Bo Gyum Kim
Department of Social Sciences., Chung-Ang university
BOAZ.
kim.bokeum@gmail.com

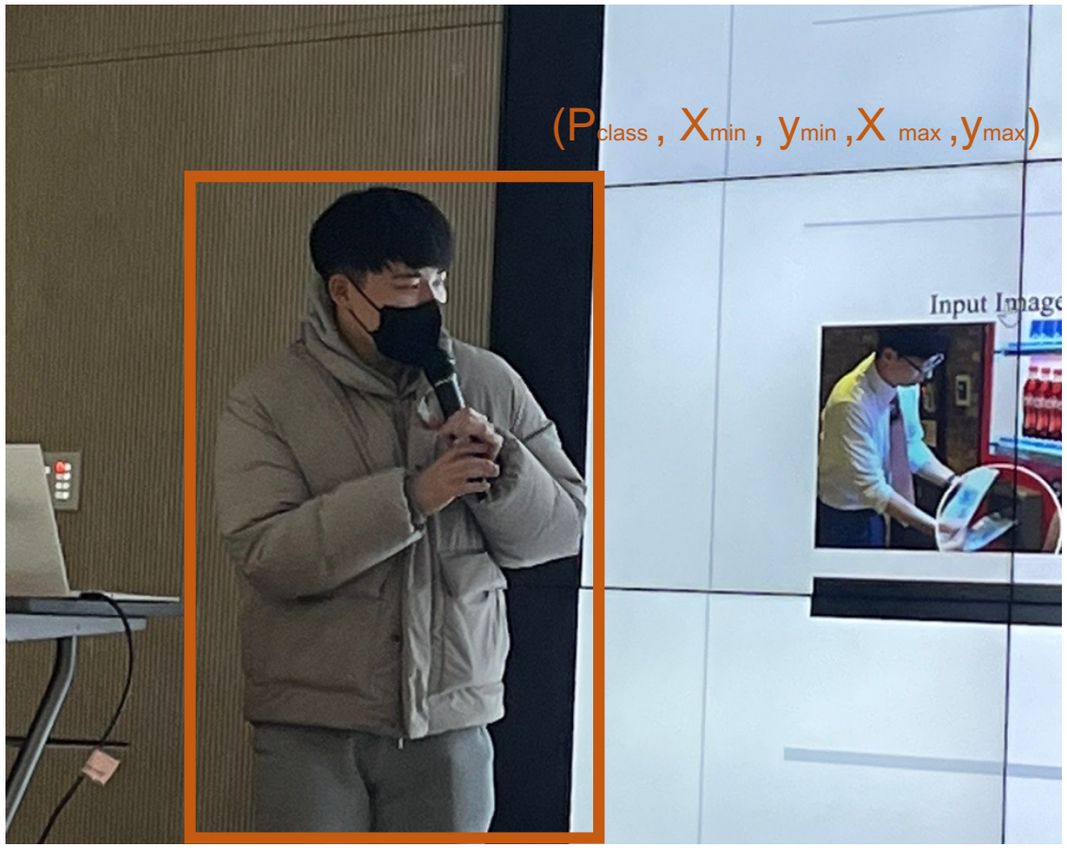
Object Detection Part 1 발제

2023.02.02

OD Overview & 2 Stage-Detector

What is Object Detection(OD)?

- 이미지에서 객체를 찾고, 그 객체가 어떤 클래스인지 분류하는 태스크
  - Classification과 Box Localization이 합쳐진 태스크
  - 한 이미지 안에서 객체의 위치와 수량을 알 수 없다는 점에서 난이도 높음
  - Instance Segmentation, Panoptic Segmentation 태스크 등의 기초가 됨



How to evaluate OD? - Overview

- 속도
  - FPS (Frames per Second) : 1초에 얼마나 많은 이미지들을 처리할 수 있는가
  - FLOPs (Floating Point Operations) : 연산량이 얼마나 많은가

- 성능
  - mAP (mean Average Precision) : 각 클래스당 AP의 평균
    - IOU
    - PR Curve
    - AP

How to evaluate OD? - mAP

- IOU (Intersection Over Union)
  - OD에서 Precision Score 산출의 기본 지표
  - Ground Truth Bounding Box와 모델이 예측한 Bounding Box의 유사성 평가
  - (GT 영역과 예측 영역의 교집합) ÷ (GT 영역과 예측 영역의 합집합) 계산
  - IOU 값의 임계치를 설정하여 넘으면 True, 미달하면 False인 방식으로 평가

IOU 40

IOU 60

- IoU 50
  - IOU 40 : False
  - IOU 60 : True
- IoU 40
  - IOU 40 : True
  - IOU 60 : True
- IoU 70
  - IOU 40 : False
  - IOU 60 : False

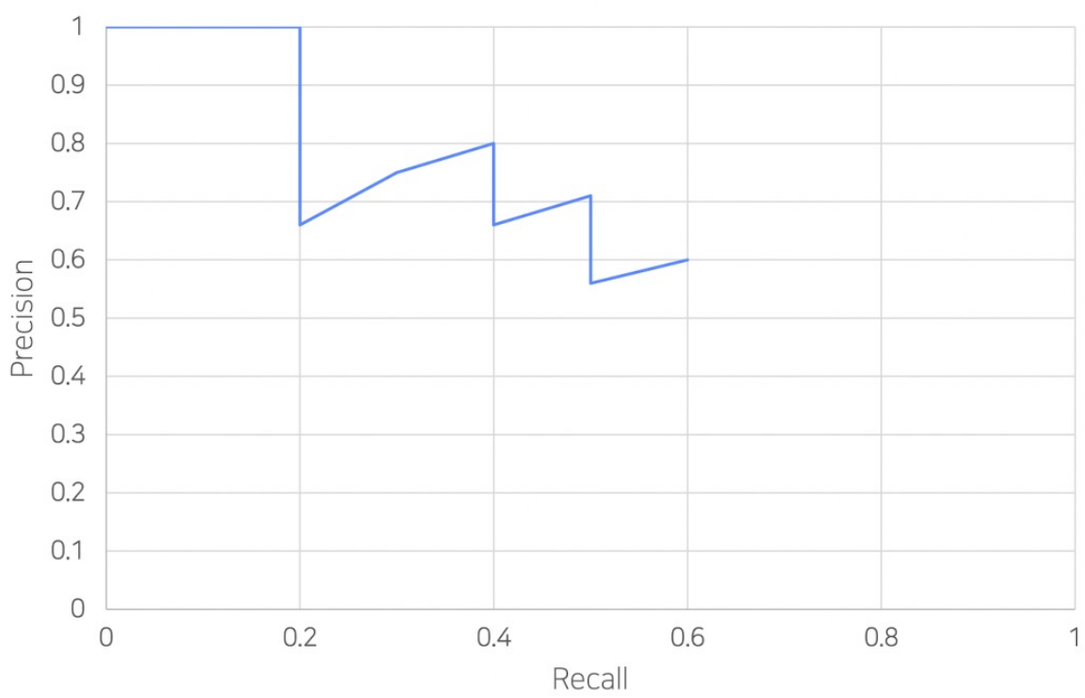
Ground Truth

Predict

How to evaluate OD? - mAP

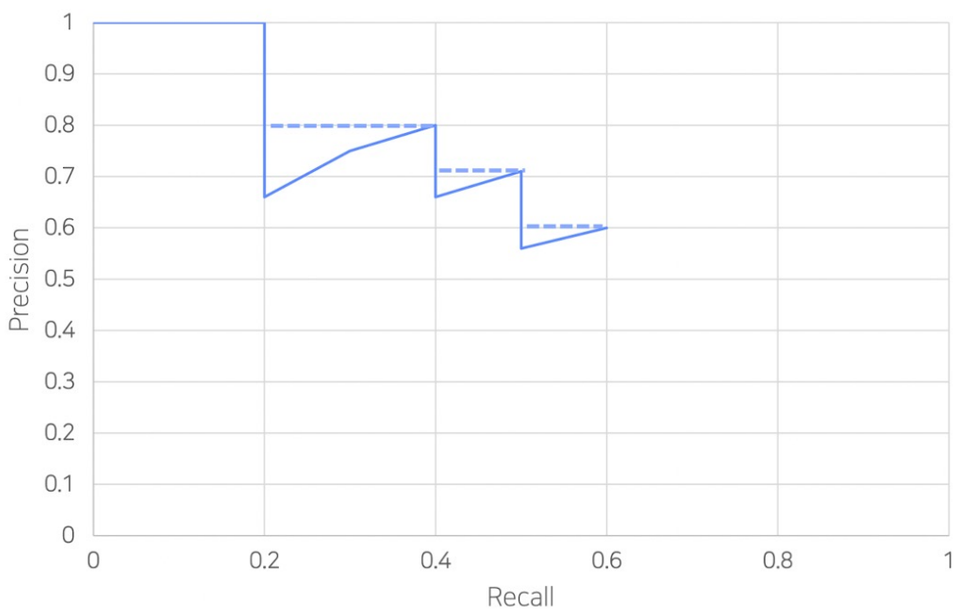
- PR Curve (Precision Recall Curve)
  - IOU 지표를 바탕으로 나온 precision, recall, confidence score를 바탕으로 작성된 그래프
  - 작성 방법
    - 모든 예측에 대해 confidence score 내림차순 정렬하여 누적 TP (True Positive), 누적 FP (False Positive) 계산
    - 각 행의 누적 TP와 누적 FP로 precision, recall 계산
    - X축이 recall, Y축이 precision인 그래프 작성

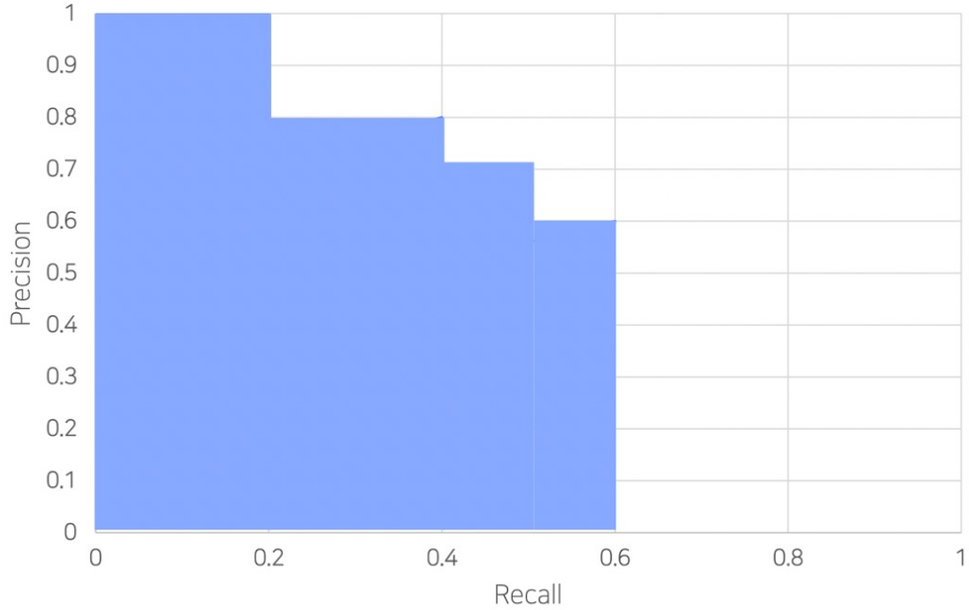
	Category	Confidence	TP / FP	누적 TP	누적 FP	Precision	Recall
8	Plastic	95%	TP	1	-	1/1 = 1	1/10 = 0.1
9	Plastic	90%	TP	2	-	2/2 = 1	2/10 = 0.2
7	Plastic	82%	FP	2	1	2/3 = 0.66	2/10 = 0.2
6	Plastic	80%	TP	3	1	3/4 = 0.75	3/10 = 0.3
1	Plastic	72%	TP	4	1	4/5 = 0.8	4/10 = 0.4
10	Plastic	70%	FP	4	2	4/6 = 0.66	4/10 = 0.4
5	Plastic	60%	TP	5	2	5/7 = 0.71	5/10 = 0.5
3	Plastic	41%	FP	5	3	5/8 = 0.63	5/10 = 0.5
6	Plastic	32%	FP	5	4	5/9 = 0.56	5/10 = 0.5
4	Plastic	10%	TP	6	4	6/10 = 0.60	6/10 = 0.6



How to evaluate OD? - mAP

- AP (Average Precision)
  - PR Curve의 면적을 계산
  - 단, 그대로 계산하지 않고 recall 구간별로 cell을 채워 넣어 계산





How to evaluate OD? - mAP

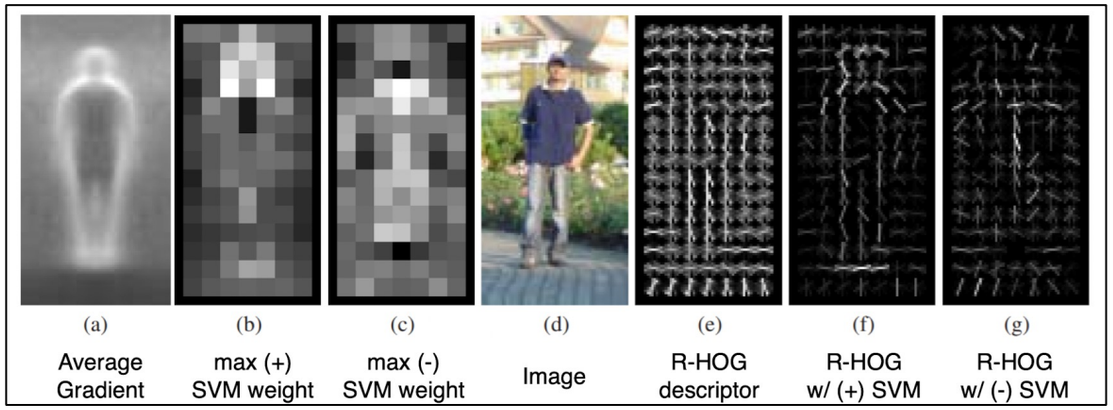
- mAP (mean Average Precision)
  - OD는 여러 개의 class에 대해서 분류하는 모델
  - 따라서 각 class에 대한 AP값을 전체적으로 고려하기 위해 평균을 사용

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

$AP_k = the\ AP\ of\ class\ k$
n = the number of classes

2 Stage Detector – Traditional Methods

- Gradient-based Detector
  - 각 객체의 경계선의 특징 모델링
- Selective Search
  - Over-segmentation 이후 서서히 영역을 합침

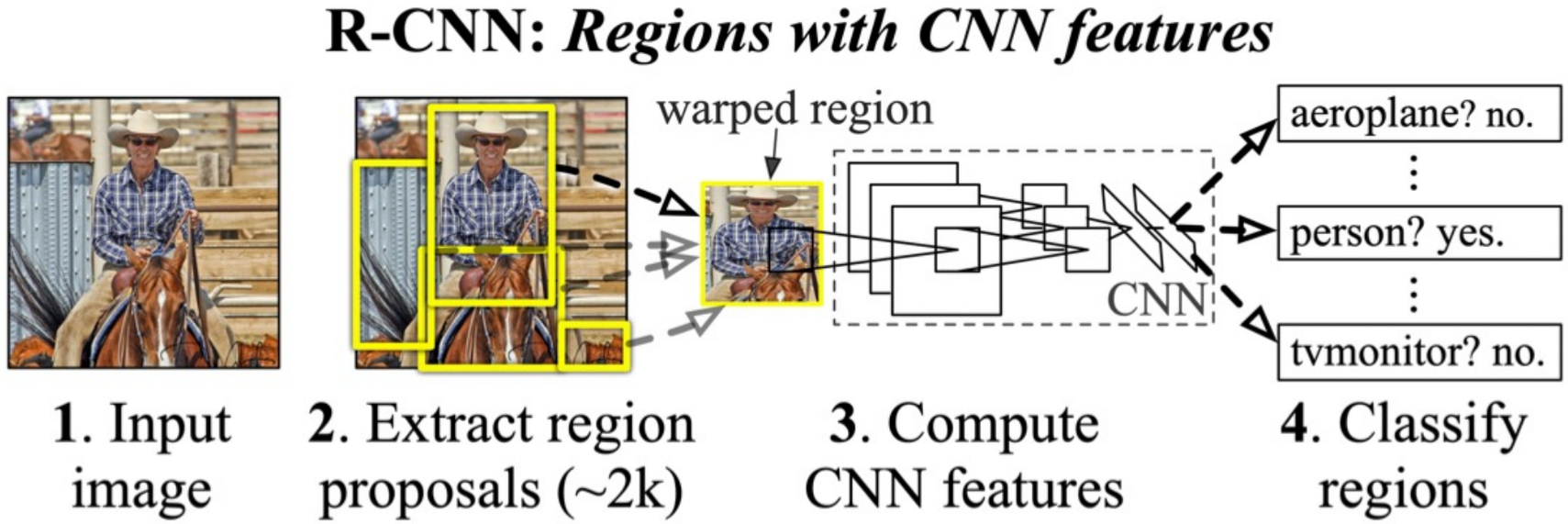


Gradient-based Detector

Selective Search

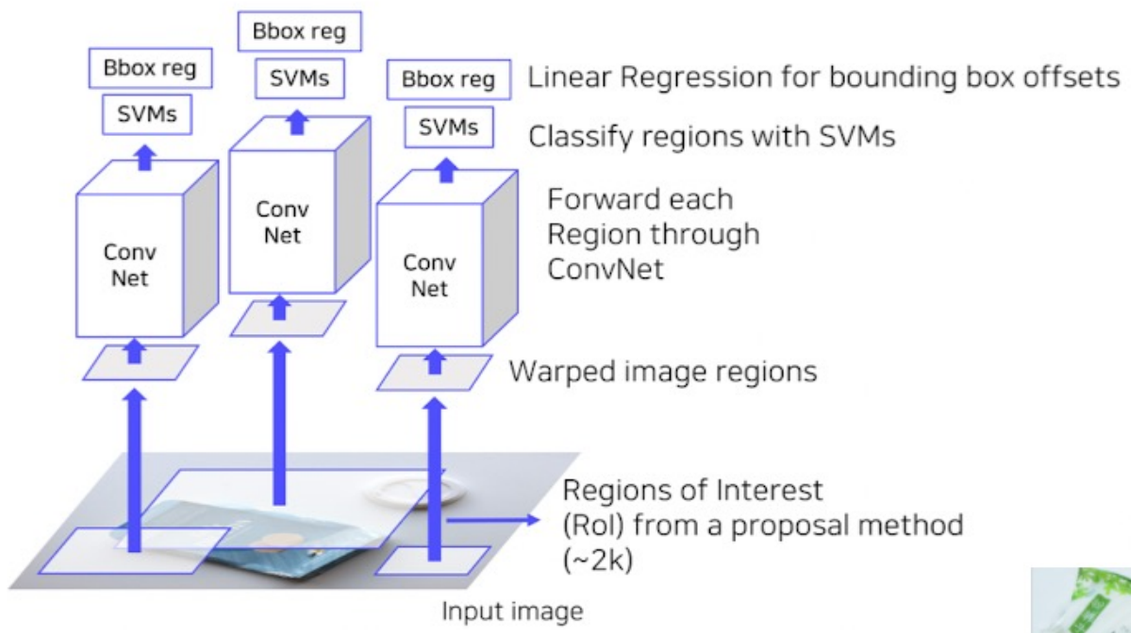
2 Stage Detector – R-CNN

- R-CNN: Regions with CNN features
- 최초로 뉴럴 네트워크를 이용한 OD 모델
  - 부분적으로만 뉴럴 네트워크를 이용하였지만 이후 End-to-End NN 모델로 개선
  - 많은 OD 모델들이 R-CNN을 바탕으로 연구를 확장



2 Stage Detector – R-CNN

- ❑ Pipeline
  - ■ Step 1: 입력 이미지 받기
  - ■ Step 2: Selective Search를 통해 약 2000개의 ROI(Region of Interest) 추출
  - ■ Step 3: ROI의 크기를 모두 동일한 사이즈로 wraping
    - ▪ CNN의 마지막인 FC Layer의 입력 사이즈가 고정이기 때문
  - ■ Step 4: ROI를 CNN에 넣어 feature 추출
    - ▪ Pretrained된 AlexNet 구조 활용
  - ■ Step 5-1: CNN의 결과 feature를 SVM에 넣어 class 분류 시행
    - ▪ Input: 2000 x 4096 features
    - ▪ Output: Class (C+1) + Confidence Score
  - ■ Step 5-2: CNN의 결과 feature를 Regression 모델을 통해 Bounding Box 예측



2 Stage Detector – R-CNN

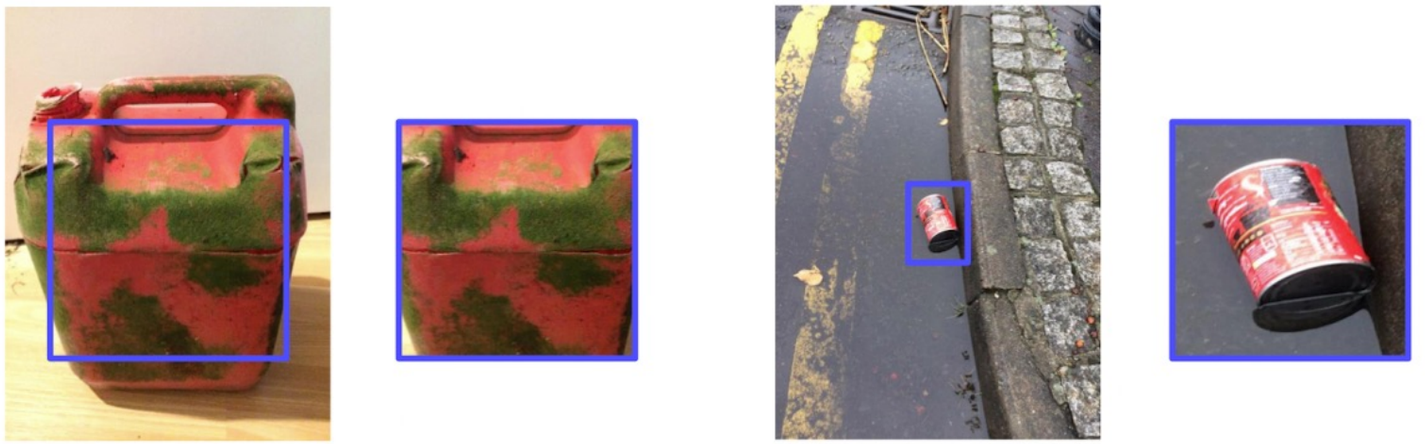
- Training
  - AlexNet
    - Domain specific finetuning
    - Dataset
      - IoU > 0.5: positive samples
      - IoU < 0.5: negative samples
      - Positive samples 32, Negative samples 96
  - Bbox Regressor
    - Dataset
      - IoU > 0.6: positive samples
    - Loss function
      - MSE Loss
  - Linear SVM
    - Hard negative mining
      - 배경으로 식별하기 어려운 샘플들을 강제로 다음 배치의 negative sample로 mining
    - Dataset
      - GT: positive samples
      - IoU < 0.3: negative samples
      - Positive samples 32, Negative samples 96

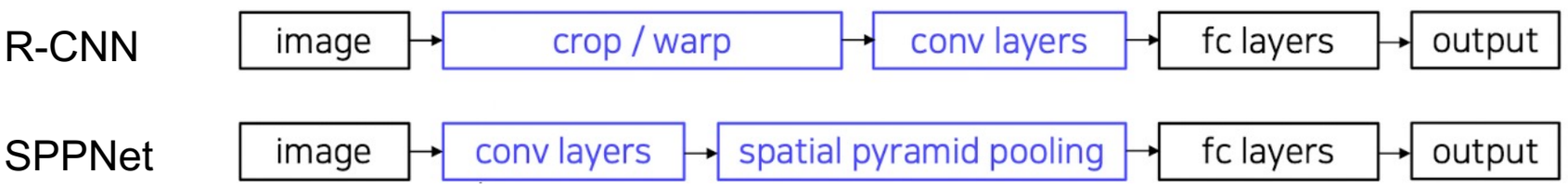
2 Stage Detector – R-CNN

- Shortcomings of R-CNN
  - 2000개의 region이 각각 CNN 통과
  - 강제 wraping으로 인한 성능 하락 가능성
  - CNN, SVM classifier, bounding box regressor 따로 학습
  - End-to-End 모델이 아님

2 Stage Detector – SPPNet

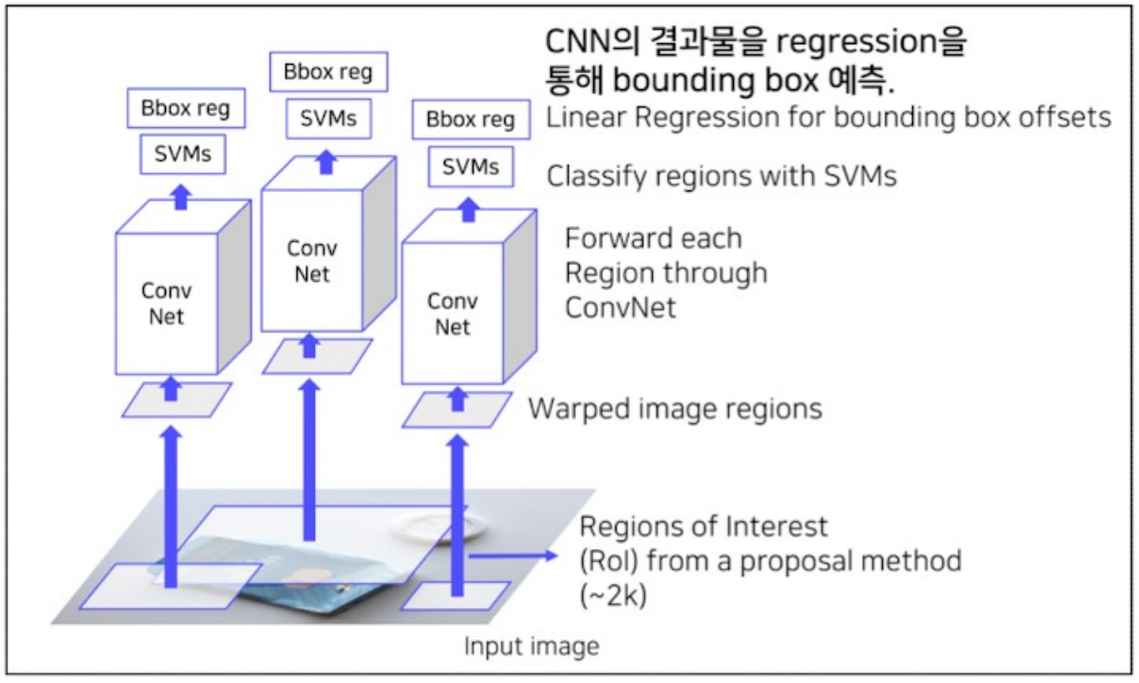
- SPPNet : Spatial Pyramid Pooling Net
  - Convolution Network의 입력 이미지 사이즈 고정으로 인한 이미지 크기 강제 조정 한계 극복
  - ROI마다 CNN을 통과하여 연산량 급증 한계 극복



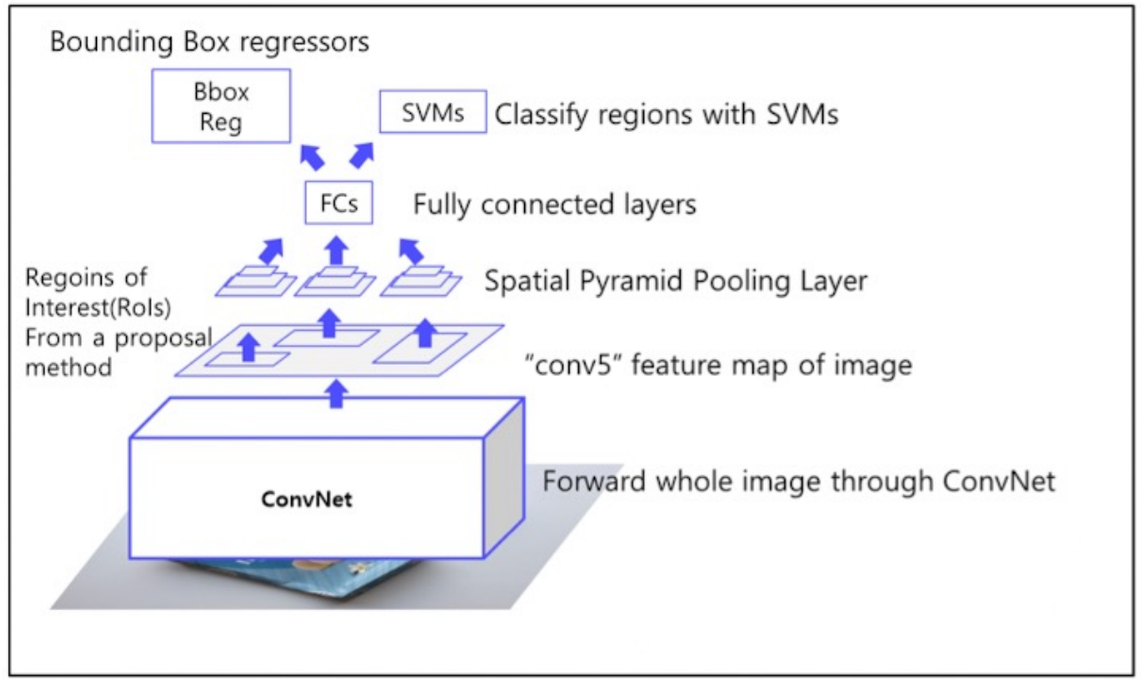


2 Stage Detector – SPPNet

- Pipeline
  - Step 1: 이미지 전체를 CNN에 통과 시켜 feature map 추출
  - Step 2: Feature map에서 ROI 추출
  - Step 3: Spatial Pyramid Pooling Layer를 통해 다양한 크기의 ROI로부터 고정된 사이즈의 feature vector 획득
  - Step 4: FC layer 통과
  - Step 5-1: SVM classifier로 class 분류
  - Step 5-2: Bbox regressor로 bbox 위치 예측 및 위치 세부 조정



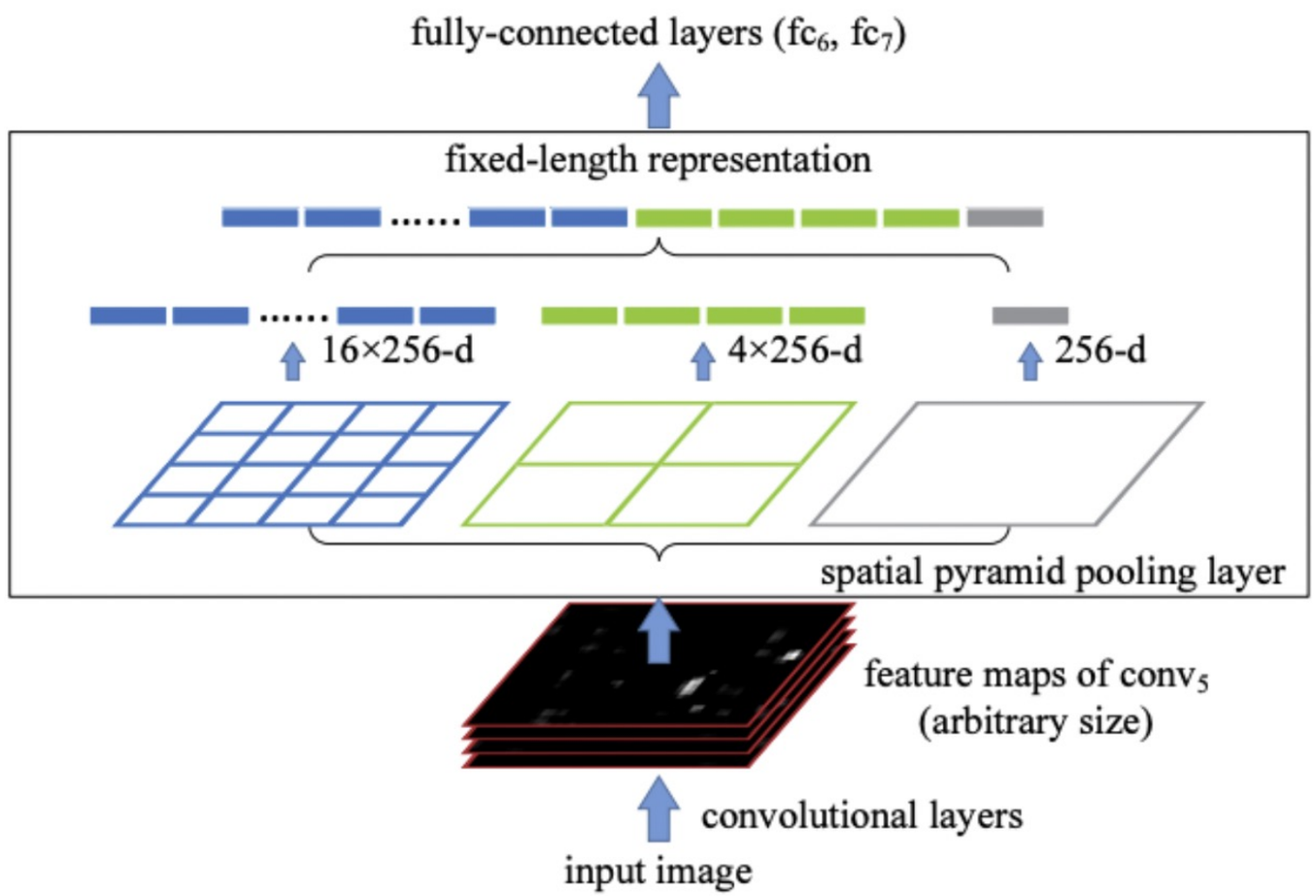
R-CNN



SPPNet

2 Stage Detector – SPPNet

- Spatial Pyramid Pooling
  - 다양한 크기의 ROI로부터 고정된 사이즈의 feature vector 획득할 수 있도록 함
  - Target feature map 사이즈를 정함
  - ROI의 영역을 binning, 구간화된 각 영역에 대해서 pooing하여 feature vector 추출
  - 출력된 feature vector들을 concat하여 고정된 사이즈의 feature vector 추출



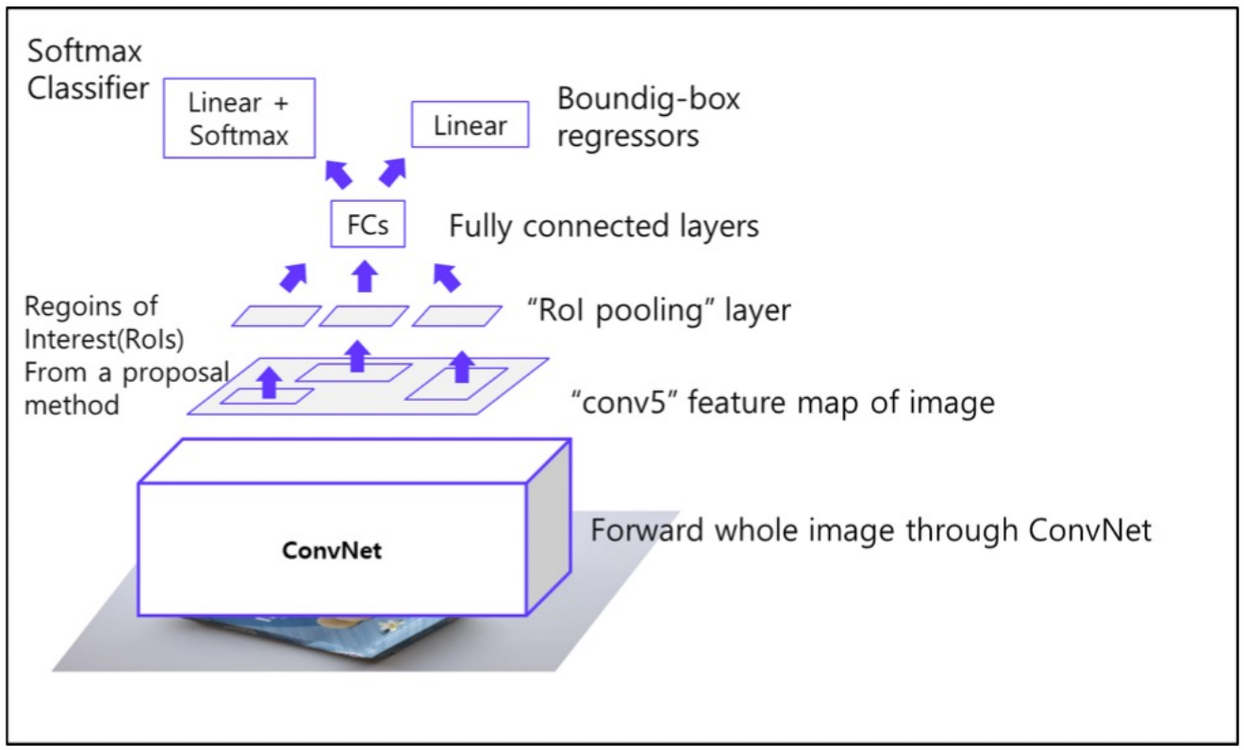
2 Stage Detector – SPPNet

- Shortcomings of R-CNN
  - ~~2000개의 region이 각각 CNN 통과~~
  - ~~강제 wraping으로 인한 성능 하락 가능성~~
  - CNN, SVM classifier, bounding box regressor 따로 학습
  - End-to-End 모델이 아님

2 Stage Detector – Fast R-CNN

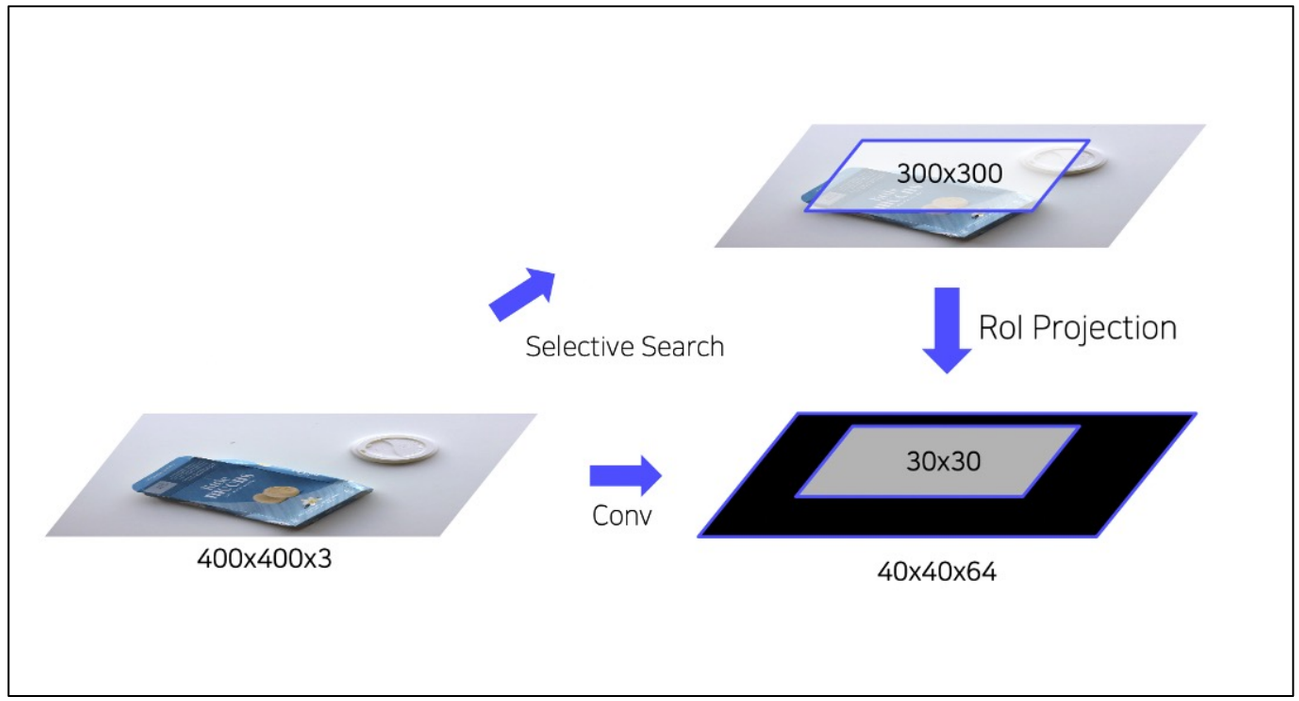
- Fast R-CNN
  - Spatial Pyramid Pooling과 유사한 ROI Pooling 이용하여 ROI의 feature vector 추출

- Pipeline
  - Step 1: 이미지 전체를 CNN에 통과 시켜 feature map 추출
    - VGG16 사용
  - Step 2: RoI Projection을 통해 feature map 상에서 RoI를 계산
  - Step 3: RoI Pooling을 통해 일정한 크기의 feature 추출
    - SPP 사용 (단, 타겟 사이즈가 1개만 존재)
  - Step 4: FC layer 통과 후 softmax classifier와 bbox regressor



2 Stage Detector – Fast R-CNN

- RoI Projection
  - 원본 이미지가 아닌 feature map에서 Selective Search를 직접적으로 사용할 수 없음
  - RoI Projection은 두 절차를 걸쳐 이를 극복
    - 원본 이미지에서 selective search를 통해 2000개의 RoI를 뽑음
    - CNN에서 feature map을 뽑고, selective search에서 뽑아낸 RoI를 그 위에 투사
      - 사이즈 조정 필요시 비율에 맞춰 조정 후 투사



2 Stage Detector – Fast R-CNN

- Training
  - Multi task loss 사용
    - Classification loss + bounding box regression
    - Loss function
      - Classification: Cross Entropy
      - BB regressor: Smooth L1
  - Dataset
    - IoU > 0.5: positive samples
    - 0.1 < IoU < 0.5: negative samples
    - Positive 25%, Negative 75%
  - Hierarchical sampling
    - R-CNN의 경우 이미지에 존재하는 RoI를 전부 저장해 사용
      - 한 배치에 서로 다른 이미지의 RoI가 포함
    - Fast R-CNN의 경우 한 배치에 한 이미지의 RoI만을 포함
      - 한 배치 안에서 연산과 메모리를 공유 가능

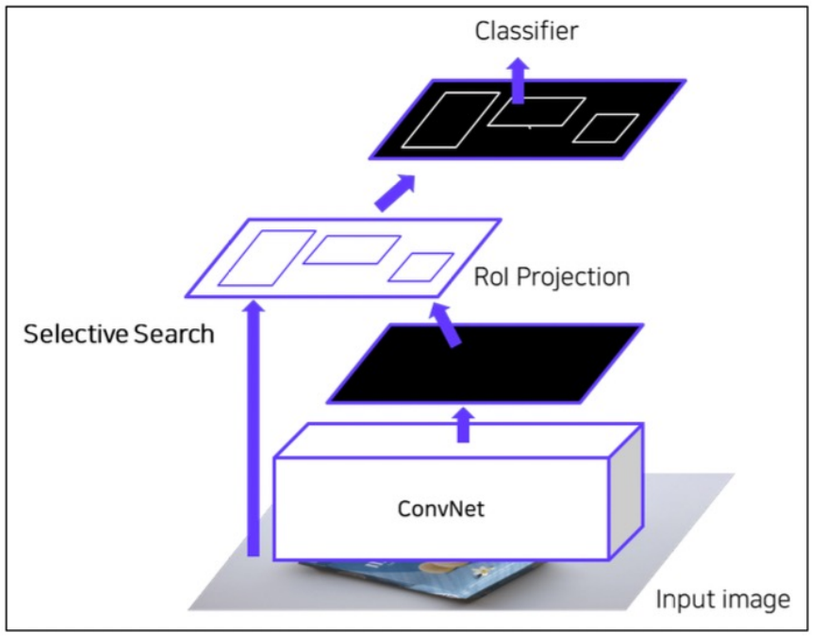
2 Stage Detector – Fast R-CNN

- Shortcomings of R-CNN
  - ~~2000개의 region이 각각 CNN 통과~~
  - ~~강제 wraping으로 인한 성능 하락 가능성~~
  - ~~CNN, SVM classifier, bounding box regressor 따로 학습~~
  - End-to-End 모델이 아님

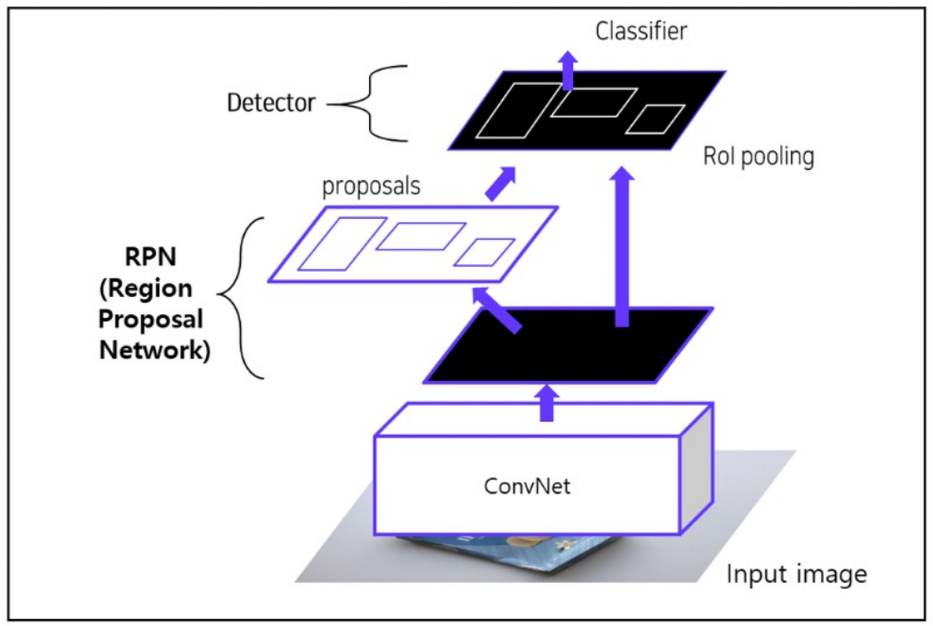
2 Stage Detector – Faster R-CNN

- Faster R-CNN
  - RPN (Region Proposal Network)를 이용해 Selective Search 대체
  - 완전한 뉴럴넷 기반 End-to-End 모델

- Pipeline
  - Step 1: 이미지를 CNN에 넣어 feature maps 추출
  - Step 2: RPN (Region Proposal Network)를 이용해 RoI 계산
    - Anchor Box 개념 사용



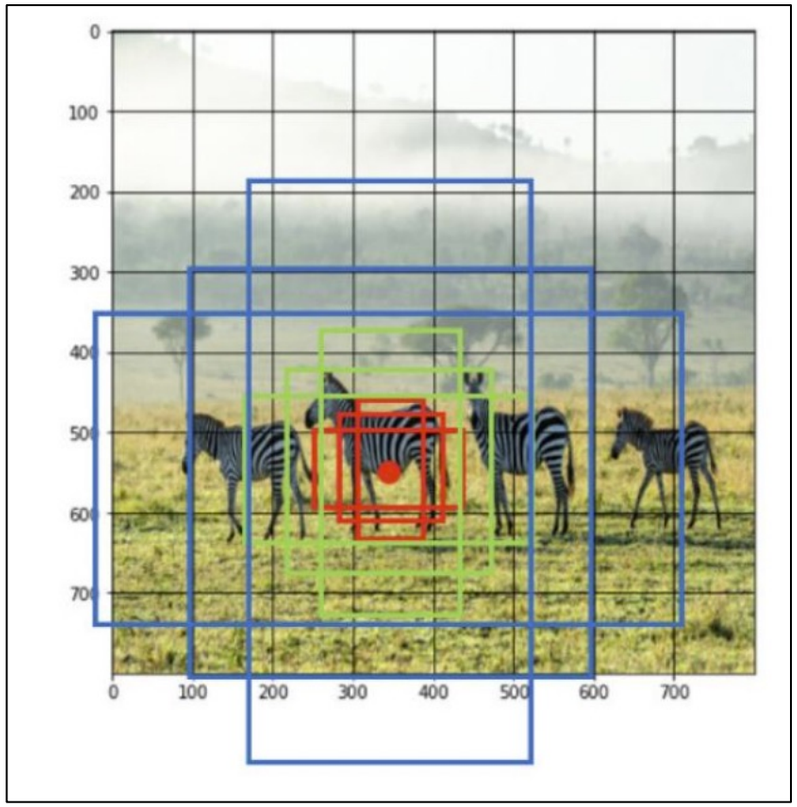
Fast R-CNN



Faster R-CNN

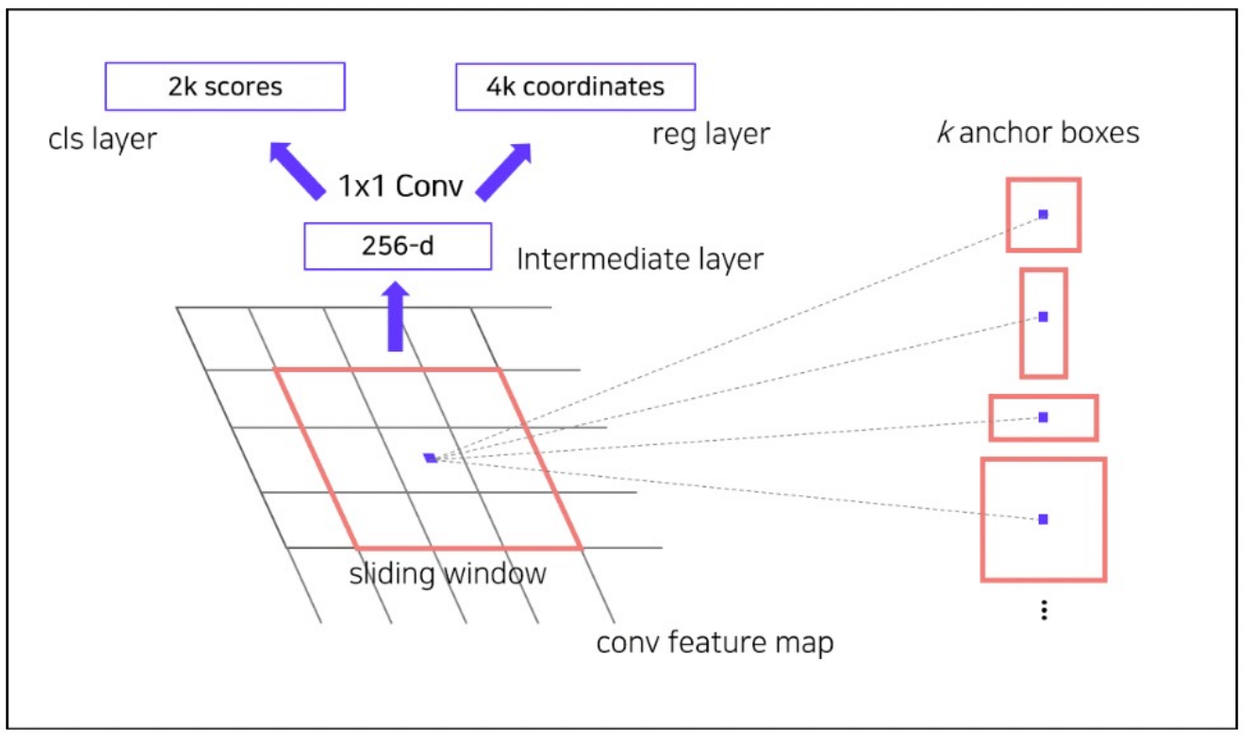
2 Stage Detector – Faster R-CNN

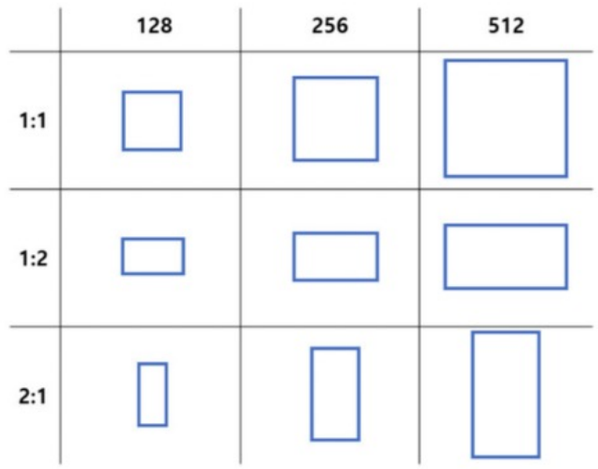
- Anchor box
  - 중심점은 같되 width와 height의 비율을 각기 다르게 하여 만든 box
  - 이미지를 셀로 나누고, 각 셀마다 크기가 다른 n개의 box를 미리 정의해두는 방식 사용



2 Stage Detector – Faster R-CNN

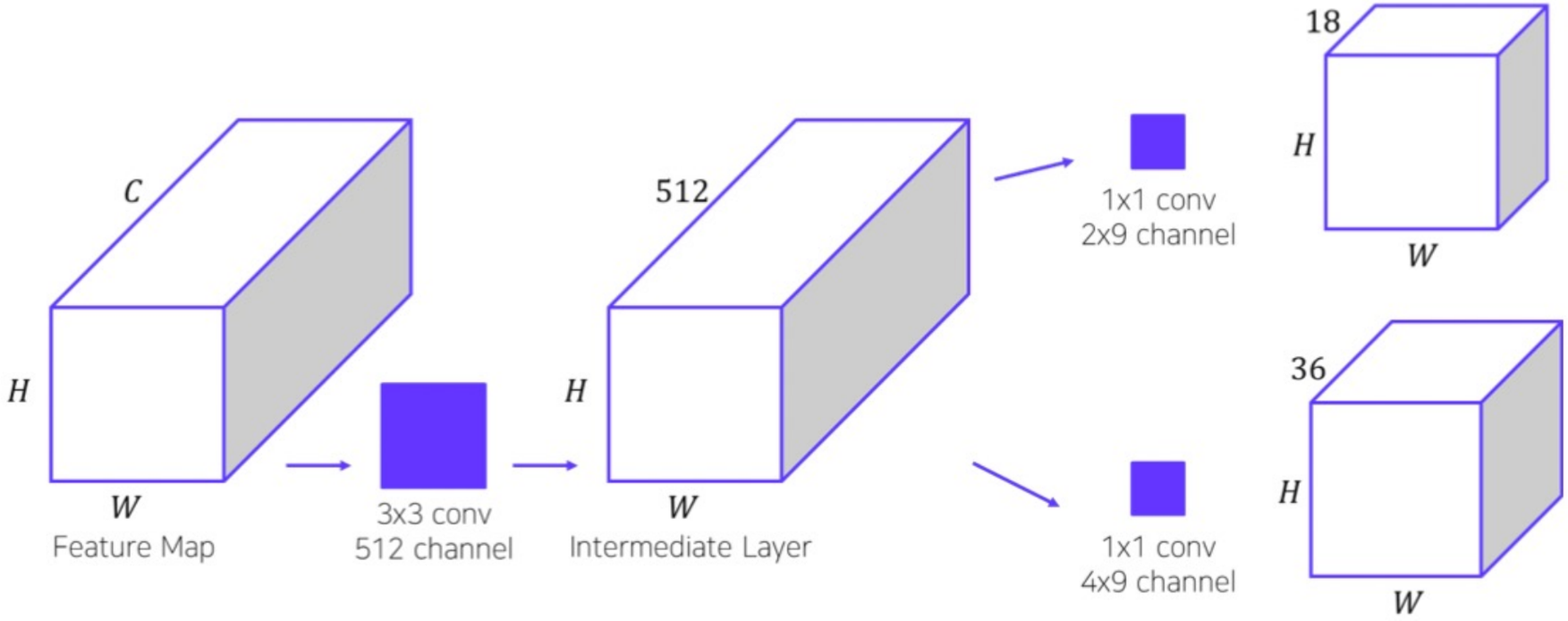
- Region Proposal Network (RPN)
  - 모들 셀에 대해서 n개의 anchor box를 적용하고, 모든 box들에 대해서 연산을 실행하는 것은 엄청난 비효율
  - RPN은 anchor box가 객체를 포함하는지를 판단
  - 만약 객체를 포함한다면 box의 크기를 미세하게 조정하는 작업 진행





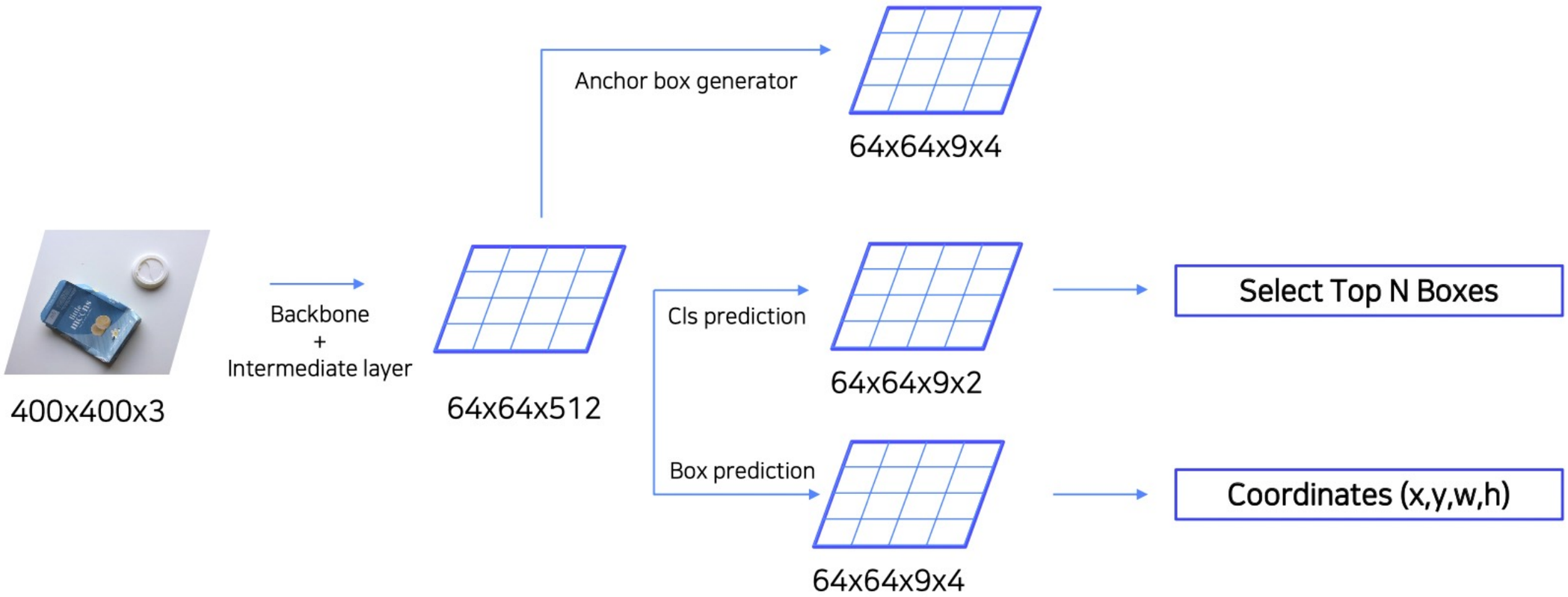
2 Stage Detector – Faster R-CNN

- Region Proposal Network (RPN) 상세
  - CNN에서 나온 feature map을 input으로 받음
  - 3x3 conv을 수행하여 intermediate layer 생성
  - 1x1 conv를 수행하여
    - binary classification 수행
      - 2 (object or not) x 9 (num of anchors) 채널 생성
    - Bounding box regression 수행
      - 4 (bounding box) x 9 (num of anchors) 채널 생성



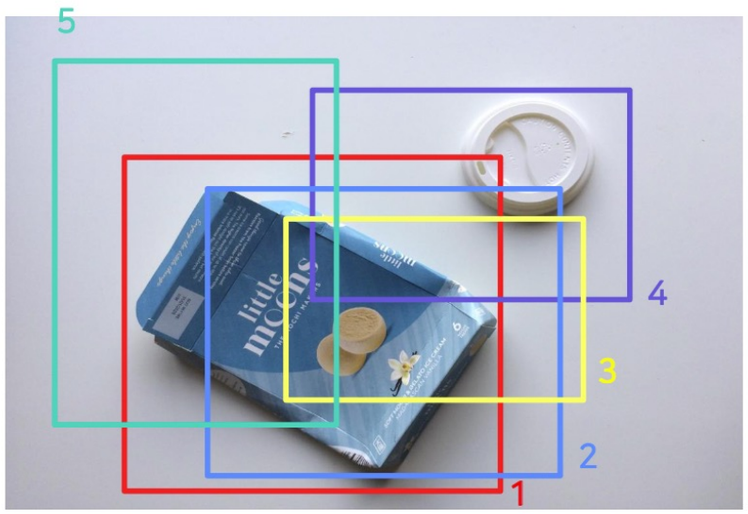
2 Stage Detector – Faster R-CNN

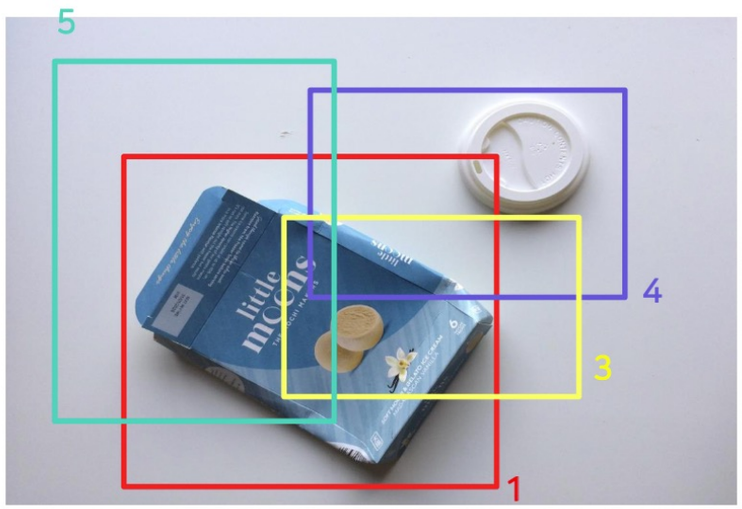
❑ Region Proposal Network (RPN) 상세



2 Stage Detector – Faster R-CNN

- Region Proposal Network (RPN) 상세
  - NMS
    - 유사한 RPN Proposals 제거하기 위해 사용
    - Class score를 기준으로 proposal 분류
    - IoU가 0.7 이상인 proposals 영역들은 중복된 영역으로 판단한 뒤 제거





Class score for each box

bb 1	bb 2	bb 5	bb 3	bb 4
0.9	0.8	0.6	0.4	0

IoU score about bb1

bb 1	bb 2	bb 5	bb 3	bb 4
1.0	0.8	0.6	0.4	0.05

Class score for each box

bb 1	bb 2	bb 5	bb 3	bb 4
0.9	0	0.6	0.4	0

2 Stage Detector – Faster R-CNN

- ❑ Region Proposal Network (RPN) Traning
  - ■ RPN 단계에서 classification과 regressor학습을 위해 앵커박스를 positive/negative samples 구분
  - ■ 데이터셋 구성
    - ▪ IoU > 0.7 or highest IoU with GT: positive samples
    - ▪ IoU < 0.3: negative samples
    - ▪ Otherwise : 학습데이터로 사용 X
  - ■ Loss

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

  - ■ Region proposal 이후 Fast RCNN 학습을 위해 positive/negative samples로 구분
  - ■ 데이터셋 구성
    - ▪ IoU > 0.5: positive samples → 32개
    - ▪ IoU < 0.5: negative samples → 96개
    - ▪ 128개의 samples로 mini-bath 구성
  - ■ RPN과 Fast R-CNN 학습을 위해 4 steps alternative training 활용
  - ■ 학습 과정이 매우 복잡해서, 최근에는 Approximate Joint Training 활용

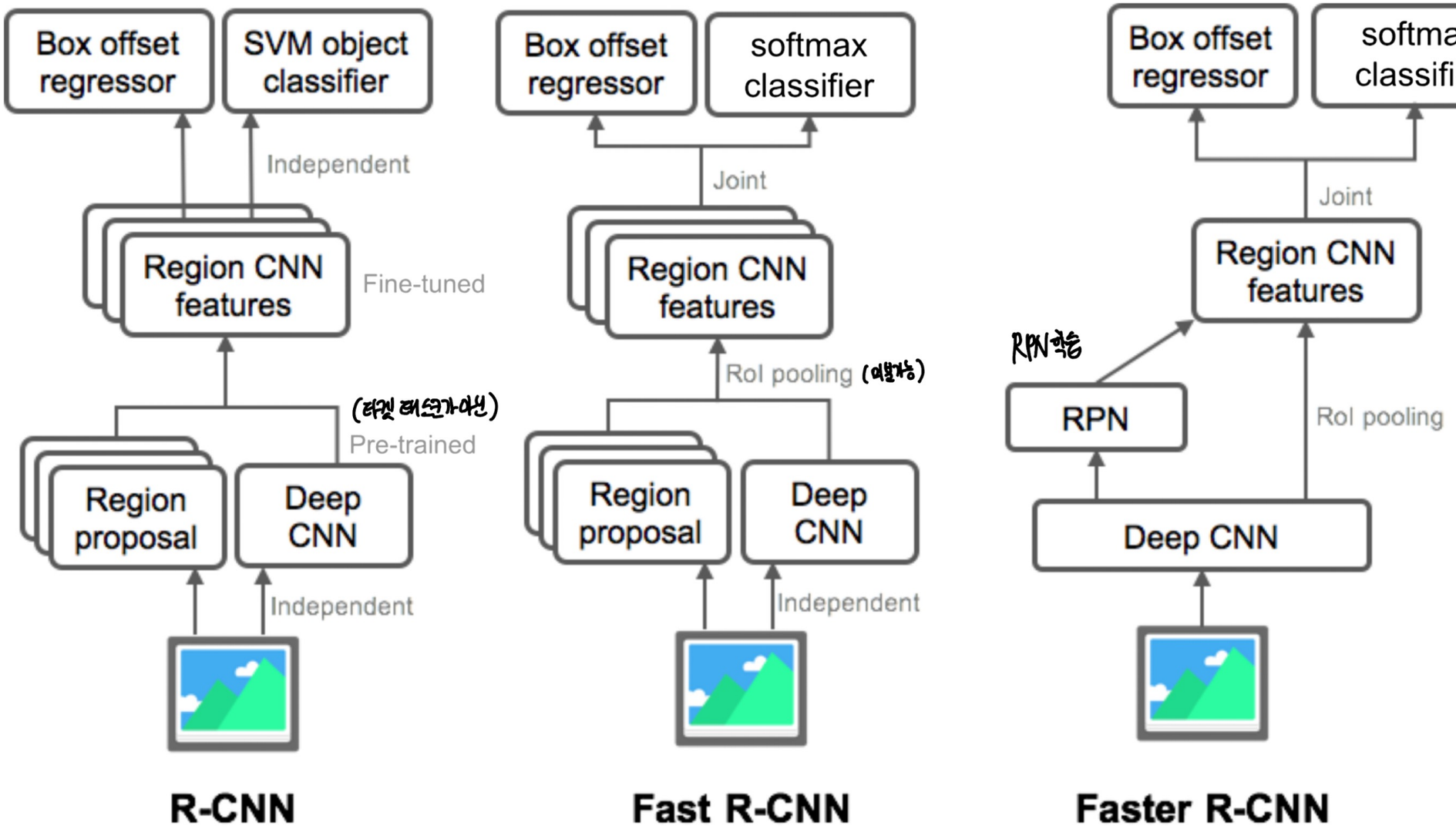
2 Stage Detector – Faster R-CNN

- Results

method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared†	300	12	67.0
RPN+VGG, shared‡	300	07++12	**70.4**

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**

2 Stage Detector – Summary



2 Stage Detector – Summary

	R-CNN	Fast R-CNN	Faster R-CNN
Classification	SVMs	Linear	Linear
Resize	Warp	RoI Pooling	RoI pooling
Region Proposal	Selective Search	Selective Search	Region Proposal Network(RPN)
End-to-end	X	X	O

Thank you