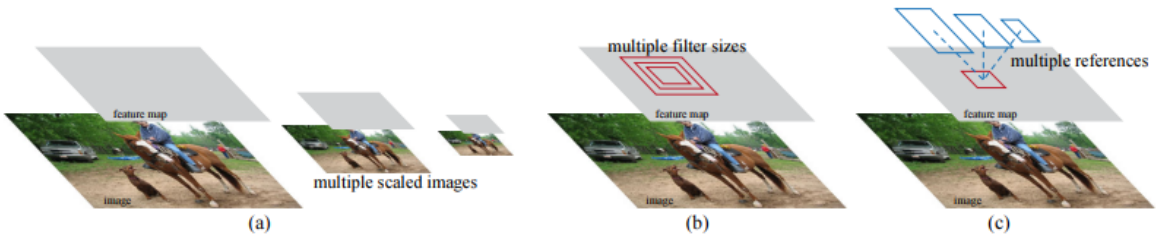
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Abstract & Introduction

- 기존의 object detection network(SPPnet, Fast R-CNN)는 객체의 위치를 가정하기 위해 영역 제안 알고리즘(Region-based Convolutional Neural Networks, R-CNNs) 에 의존함 → 제안간 합성곱을 공유하게 되며 시간과 비용이 줄었지만, test-time에서 계산 병목 문제
    - Selective Search나 EdgeBoxes는 CPU에서 느리게 작동함. 하지만 CNN은 GPU를 활용



(a), (b) 기존의 방법들로 (a)는 이미지와 feature map의 pyramid를 만들고 모든 scale에서 분류기를 실행함. (b)는 다양한 scale filter pyramid를 feature map 위에서 실행함. (c)는 본 논문에서 제안된 방법으로 regression function에서 anchor box의 pyramid를 사용함

- 본 논문에서는 detectin network와 전체 이미지에 대한 합성곱의 특성을 공유하는 Region Proposal Network(Fully convolutional network로 grid 상 각 위체에서 객체의 경계와 객체성(objectiveness) 점수를 동시에 예측(regression). 이때 test 시점에서 합성곱을 공유함으로써 제안을 계산하는데 드는 추가 비용이 매우 작아짐)를 제안함
    - 이때 anchor box를 도입하여 폭넓은 크기와 aspect ratio(종횡비)를 가진 영역 제안을 효율적으로 예측하도록 설계됨.
- RPN + Fast R-CNN 객체 탐지 네트워크를 통합하기 위해, proposal 과정 fine-tuning과 object detection 에서 fine-tuning을 번갈아가면서 수행하는 training

scheme을 제안함. → 두 task의 합성곱의 특성을 공유하는 통합된 네트워크를 만들어 냄 + 빠르게 수렴이 가능

- RPN+Fast R-CNN의 경우 PASCAL VOC 탐지 벤치마크에서 Selective Search+Fast R-CNN 보다 더 나은 정확도를 달성함. Pinterest 같은 상업 서비스에도 도입되어 실제 활용됨.
- test 시간에서 걸리는 proposal를 생성하는 실제 시간은 10ms 정도

Related Work

Object Proposal

- 널리 사용되는 방식은 super-pixel 초픽셀을 그룹화 하는 방식 (Selective Search 등), 또는 sliding window 기반 방식 (window 내 objectiveness, EdgeBoxes 등) 이 있음 → 이러한 객체 제안 방법은 detector와 독립적인 외부 모듈로 사용됨 (Selective Search 기반 객체 탐지기, R-CNNs 등)

Deep Network

- R-CNN 방법은 CNN을 end-to-end로 학습시켜, proposal region 제안된 영역을 객체 클래스 또는 배경으로 분류함. 이때 RNN은 주로 분류기 역할을 하여, bounding box regression으로 보정은 하지만, **객체의 경계를 직접 예측**하지는 않음.
- 이전 여러 논문에서, 객체의 bounding box를 예측하기 위해 deep network를 사용하는 방법을 제안해옴.
  - OverFeat: fully-connected layer를 학습시켜 단일한 객체의 위치를 예측하게 함
  - MultiBox: 네트워크의 마지막 fully-connected layer에서 클래스와 관한 박스 여러개를 동시에 예측하여 region proposal 을 생성함 → R-CNN에서 사용됨. 이는 단일 이미지나 여러개의 큰 이미지를 crop 하는데 적용되고, fully convolutional 은 아님. 또한 proposal network와 detection network에서 특징을 공유하지 않음
  - DeepMask: 본 논문에서와 동시에 segmentation 제안을 학습하기 위해 제안됨
- 합성곱의 연산을 공유하여 효율적이면서도 정확도를 높이는 연구가 점점 더 많은 관심을 받고 있음.
  - OverFeat: 분류, 위치 추정, 탐지를 위해 이미지 피라미드에서 합성곱의 특징을 계산함
  - Spatial Pyramid Pooling: 공유된 합성곱 feature map 위에서 크기에 맞게 조정이 가능한 adaptively-sized pooling을 수행함.

    - 위의 딥러닝 기반 방법들은 합성곱을 공유하지 않음. Fast R-CNN은 공유된 합성곱의 특징 위에 end-to-end object detection 학습을 가능하게 함

Method

- object detection Faster R-CNN = region proposal 을 위한 deep fully convolutional network + 제안된 영역을 사용하는 Fast R-CNN detector → 전체 시스템은 객체 탐지를 위한 단일 통합된 네트워크임 (RPN은 Fast R-CNN에게 어디를 봐야할지를 알려주는 역할)

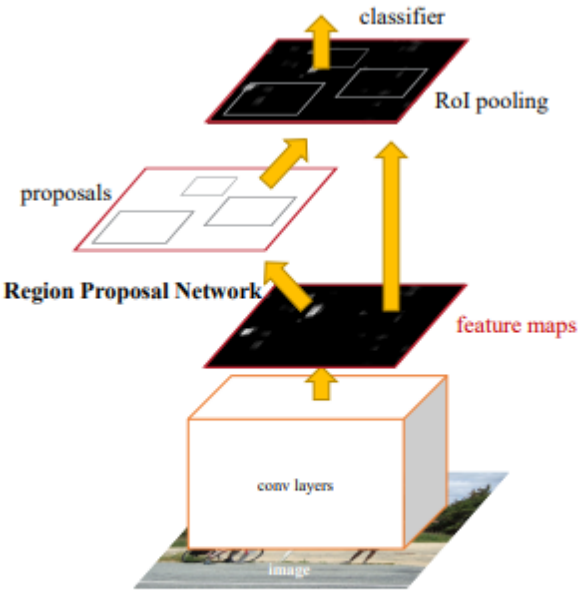


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

Region Proposal Network

- RPN은 크기와 상관없이 이미지를 입력으로 받아, 각각 objectness score가 있는 사각형 영역 제안을 출력함. 이 과정을 fully convolutional network로 모델링함. 이때 최종 목표는 RPN이 detector network인 Fast R-CNN과 계산을 공유하는 것이므로 두 네트워크가 공통의 합성곱 계층을 가진다고 가정함
- 본 논문에서 실험을 위해 사용된 2가지 모델은 ZF model (공유 가능한 합성곱 계층이 5개), VGG-16 model (공유 가능한 합성곱 계층이 13개)
- 영역 제안을 생성하기 위해, 공유된 마지막 합성곱 계층의 output feature map 위에 작은 네트워크를 sliding 시킴 → 이 네트워크는 nxn 차원의 window를 입력으로 받음 + 각 sliding window는 저차원의 특징으로 매핑됨, 이 feature는 두개의 (box regression/box classification) fully connected layer에 전달됨

→ 따라서 nxn convolution layer + 1×1 convolutional layer (회귀, 분류 각각)

→ 이 네트워크는 sliding window 방식으로 작동하기 때문에, fully connected layer는 모든 공간 위치에서 공유됨

- 본 논문에서는, n=3을 이용하여 입력 이미지에서 유효 수용 영역은 ZF의 경우 171픽셀, VGG의 경우 228 픽셀임.

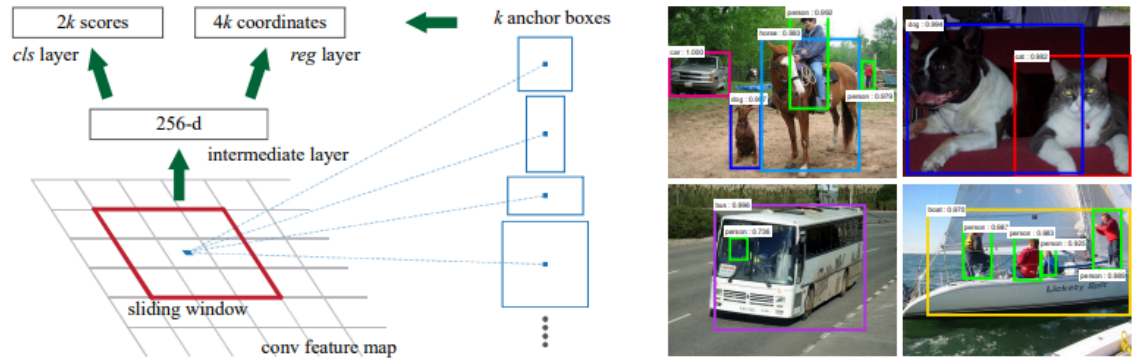


Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

- 각 sliding window 위치마다, 동시에 여러개의 영역 제안을 예측함 → 위치 당 생성 가능한 최대 proposal의 개수를 k 개라고 할 때, reg 는 k개의 박스 좌표를 enocoding 하는 4k개 출력, 그리고 cls는 각 제안에 대한 객체일 확률/아닌 확률을 나타내는 2k 개의 점수 출력
- 이 k 개의 제안은 k개의 anchor boxes를 기준으로 파라미터화됨. Anchor box는 sliding window의 중심에 위치하여, scale과 aspect ratio를 가짐.
  - 기본 설정으론 3개의 ratio,와 3개의 aspect ratio → k=9개의 anchor가 생성됨
- 이때 중요한 것은 translation invariance (변환 불변성)으로 앵커나 proposal의 위치를 계산하는 함수들이 이미지 이동에 영향을 받지 않는다는 것
  - 예를 들어, 이미지 안의 객체를 이동시켰을 때, 제안 위치도 동일하게 이동해서 네트워크는 새 위치에서도 동일한 제안을 예측할 수 있어야 함
  - MultiBox는 800개의 앵커를 k-means로 생성하고 이는 불변성 보장 x
  - 또한 이는 모델 크기를 감소시키는데, MultiBox output layer (4 + 1) × 800 fullly connected layer인 반면, Faster R-CNN output layer는 (4 + 2) × 9 convolutional layer가 됨
  - 모델 파라미터 수도, MultiBox는 610만인 반면, Faster R-CNN은 2.8만 → 과적합 가능성이 낮기에 적은 데이터셋에도 유리해짐

- 기존의 Anchor design은 2가지 방식: 1) 이미지 feauture map 피라미드를 만들어, 모든 스케일에서 detection (계산 비용 문제 있음) 2) 슬라이딩 윈도우에서 다양한 크기의 필터 사용 (=pyramid of filters)
- 본 논문의 Anchor design은 pyramid of anchors 로 단일 필터를 슬라이딩하면서 여러 scale와 aspect ratio를 갖는 앵커 박스들을 기준으로 reg+cls 수행 → 추가적인 계산 없이 공유된 feature map을 통해 다양한 크기를 처리할 수 있게 해줌
- RPN cls에선 positive label를 갖는다는건, 앵커가 ground-truth box와 가장 높은 IoU를 갖거나 (positive label이 없을 수도 있기 때문에) 또는 앵커의 IoU가 0.7이상이라는 것 → 따라서 하나의 ground-truth box가 여러 앵커에 positive label을 부여할 수 있음
    - 추가적으로 IoU가 0.3 이하면 negative label, 그리고 p/n이 둘다 아닌 앵커는 학습에서 제외됨

[Faster R-CNN의 multitask loss ftn]

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

이때 pi는 예측값(확률), pi*는 실제 label, ti는 예측된 bounding box 좌표값, ti*는 실제 bounding box 좌표값. Lamdba는 cls/reg 조정하는 가중치로 기본값은 10

- 따라서 reg loss는 positive label에 대해서만 계산됨

[Bounding Box Regression]

$$\begin{aligned} t_{\mathrm{x}} &= (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}} &= \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}), \\ t_{\mathrm{x}}^* &= (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}}^* = (y^* - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}}^* &= \log(w^*/w_{\mathrm{a}}), \quad t_{\mathrm{h}}^* = \log(h^*/h_{\mathrm{a}}), \end{aligned} \quad (2)$$

- 기존의 방식과 달리 고정된 크기의 필터를 사용하더라도 다양한 scale의 박스를 예측할 수 있게 됨

- RPN은 역전파와 SGD로 end-to-end 학습이 가능함. 이때 Fast R-CNN에서 사용된 **image-centric sampling** 방식을 사용
    - 각 mini-batch는 단일 이미지에서 생성되고 많은 pos/neg 앵커를 포함함. pos 비율이 낮을 때는 pos/neg ratio를 최대 1:1로 맞춤
- 이때 RPN은 proposal만 생성, Fast R-CNN은 해당 proposal로 객체를 탐지 → 따로 학습되면 합성곱 계층이 다르게 변형되어 layer간 공유가 어려움
- 따라서 두 네트워크를 따로 학습하는 대신 feature을 공유하면서 학습하는 3가지 방법을 제안
    a. Alternating training: 먼저 RPN을 학습하고, 이 proposal을 이용해 Fast R-CNN을 학습 + Fast R-CNN에서 학습된 네트워크는 다시 RPN 초기화에 사용 + 반복
    b. Approximate joint training: RPN과 Fast R-CNN을 하나의 네트워크로 병합. Optimizer의 한번의 iteration마다 RPN이 proposal 생성 → 이때 생성된 proposal은 고정된 것으로 간주하고 Fast R-CNN에서의 역전파는 RPN까지 도달하지 않음. (위보다 빠르지만 정확도 떨어짐)
    c. Non-approximate joint training: RPN에서 생성된 바운딩 박스는 Fast R-CNN에서 다시 사용되는데 이때 **RoI pooling laye**r가 사용됨
- 본 논문에서는 4-Step Alternating Training 알고리즘을 채택
    1. ImageNet 사전학습된 네트워크 사용해서 RPN 학습
    2. RPN의 proposal을 사용해서 Fast R-CNN detector 학습 ! 이때 두 네트워크는 합성곱 계층 공유 Xx
    3. Fast R-CNN detector로 RPN 재학습 (공유 layer 고정 + RPN layer만 학습)
    4. 공유 layer 고정 + Fast R-CNN layer만 학습 → 두 layer가 같은 합성곱 계층을 공유하게 됨
- 이때 RPN과 Fast R-CNN은 단일 스케일 이미지에서 학습됨 (긴 변이 600 pixel)

Table 1: the learned average proposal size for each anchor using the ZF net (numbers for $s = 600$).

anchor	128^2, 2:1	128^2, 1:1	128^2, 1:2	256^2, 2:1	256^2, 1:1	256^2, 1:2	512^2, 2:1	512^2, 1:1	512^2, 1:2
proposal	188×111	113×114	70×92	416×229	261×284	174×332	768×437	499×501	355×715

Table 2: Detection results on **PASCAL VOC 2007 test set** (trained on VOC 2007 trainval). The detectors are Fast R-CNN with ZF, but using various proposal methods for training and testing.

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

Experiments & Conclusion

Table 3: Detection results on **PASCAL VOC 2007 test set**. The detector is Fast R-CNN and VGG-16. Training data: "07": VOC 2007 trainval, "07+12": union set of VOC 2007 trainval and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. †: this number was reported in [2]; using the repository provided by this paper, this result is higher (68.1).

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**

Table 4: Detection results on **PASCAL VOC 2012 test set**. The detector is Fast R-CNN and VGG-16. Training data: "07": VOC 2007 trainval, "07++12": union set of VOC 2007 trainval+test and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. †: http://host.robots.ox.ac.uk:8080/anonymous/HZJTQA.html. ‡: http://host.robots.ox.ac.uk:8080/anonymous/YNPLXB.html. §: http://host.robots.ox.ac.uk:8080/anonymous/XEDH10.html.

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**
RPN+VGG, shared§	300	COCO+07++12	**75.9**

Table 5: **Timing** (ms) on a K40 GPU, except SS proposal is evaluated in a CPU. "Region-wise" includes NMS, pooling, fully-connected, and softmax layers. See our released code for the profiling of running time.

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**

Table 6: Results on PASCAL VOC 2007 test set with Fast R-CNN detectors and VGG-16. For RPN, the train-time proposals for Fast R-CNN are 2000. RPN* denotes the unsharing feature version.

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
SS	2000	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
RPN*	300	07	68.5	74.1	77.2	67.7	53.9	51.0	75.1	79.2	78.9	50.7	78.0	61.1	79.1	81.9	72.2	75.9	37.2	71.4	62.5	77.4	66.4
RPN	300	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
RPN	300	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
RPN	300	COCO+07+12	**78.8**	**84.3**	**82.0**	**77.7**	**68.9**	**65.7**	**88.1**	**88.4**	**88.9**	**63.6**	**86.3**	**70.8**	**85.9**	**87.6**	**80.1**	**82.3**	**53.6**	**80.4**	**75.8**	**86.6**	**78.9**

Table 7: Results on PASCAL VOC 2012 test set with Fast R-CNN detectors and VGG-16. For RPN, the train-time proposals for Fast R-CNN are 2000.

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
SS	2000	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	**65.7**	80.4	64.2
RPN	300	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
RPN	300	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
RPN	300	COCO+07++12	**75.9**	**87.4**	**83.6**	**76.8**	**62.9**	**59.6**	**81.9**	**82.0**	**91.3**	**54.9**	**82.6**	**59.0**	**89.0**	**85.5**	**84.7**	**84.1**	**52.2**	**78.9**	65.5	**85.4**	**70.2**

Table 8: Detection results of Faster R-CNN on PASCAL VOC 2007 test set using **different settings of anchors**. The network is VGG-16. The training data is VOC 2007 trainval. The default setting of using 3 scales and 3 aspect ratios (69.9%) is the same as that in Table 3.

settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128^2	1:1	65.8
	256^2	1:1	66.7
1 scale, 3 ratios	128^2	{2:1, 1:1, 1:2}	68.8
	256^2	{2:1, 1:1, 1:2}	67.9
3 scales, 1 ratio	$\{128^2, 256^2, 512^2\}$	1:1	**69.8**
3 scales, 3 ratios	$\{128^2, 256^2, 512^2\}$	{2:1, 1:1, 1:2}	**69.9**

Table 9: Detection results of Faster R-CNN on PASCAL VOC 2007 test set using **different values of** λ in Equation (1). The network is VGG-16. The training data is VOC 2007 trainval. The default setting of using $\lambda = 10$ (69.9%) is the same as that in Table 3.

λ	0.1	1	10	100
mAP (%)	67.2	68.9	69.9	69.1

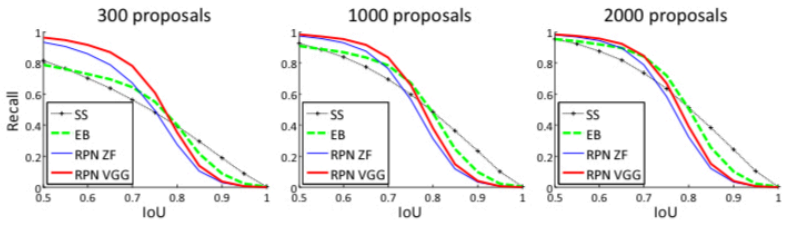


Figure 4: Recall *vs.* IoU overlap ratio on the PASCAL VOC 2007 test set.

Table 10: **One-Stage Detection *vs.* Two-Stage Proposal + Detection**. Detection results are on the PASCAL VOC 2007 test set using the ZF model and Fast R-CNN. RPN uses unshared features.

	proposals		detector	mAP (%)
Two-Stage	RPN + ZF, unshared	300	Fast R-CNN + ZF, 1 scale	58.7
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 1 scale	53.8
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 5 scales	53.9

Table 11: Object detection results (%) on the **MS COCO** dataset. The model is VGG-16.

			COCO val		COCO test-dev	
method	proposals	training data	mAP@.5	mAP@[.5, .95]	mAP@.5	mAP@[.5, .95]
Fast R-CNN [2]	SS, 2000	COCO train	-	-	35.9	19.7
Fast R-CNN [impl. in this paper]	SS, 2000	COCO train	38.6	18.9	39.3	19.3
Faster R-CNN	RPN, 300	COCO train	41.5	21.2	42.1	21.5
Faster R-CNN	RPN, 300	COCO trainval	-	-	**42.7**	**21.9**

- 효율적이고 정확한 영역 제안을 위해 RPN을 제안
- dectetor network와 합성곱 feature를 공유함으로써 영역 제안 단계 드는 비용을 감소시킴 → 거의 실시간에 가까운 속도로 작동이 가능