Faster R-CNN

: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Research Purpose

1. 기존의 객체 탐지 모델은 Region Proposal의 추출에 **Selective Search** 와 같은 방식을 사용하여 연산량과 속도 면에서 한계가 있었음.

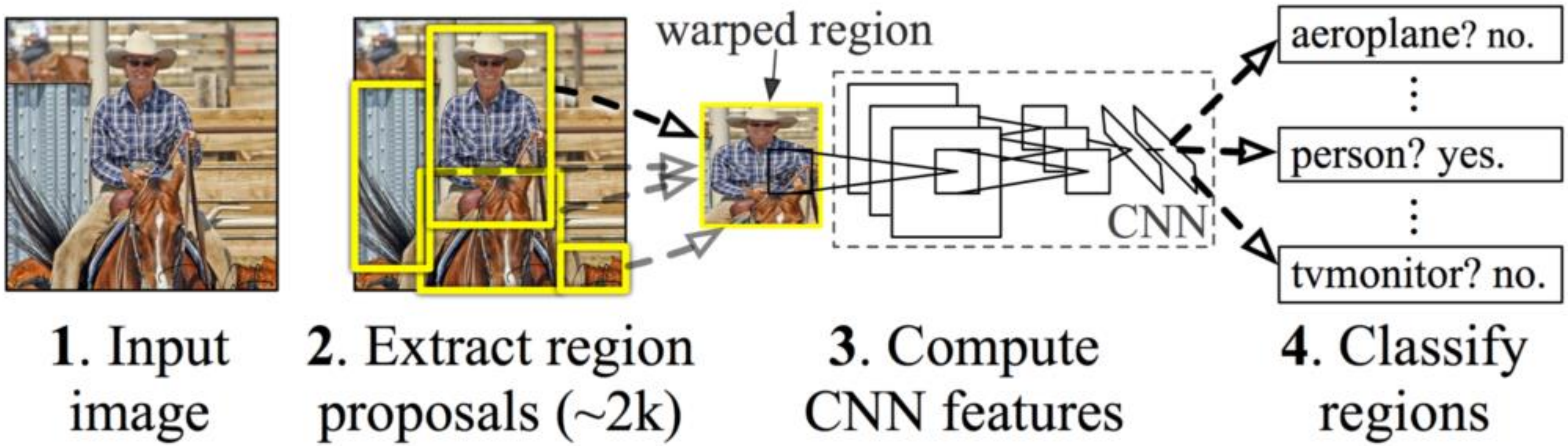
2. Faster R-CNN은 **Region Proposal Network(RPN)**를 도입해, Region Proposal 영역을 CNN 기반으로 학습하여 생성함으로써 연산 속도를 개선함.

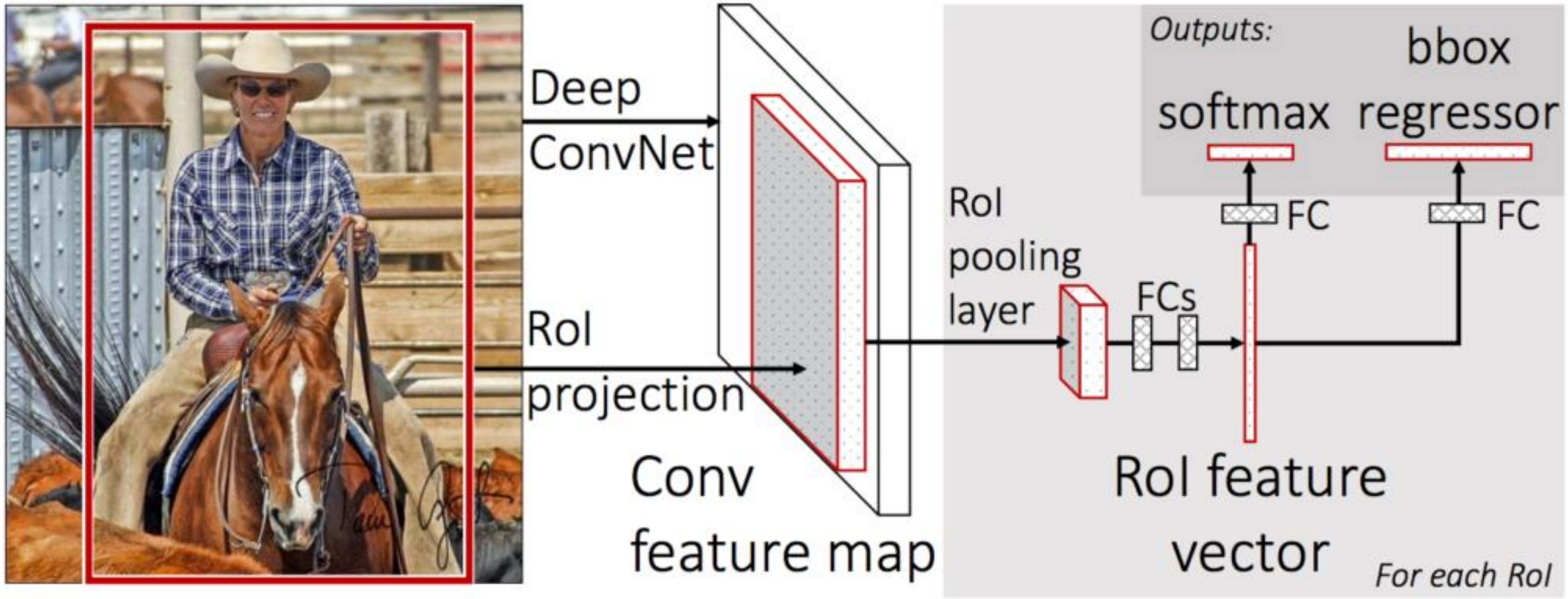
3. RPN을 통해 Region Proposal 생성과 Feature Extraction을 통합하며, 같은 Convolution Layer를 공유해 연산량을 줄이고 효율성을 높임.

RPN (Region Proposal Network)

- R-CNN과 Fast R-CNN이 CPU 상에서 작동하는 Selective Search algorithm 을 이용해 region proposal을 찾아내던 것을 GPU로 옮겨 proposal을 추론하는 방법.
- Transformer 이전 model 임.

Convolutional feature를 공유하여 효율적인 네트워크 구조를 구성함.


warped region
aeroplane? no.
person? yes.
tvmonitor? no.
CNN
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions


Deep ConvNet
RoI projection
Conv feature map
RoI pooling layer
FCs
Outputs:
softmax
bbox regressor
FC
FC
RoI feature vector
For each RoI

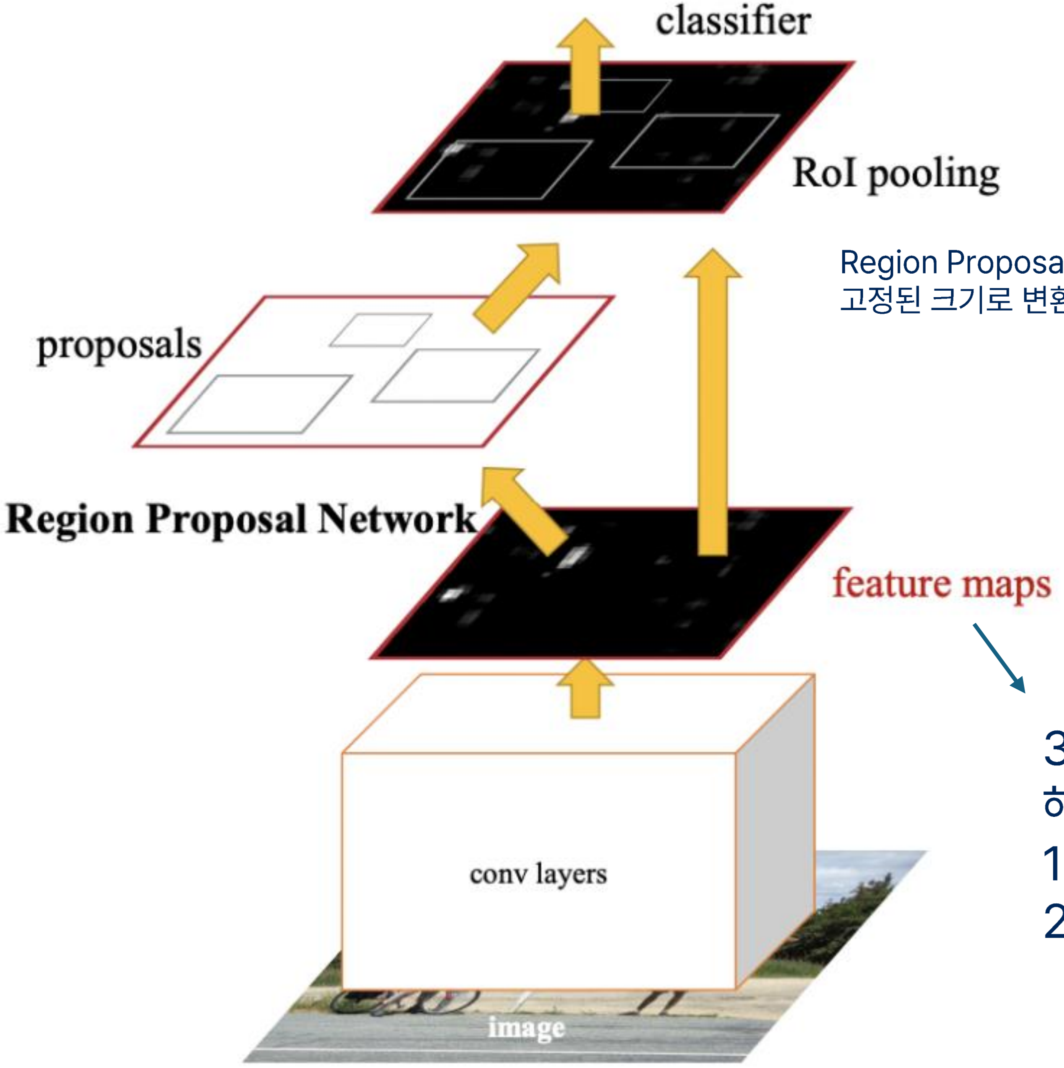
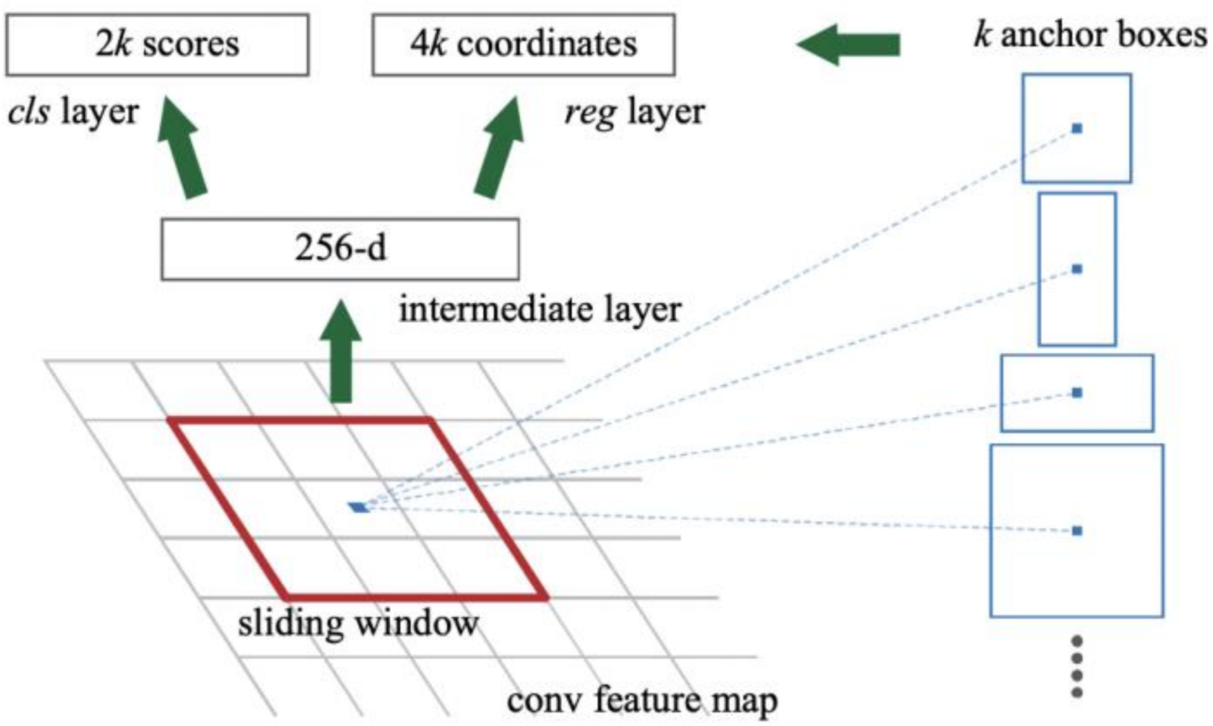


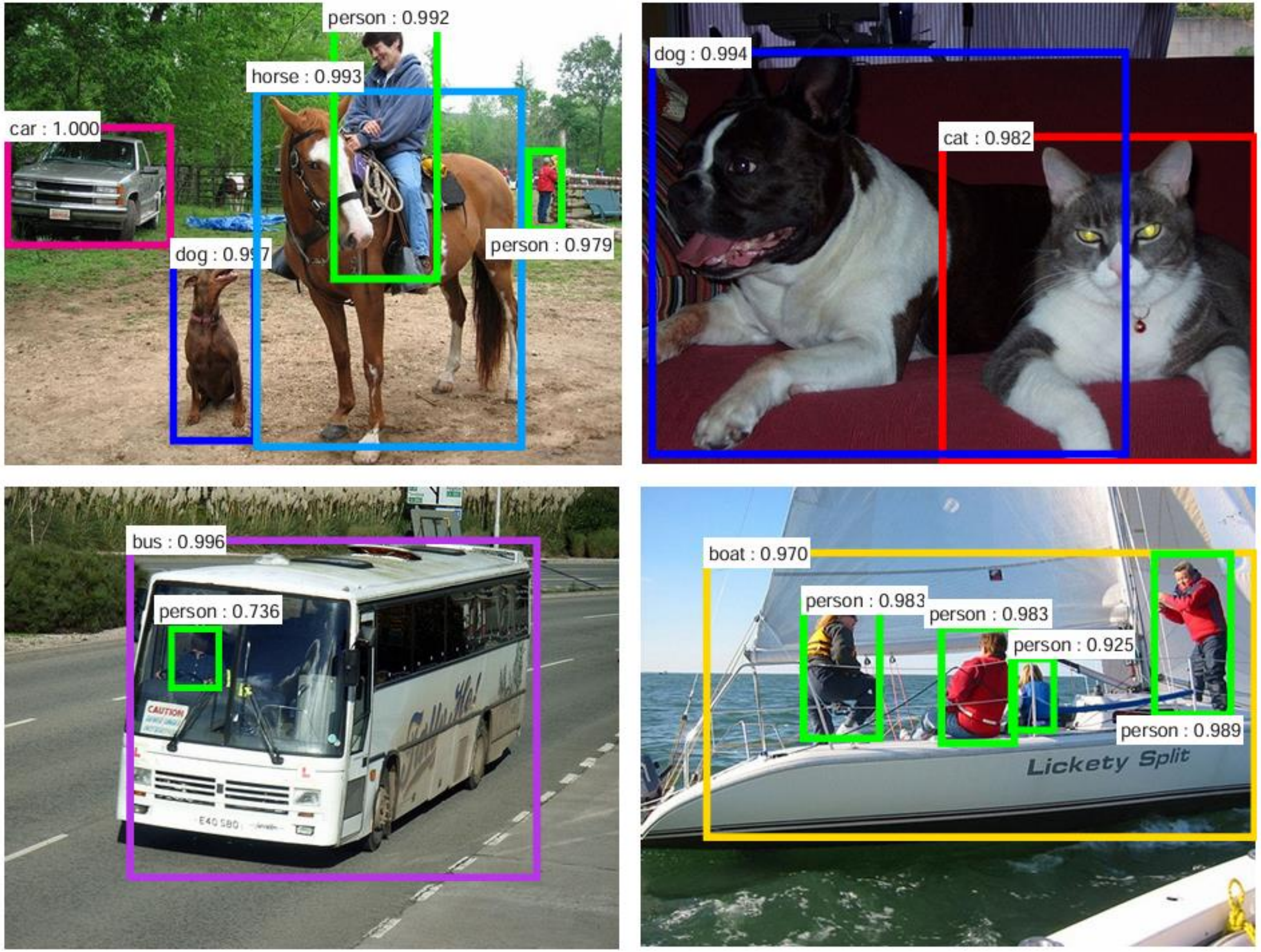
Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

Region Proposal을
고정된 크기로 변환.



3x3 필터를 적용해 각 위치의 주변 정보 요약
해당 결과를 바탕으로

1. Classification (Objectness Score) – 객체 O/X
2. Bounding Box Regression – Anchor Box 기반, 크기와 위치 조정



Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios

Loss Function

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

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a,$$
$$t_w = \log(w/w_a), \quad t_h = \log(h/h_a),$$
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a,$$
$$t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a),$$

RPN의 Loss Function
Pi : i번째 Anchor Box가 객체일 확률. (0~1사이)
Pi* : 1 객체 / 0 배경

ti : 모델이 예측한 Bounding Box의 좌표 조정값
ti * : 계산된 실제 조정값.

Bounding Box Regression

Experiment

train-time region proposals		test-time region proposals		
method	# boxes	method	# proposals	mAP (%)
SS	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	**59.9**
ablation experiments follow below				
RPN+ZF, unshared	2000	RPN+ZF, unshared	300	58.7
SS	2000	RPN+ZF	100	55.1
SS	2000	RPN+ZF	300	56.8
SS	2000	RPN+ZF	1000	56.3
SS	2000	RPN+ZF (no NMS)	6000	55.2
SS	2000	RPN+ZF (no *cls*)	100	44.6
SS	2000	RPN+ZF (no *cls*)	300	51.4
SS	2000	RPN+ZF (no *cls*)	1000	55.8
SS	2000	RPN+ZF (no *reg*)	300	52.1
SS	2000	RPN+ZF (no *reg*)	1000	51.3
SS	2000	RPN+VGG	300	59.2

Experiment

train-time region proposals		test-time region proposals		
method	# boxes	method	# proposals	mAP (%)
SS	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	**59.9**
ablation experiments follow below				
RPN+ZF, unshared	2000	RPN+ZF, unshared	300	58.7
SS	2000	RPN+ZF	100	55.1
SS	2000	RPN+ZF	300	56.8
SS	2000	RPN+ZF	1000	56.3
SS	2000	RPN+ZF (no NMS)	6000	55.2
SS	2000	RPN+ZF (no *cls*)	100	44.6
SS	2000	RPN+ZF (no *cls*)	300	51.4
SS	2000	RPN+ZF (no *cls*)	1000	55.8
SS	2000	RPN+ZF (no *reg*)	300	52.1
SS	2000	RPN+ZF (no *reg*)	1000	51.3
SS	2000	RPN+VGG	300	59.2

Experiment

train-time region proposals		test-time region proposals		
method	# boxes	method	# proposals	mAP (%)
SS	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	**59.9**
ablation experiments follow below				
RPN+ZF, unshared	2000	RPN+ZF, unshared	300	58.7
SS	2000	RPN+ZF	100	55.1
SS	2000	RPN+ZF	300	56.8
SS	2000	RPN+ZF	1000	56.3
SS	2000	RPN+ZF (no NMS)	6000	55.2
SS	2000	RPN+ZF (no *cls*)	100	44.6
SS	2000	RPN+ZF (no *cls*)	300	51.4
SS	2000	RPN+ZF (no *cls*)	1000	55.8
SS	2000	RPN+ZF (no *reg*)	300	52.1
SS	2000	RPN+ZF (no *reg*)	1000	51.3
SS	2000	RPN+VGG	300	59.2

method	# proposals	data	mAP (%)
SS	2000	07	66.9†
SS	2000	07+12	70.0
RPN+VGG, unshared	300	07	68.5
RPN+VGG, shared	300	07	69.9
RPN+VGG, shared	300	07+12	73.2
RPN+VGG, shared	300	COCO+07+12	78.8

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
ZF	RPN + Fast R-CNN	31	3	25	59	17 fps

Conclusion

- Faster R-CNN은 RPN을 도입하여 기존 SS 기반 방법보다 높은 성능과 효율성을 달성함.
- RPN은 Feature Map을 공유하고 학습 기반 Region Proposal 방식을 사용하여 객체 탐지의 정확도를 향상시킴.
- Faster R-CNN은 여전히 연산 문제가 발생하므로, Transformer 기반 객체 탐지 모델 (DERT 등) 을 통해 CNN 기반 한계를 보완할 수 있음.

감사합니다.