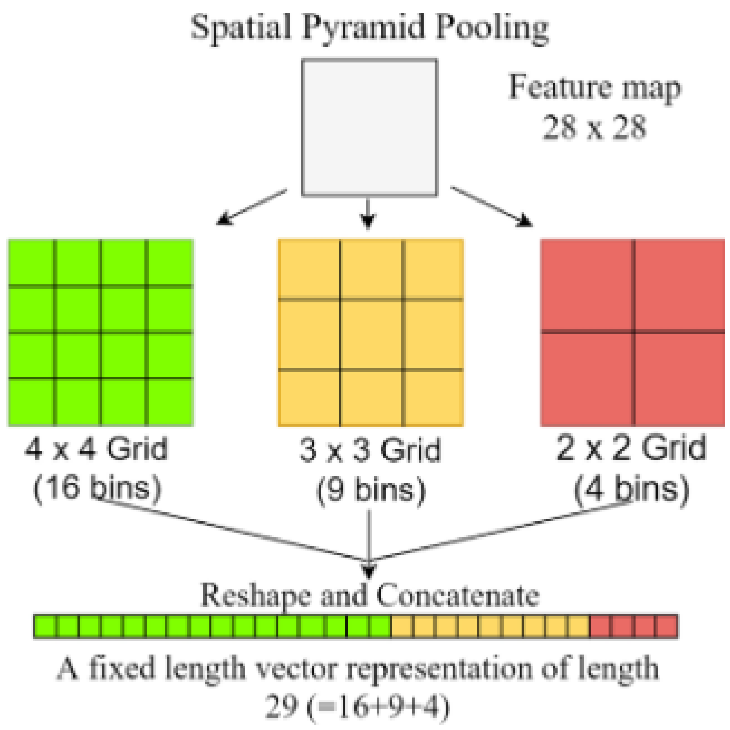
SVM, Bounding Box Regressor가 CNN과 분리
CNN이 고정되므로, SVM, Bounding Box Regressor의 결과로 CNN을 업데이트할 수 없어서 end-to-end 방식으로 학습 불가

Fast R-CNN

Fast R-CNN


CNN
RoI
projection
RoI
pooling
FCs
Outputs:
softmax
bbox
regressor
FC
FC
RoI
feature vector

RoI pooling



SPP (Spatial Pyramid Pooling)

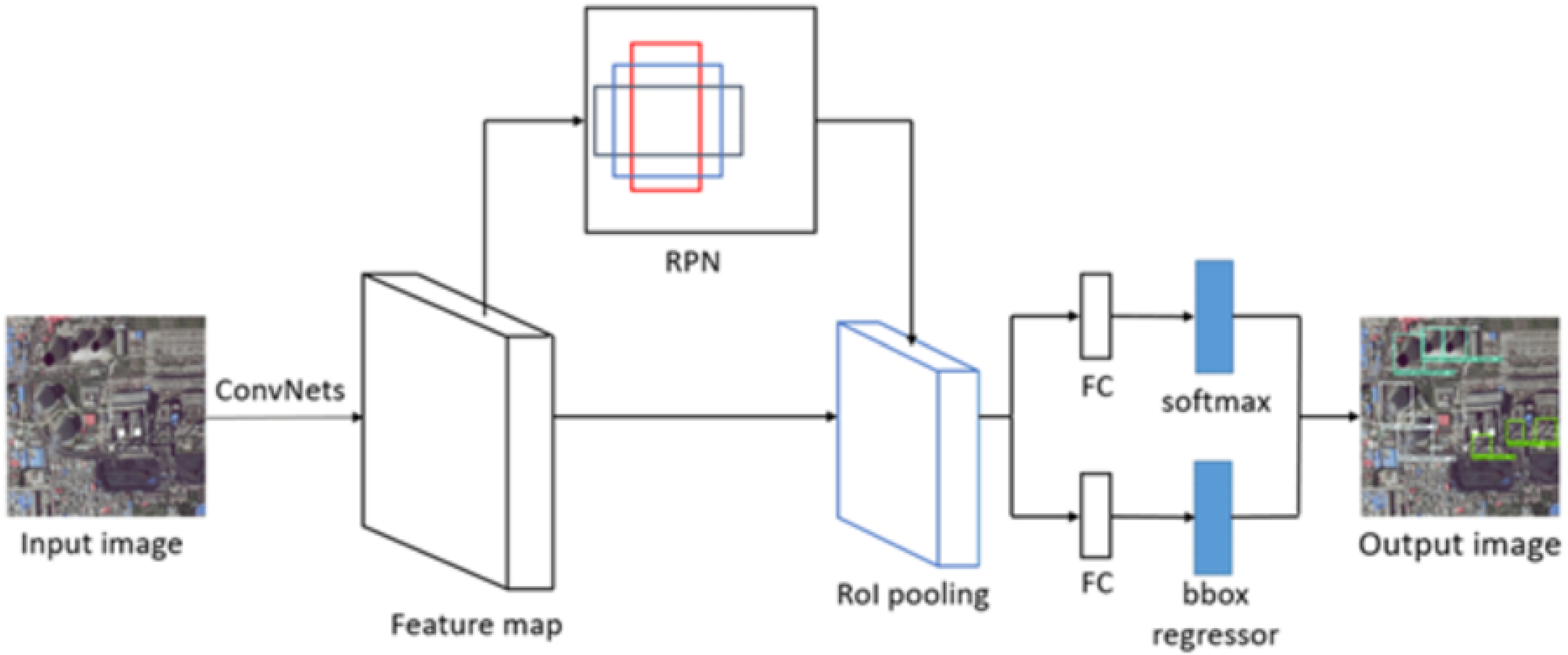
- RoI Pooling에 사용되는 알고리즘
- 4x4 , 2x2, 1x1의 bin들을 가진 각각의 SPP Layer을 통과 후 Concatenate(1차원 특성 벡터)

R-CNN과의 차이 및 한계

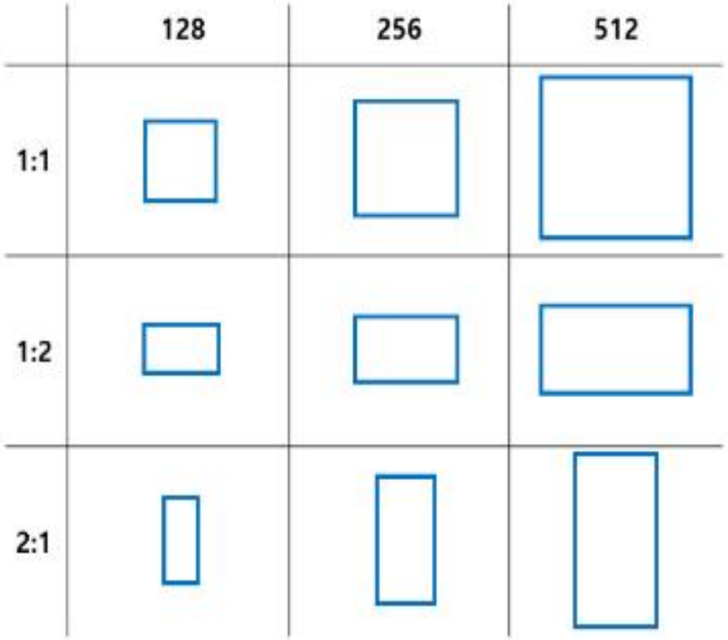
- 약 2000번의 CNN 연산을 한 번의 CNN 연산으로 바꿔 속도 향상
- 단일 손실 함수를 사용. Classification과 Bounding Box Regressor을 동시에 최적화
- 위의 이유로 end-to-end로 학습 가능
- 여전히 cpu 기반 selective search를 사용.

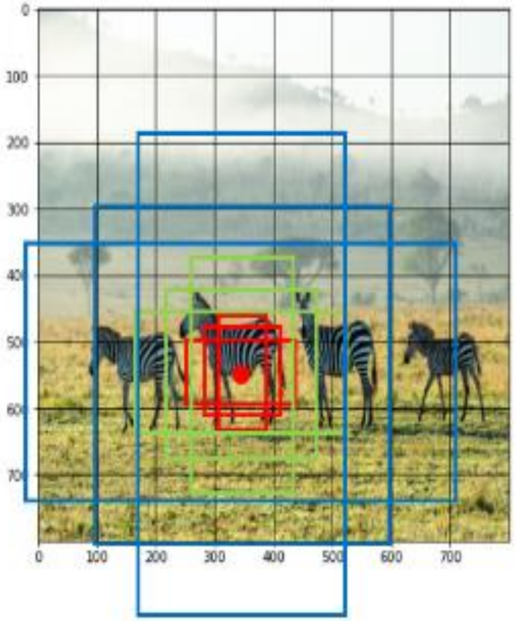
Faster R-CNN

Faster R-CNN


RPN
ConvNets
Input image
Feature map
RoI pooling
FC
softmax
FC
bbox
regressor
Output image

Faster R-CNN

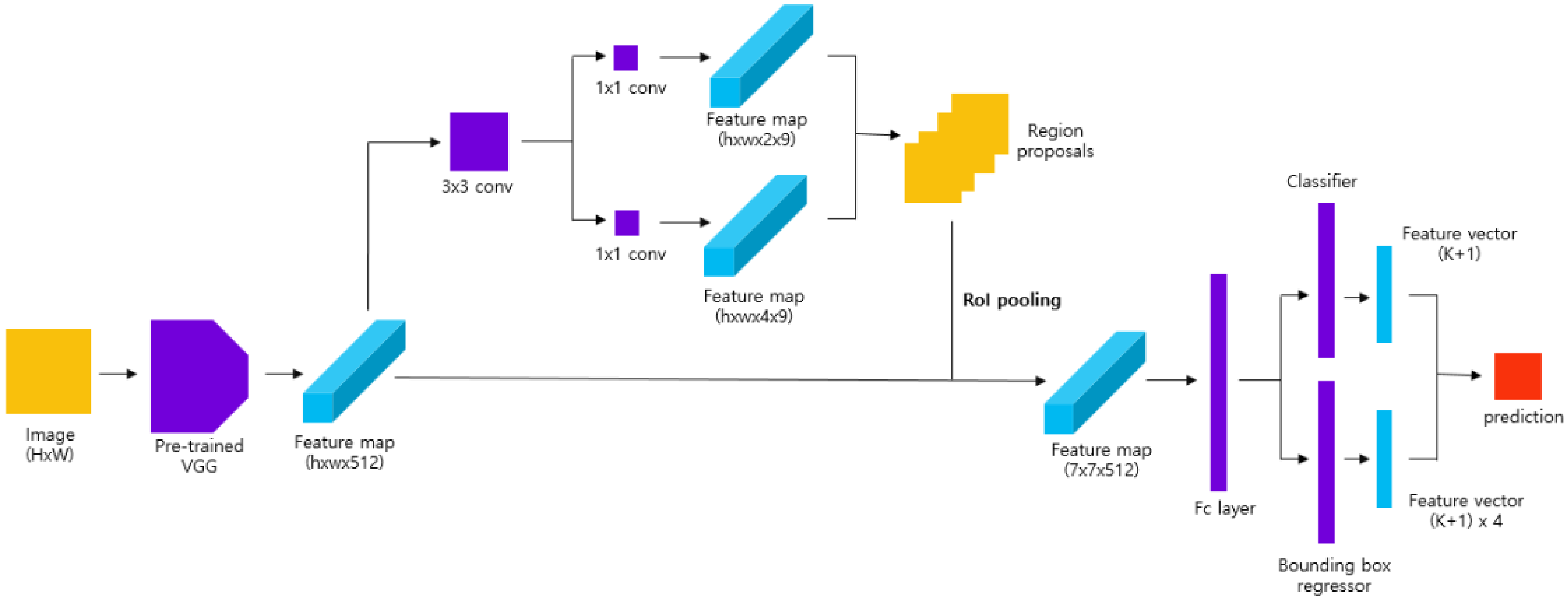




3가지 scale, ratio

각 그리드마다 9개의
anchor box 생성

Faster R-CNN



1x1 conv
Feature map
(hxwx2x9)
Region
proposals
3x3 conv
1x1 conv
Feature map
(hxwx4x9)
Classifier
Feature vector
(K+1)
RoI pooling
Image
(HxW)
Pre-trained
VGG
Feature map
(hxwx512)
Feature map
(7x7x512)
prediction
Fc layer
Feature vector
(K+1) x 4
Bounding box
regressor

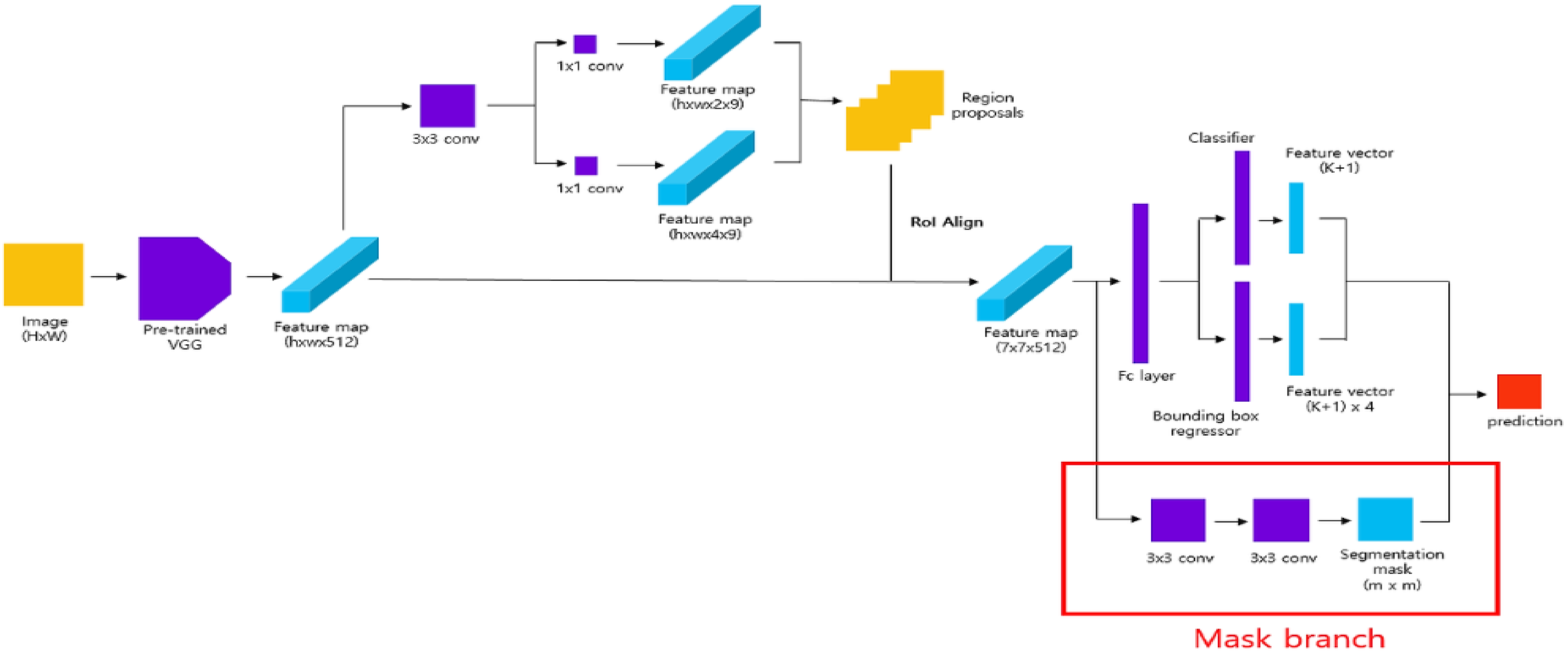
Fast R-CNN 차이점

- Selective Search 대신 RPN을 사용하여 region proposals 생성 속도 개선

- 동일한 CNN 특징 맵을 공유하여 연산 자원 효율 극대화

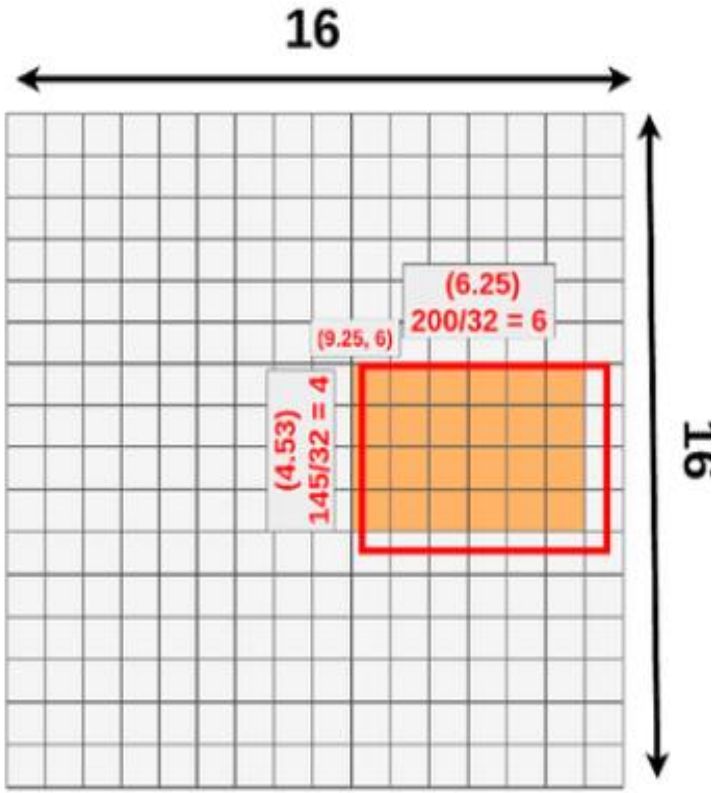
Mask R-CNN

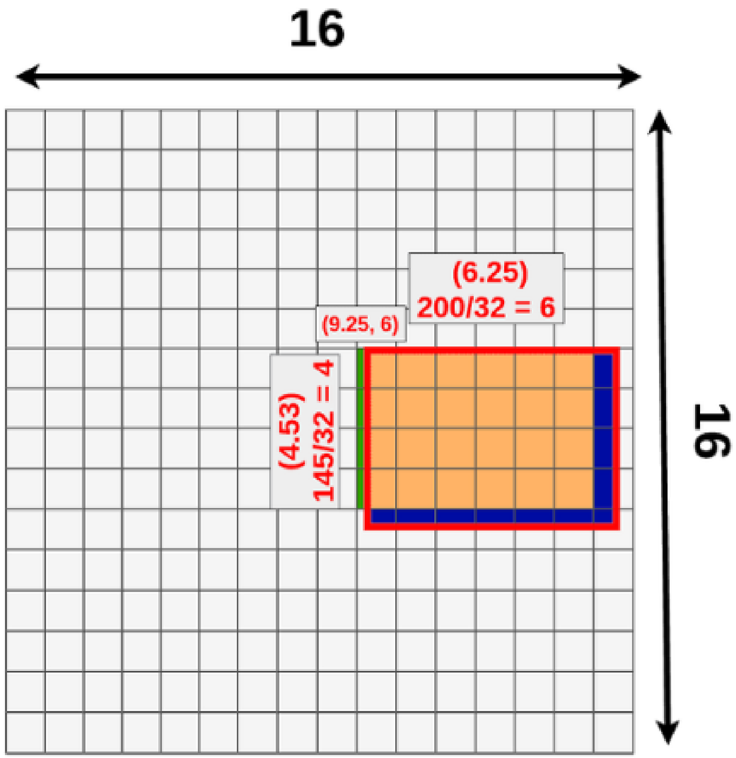
Mask R-CNN



RoI Pooling에서 RoI Align으로 변경
Mask branch 추가

RoI Pooling의 문제점



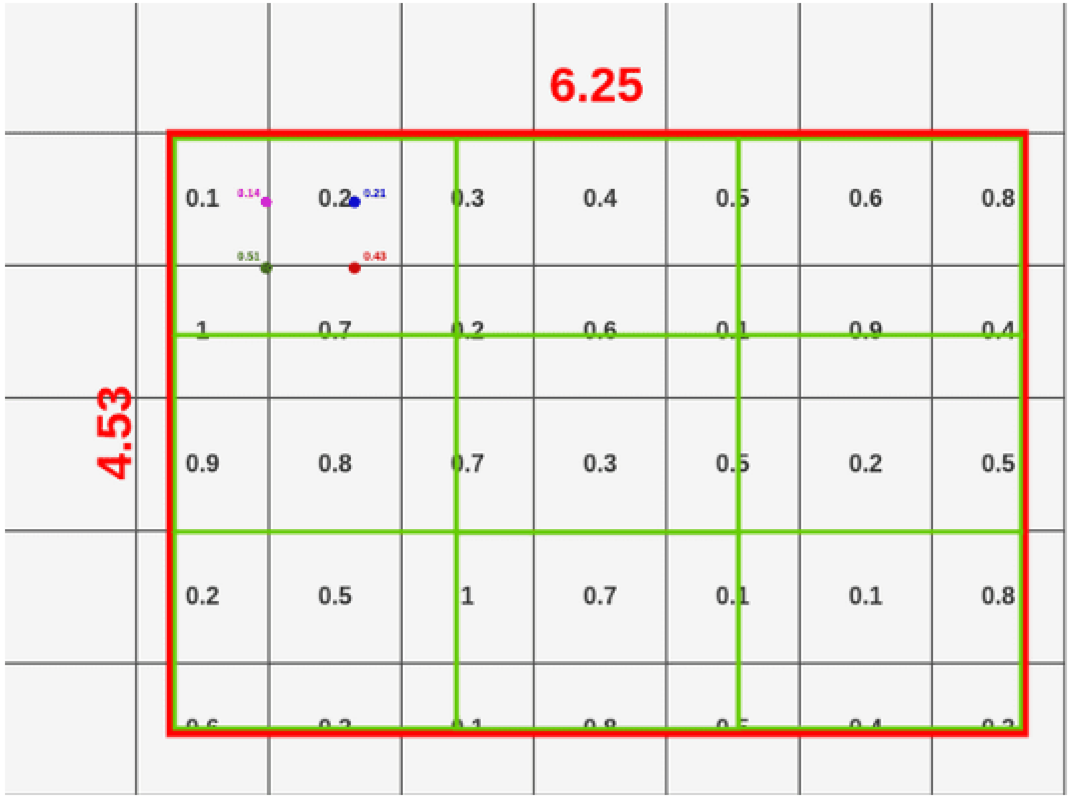


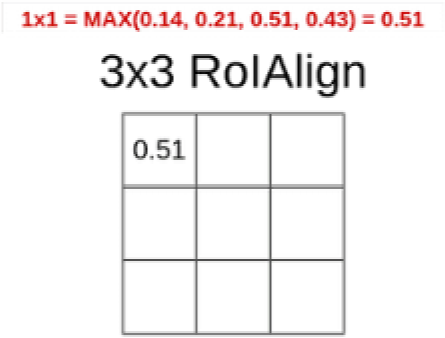
Quantization(양자화) 문제

실수를 정수로 바꾸는 과정에서 데이터 소실 발생

픽셀 단위로 진행하는 segmentation에서는 부정적인 영향을 준다

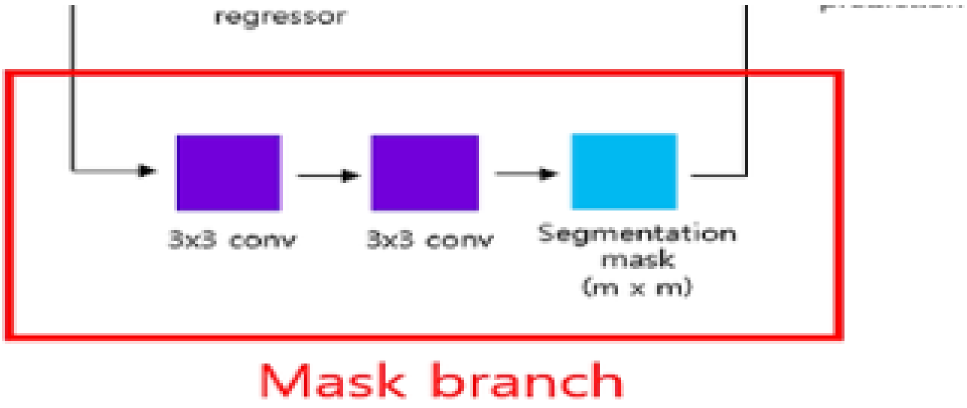
RoI Align

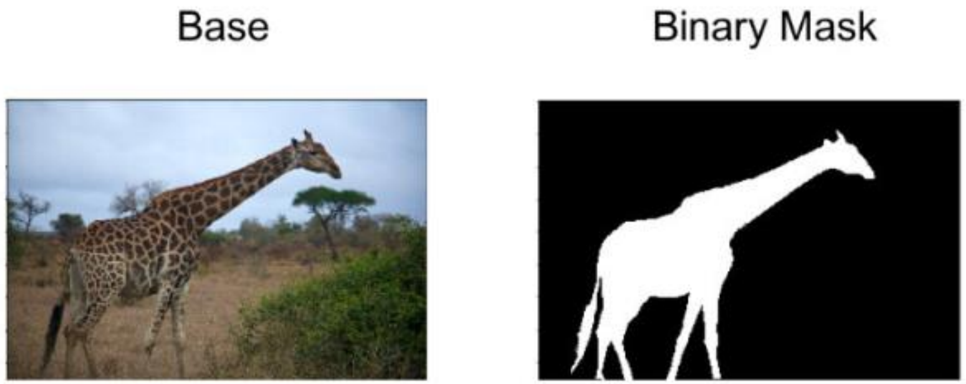




하나의 cell에 있는 4개의 sampling point에 대하여 max pooling

Mask branch



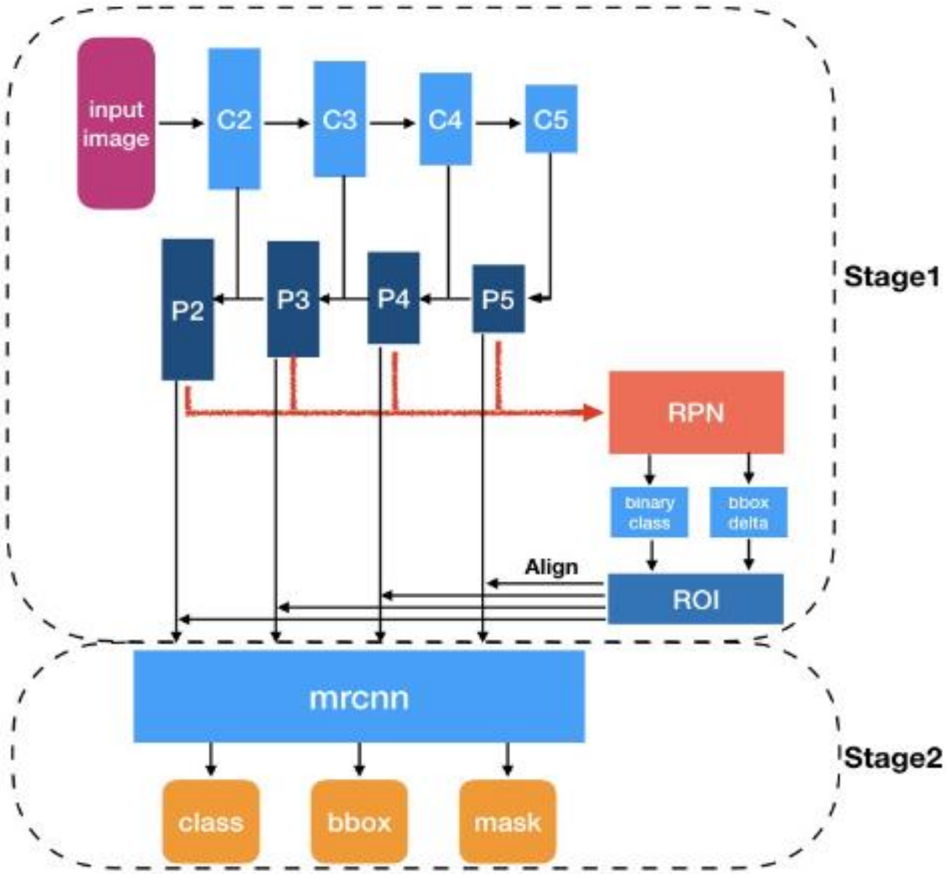


Binary mask

각각의 RoI에 작은 크기의 FCN이 추가된 형태

클래스 단위로 mask를 생성한 후 픽셀이 해당 클래스에 해당되는지 여부를 표시

FPN(Feature Pyramid Network)



ResNet-FPN

다중 스케일 특성(Multi-scale Features)

정확한 객체 검출 및 분할

성능 향상

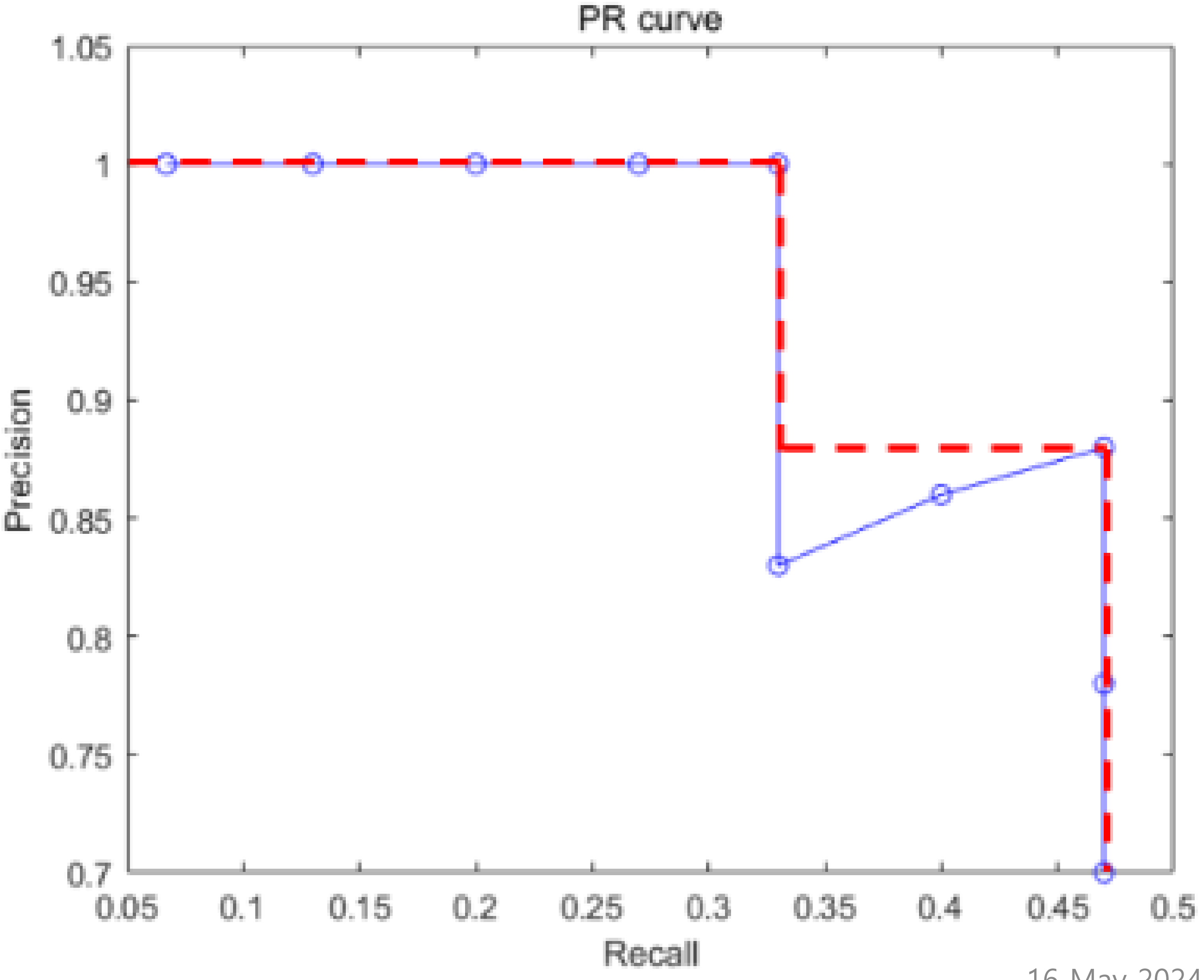
Mask R-CNN 결과

결과

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Table 1. **Instance segmentation** *mask* AP on COCO `test-dev`. MNC [10] and FCIS [26] are the winners of the COCO 2015 and 2016 segmentation challenges, respectively. Without bells and whistles, Mask R-CNN outperforms the more complex FCIS+++, which includes multi-scale train/test, horizontal flip test, and OHEM [38]. All entries are *single-model* results.

평가 지표



AP는 Precision-Recall 곡선 아래의 면적을 계산한 값

모델이 다양한 임계 값에서 얼마나 일관되게 높은 Precision과 Recall을 유지하는지를 평가합니다

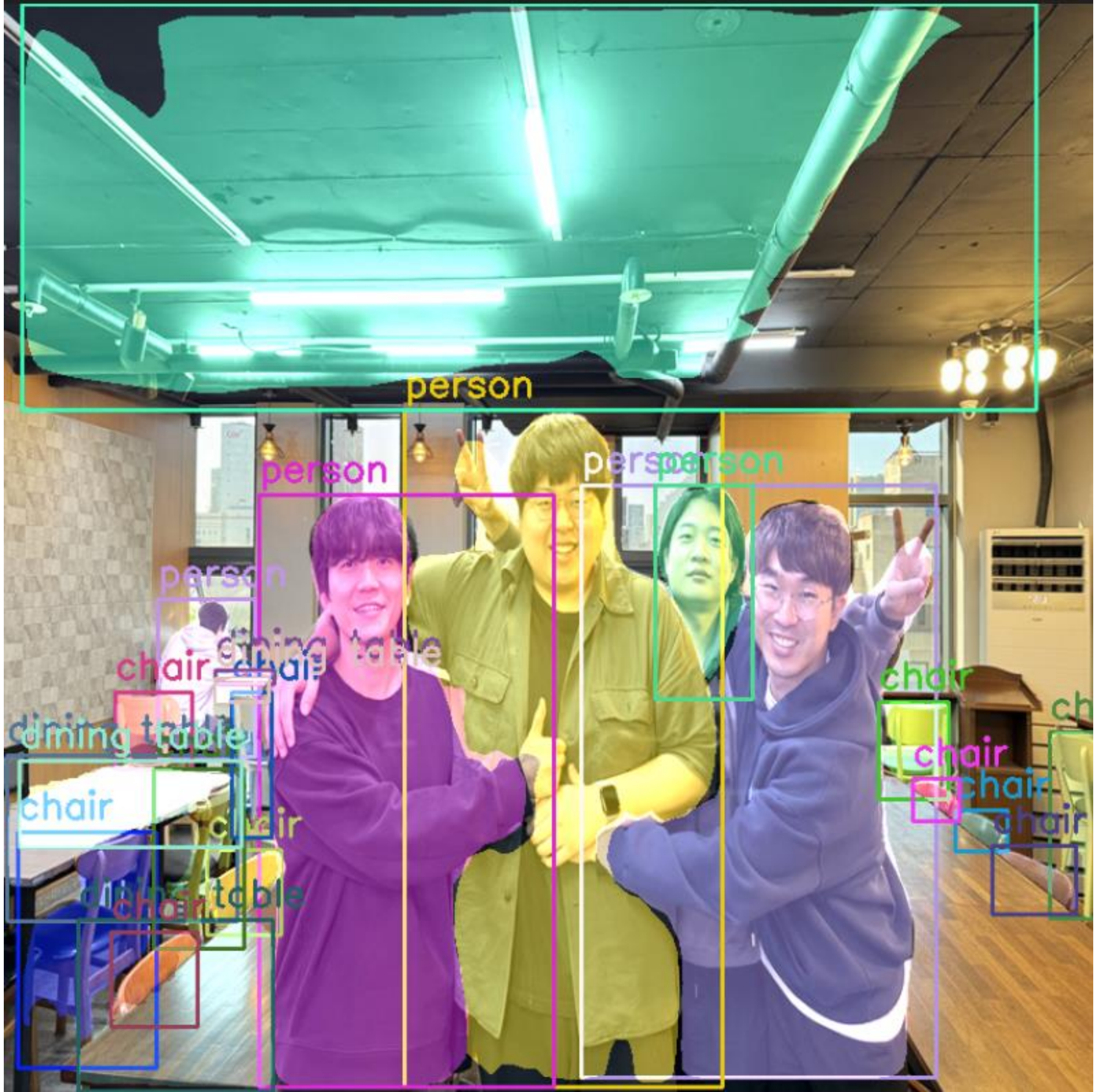
코드 구현

```python
def get_instance_segmentation_model(num_classes):
    # load an instance segmentation model pre-trained on COCO
    model = torchvision.models.detection.maskrcnn_resnet50_fpn(pretrained=True)

    # get the number of input features for the classifier
    in_features = model.roi_heads.box_predictor.cls_score.in_features
    # replace the pre-trained head with a new one
    model.roi_heads.box_predictor = FastRCNNPredictor(in_features, num_classes)

    # now get the number of input features for the mask classifier
    in_features_mask = model.roi_heads.mask_predictor.conv5_mask.in_channels
    hidden_layer = 256
    # and replace the mask predictor with a new one
    model.roi_heads.mask_predictor = MaskRCNNPredictor(in_features_mask,
                                                       hidden_layer,
                                                       num_classes)

    return model
```


person
person
person
person
person
chair
dining table
chair
chair
chair
chair
chair
chair
dining table
chair

Reference

Rich feature hierarchies for accurate object detection and semantic segmentation, Ross B. Girshick, 2014
(https://arxiv.org/pdf/1311.2524.pdf)

Fast R-CNN, Ross B. Girshick, 2015
(https://arxiv.org/pdf/1504.08083.pdf)

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Shaoqing Ren, 2015
(https://arxiv.org/pdf/1506.01497.pdf)

Mask R-CNN, Kaiming He, 2017
(https://arxiv.org/pdf/1703.06870.pdf)

Q&A