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

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

**Abstract**

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with “attention” mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

최신 객체 감지 네트워크는 객체 위치를 가설화하기 위한 지역 제안 알고리즘에 의존합니다. SPPnet [1] 및 Fast R-CNN [2]과 같은 발전으로 탐지 네트워크의 실행 시간이 단축되어 지역 제안 계산이 병목 현상으로 노출되었습니다. 본 연구에서는 전체 이미지 컨볼루션 기능을 탐지 네트워크와 공유하여 거의 비용 없는 지역 제안을 가능하게 하는 RPN(Region Proposal Network)을 소개합니다. RPN은 각 위치에서 객체 경계와 객체성 점수를 동시에 예측하는 완전한 컨볼루션 네트워크입니다. RPN은 Fast R-CNN에서 탐지에 사용하는 고품질 지역 제안을 생성하도록 엔드 투 엔드 교육을 받습니다. 우리는 또한 RPN과 Fast R-CNN의 컨볼루션 기능을 공유함으로써 단일 네트워크로 통합합니다. 최근에 "주의" 메커니즘과 함께 널리 사용되는 신경 네트워크 용어를 사용하여 RPN 구성요소는 통합 네트워크에 어디를 찾아야 할지 알려줍니다. 매우 심층적인 VGG-16 모델[3]의 경우, PASCAL VOC 2007, 2012 및 MS COCO 데이터셋에서 이미지당 300개의 제안만으로 최신 객체 감지 정확도를 달성하면서 GPU에서 5fps(모든 단계 포함)의 프레임률을 가지고 있습니다. ILSVRC 및 COCO 2015 대회에서 Faster R-CNN 및 RPN은 여러 트랙에서 1위를 차지한 엔트리의 토대입니다. 코드가 공개되었습니다.

1. **Introduction**

Recent advances in object detection are driven by the success of region proposal methods (e.g., [4]) and region-based convolutional neural networks (RCNNs) [5]. Although region-based CNNs were computationally expensive as originally developed in [5], their cost has been drastically reduced thanks to sharing convolutions across proposals [1], [2]. The latest incarnation, Fast R-CNN [2], achieves near real-time rates using very deep networks [3], when ignoring the time spent on region proposals. Now, proposals are the test-time computational bottleneck in state-of-the-art detection systems.

물체 검출의 최근 발전은 지역 제안 방법 (예: [4])과 지역 기반 컨볼루션 신경 네트워크 (RCNN)의 성공에 의해 추진된다 [5]. 지역 기반 CNN은 [5]에서 처음 개발한 것처럼 계산적으로 비용이 많이 들었지만, 제안서 [1], [2]에 걸쳐 경합을 공유함으로써 비용이 대폭 절감되었습니다. 최신 제품인 Fast R-CNN [2]은 지역 제안에 소요되는 시간을 무시해도 매우 깊은 네트워크를 사용하여 실시간에 가까운 속도를 달성합니다[3]. 이제 제안서는 최첨단 탐지 시스템에서 테스트 시간의 계산 병목 현상이 됩니다.

Region proposal methods typically rely on inexpensive features and economical inference schemes. Selective Search [4], one of the most popular methods, greedily merges superpixels based on engineered low-level features. Yet when compared to efficient detection networks [2], Selective Search is an order of magnitude slower, at 2 seconds per image in a CPU implementation. EdgeBoxes [6] currently provides the best tradeoff between proposal quality and speed, at 0.2 seconds per image. Nevertheless, the region proposal step still consumes as much running time as the detection network.

지역 제안 방법은 일반적으로 저렴한 특징과 경제적 추론 체계에 의존한다. 가장 인기 있는 방법 중 하나인 선택적 검색[4]은 엔지니어링된 저수준 기능을 기반으로 슈퍼픽셀을 의도적으로 병합합니다. 그러나 효율적인 탐지 네트워크[2]와 비교할 때, 선택적 검색은 CPU 구현에서 이미지당 2초씩 더 느립니다. EdgeBoxes[6]는 현재 이미지당 0.2초로 제안 품질과 속도 간에 최상의 트레이드오프를 제공합니다. 그럼에도 불구하고 지역 제안 단계는 여전히 탐지 네트워크만큼의 실행 시간을 소비합니다.

One may note that fast region-based CNNs take advantage of GPUs, while the region proposal methods used in research are implemented on the CPU, making such runtime comparisons inequitable. An obvious way to accelerate proposal computation is to reimplement it for the GPU. This may be an effective engineering solution, but re-implementation ignores the down-stream detection network and therefore misses important opportunities for sharing computation.

빠른 지역 기반 CNN은 GPU를 활용하는 반면 연구에 사용되는 지역 제안 방식은 CPU에 구현되기 때문에 이러한 런타임 비교가 불가능하다는 점에 주목해야 할 것이다. 제안서 계산을 가속화하는 분명한 방법은 GPU를 위해 이를 다시 구현하는 것입니다. 이는 효과적인 엔지니어링 솔루션일 수 있지만, 다시 구현하면 다운스트림 탐지 네트워크가 무시되므로 계산을 공유할 수 있는 중요한 기회를 놓치게 됩니다.

In this paper, we show that an algorithmic change—computing proposals with a deep convolutional neural network—leads to an elegant and effective solution where proposal computation is nearly cost-free given the detection network’s computation. To this end, we introduce novel Region Proposal Networks (RPNs) that share convolutional layers with state-of-the-art object detection networks [1], [2]. By sharing convolutions at test-time, the marginal cost for computing proposals is small (e.g., 10ms per image).

이 논문에서 우리는 알고리즘적 변화(심층 신경망을 통한 제안 컴퓨팅)가 탐지 네트워크의 연산을 감안할 때 제안 연산이 거의 비용 부담 없는 우아하고 효과적인 솔루션으로 이어진다는 것을 보여줍니다. 이를 위해 최신 객체 감지 네트워크[1], [2]와 컨볼루션 계층을 공유하는 새로운 지역 제안 네트워크(RPN)를 소개한다. 테스트 시점에 컨볼루션을 공유함으로써 제안서 컴퓨팅에 대한 한계 비용이 작습니다(예: 이미지당 10ms).

Our observation is that the convolutional feature maps used by region-based detectors, like Fast RCNN, can also be used for generating region proposals. On top of these convolutional features, we construct an RPN by adding a few additional convolutional layers that simultaneously regress region bounds and objectness scores at each location on a regular grid. The RPN is thus a kind of fully convolutional network (FCN) [7] and can be trained end-toend specifically for the task for generating detection proposals.

Fast RCN과 같이 지역 기반 검출기가 사용하는 컨볼루션 피쳐 맵도 지역 제안 생성에 사용할 수 있다는 것이 우리의 관찰입니다. 이러한 컨볼루션 기능 위에, 우리는 정규 그리드의 각 위치에서 영역 경계와 객체성 점수를 동시에 회귀시키는 몇 개의 추가 컨볼루션 레이어를 추가하여 RPN을 구성한다. 따라서 RPN은 일종의 완전 콘볼루션 네트워크(FCN)[7]이며, 검출 제안서 생성 작업에 대해 특별히 엔드 투 엔드 훈련을 받을 수 있다.

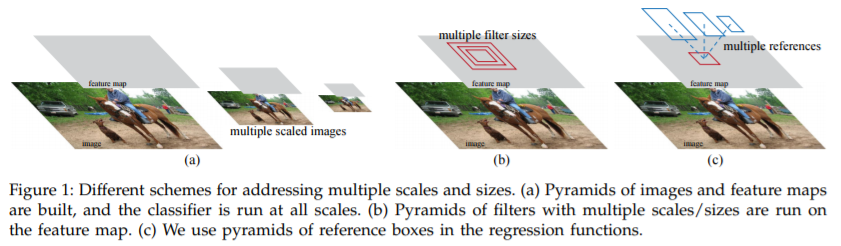


Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

그림 1: 복수의 축척과 크기에 대응하는 다른 방식. (a) 이미지 및 피처 맵의 피라미드를 구축하고 분류기를 모든 축척으로 실행한다. (b) 피처 맵에서 여러 축척/사이즈를 가진 필터의 피라미드를 실행한다. (c) 회귀 함수에서 참조 상자의 피라미드를 사용한다.

RPNs are designed to efficiently predict region proposals with a wide range of scales and aspect ratios. In contrast to prevalent methods [8], [9], [1], [2] that use pyramids of images (Figure 1, a) or pyramids of filters (Figure 1, b), we introduce novel “anchor” boxes that serve as references at multiple scales and aspect ratios. Our scheme can be thought of as a pyramid of regression references (Figure 1, c), which avoids enumerating images or filters of multiple scales or aspect ratios. This model performs well when trained and tested using single-scale images and thus benefits running speed.

RPN은 광범위한 척도 및 가로 세로 비율을 사용하여 지역 제안을 효율적으로 예측하도록 설계되었습니다. 이미지의 피라미드(그림 1, a) 또는 필터의 피라미드(그림 1, b)를 사용하는 일반적인 방법[8], [9], [1], [2]과 달리, 우리는 다중 척도 및 가로 세로 비율에서 참조 역할을 하는 새로운 "앵커" 상자를 도입한다. 우리의 계획은 여러 축척 또는 가로 세로 비율의 이미지나 필터를 열거하지 않는 회귀 참조의 피라미드로 생각할 수 있다(그림 1, c). 이 모델은 단일 스케일 이미지를 사용하여 교육 및 테스트했을 때 성능이 우수하여 실행 속도에 유리합니다.

To unify RPNs with Fast R-CNN [2] object detection networks, we propose a training scheme that alternates between fine-tuning for the region proposal task and then fine-tuning for object detection, while keeping the proposals fixed. This scheme converges quickly and produces a unified network with convolutional features that are shared between both tasks.

RPN을 Fast R-CNN [2] 객체 감지 네트워크와 통합하기 위해 제안서를 고정하는 동시에 지역 제안 작업에 대한 미세 조정과 객체 감지를 위한 미세 조정을 번갈아 수행하는 교육 체계를 제안합니다. 이 계획은 빠르게 통합되고 두 작업 간에 공유되는 컨볼루션 기능을 갖춘 통합 네트워크를 생성합니다.

We comprehensively evaluate our method on the PASCAL VOC detection benchmarks [11] where RPNs with Fast R-CNNs produce detection accuracy better than the strong baseline of Selective Search with Fast R-CNNs. Meanwhile, our method waives nearly all computational burdens of Selective Search at test-time—the effective running time for proposals is just 10 milliseconds. Using the expensive very deep models of [3], our detection method still has a frame rate of 5fps (including all steps) on a GPU, and thus is a practical object detection system in terms of both speed and accuracy. We also report results on the MS COCO dataset [12] and investigate the improvements on PASCAL VOC using the COCO data. Code has been made publicly available at https://github.com/shaoqingren/faster\_rcnn (in MATLAB) and https://github.com/rbgirshick/py-faster-rcnn (in Python).

우리는 빠른 R-CNN을 사용하는 RPN이 빠른 R-CNN을 사용하는 선택적 검색의 강력한 기준보다 더 나은 탐지 정확도를 생성하는 PASCAL VOC 탐지 벤치마크[11]에 대해 우리의 방법을 종합적으로 평가합니다. 반면, 우리의 방법은 테스트 시간에 선택적 검색의 거의 모든 계산 부담을 포기합니다. 제안의 효과적인 실행 시간은 단지 그것입니다.t 10밀리초. [3]의 매우 고가의 심층 모델을 사용하는 NAT의 탐지 방법은 GPU에서 여전히 프레임률이 5fps(모든 단계 포함)이므로 속도와 정확도 측면에서 실용적인 객체 탐지 시스템입니다. 또한 MS COCO 데이터 세트[12]에 대한 결과를 보고하고 COCO 데이터를 사용하여 PASCAL VOC의 개선 사항을 조사합니다. 코드는 https://github.com/shaoqingren/faster\_rcnn(MATLAB) 및 https://github.com/rbgirshick/py-faster-rcnn(Python)에서 공개되었습니다.

A preliminary version of this manuscript was published previously [10]. Since then, the frameworks of RPN and Faster R-CNN have been adopted and generalized to other methods, such as 3D object detection [13], part-based detection [14], instance segmentation [15], and image captioning [16]. Our fast and effective object detection system has also been built in commercial systems such as at Pinterests [17], with user engagement improvements reported.

이 원고의 예비 버전은 이전에 출판되었습니다[10]. 그 이후, RPN과 Faster R-CNN의 프레임워크가 채택되어 3D 객체 감지[13], 부품 기반 감지[14], 인스턴스 분할[15], 이미지 캡션[16]과 같은 다른 방법으로 일반화되었습니다. NAT의 빠르고 효과적인 객체 감지 시스템은 Pinterests[17]와 같은 상업용 시스템에도 구축되었으며 사용자 참여도 개선되었다고 보고되었습니다.

In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the basis of several 1st-place entries [18] in the tracks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. RPNs completely learn to propose regions from data, and thus can easily benefit from deeper and more expressive features (such as the 101-layer residual nets adopted in [18]). Faster R-CNN and RPN are also used by several other leading entries in these competitions2. These results suggest that our method is not only a cost-efficient solution for practical usage, but also an effective way of improving object detection accuracy.

ILSVRC 및 COCO 2015 대회에서 Faster R-CNN 및 RPN은 ImageNet 탐지, ImageNet 현지화, COCO 탐지 및 COCO 세분화 트랙의 여러 1위 항목[18]의 기본입니다. RPN은 데이터에서 영역을 제안하는 방법을 완전히 학습하기 때문에 더 깊고 표현력이 뛰어난 기능(예: [18]에서 채택된 101층 잔류망)을 쉽게 이용할 수 있다. 더 빠른 R-CNN과 RPN은 이러한 경기의 다른 몇몇 선두 참가자들에 의해서도 사용됩니다2. 이러한 결과는 당사의 방법이 실제 사용을 위한 비용 효율적인 솔루션일 뿐만 아니라 객체 감지 정확도를 향상시킬 수 있는 효과적인 방법임을 시사합니다.

1. **Related Work**

**Object Proposals**. There is a large literature on object proposal methods. Comprehensive surveys and comparisons of object proposal methods can be found in [19], [20], [21]. Widely used object proposal methods include those based on grouping super-pixels (e.g.,Selective Search [4], CPMC [22], MCG [23]) and those based on sliding windows (e.g., objectness in windows [24], EdgeBoxes [6]). Object proposal methods were adopted as external modules independent of the detectors (e.g., Selective Search [4] object detectors, RCNN [5], and Fast R-CNN [2]).

객체 제안. 객체 제안 방법에 대한 많은 문헌이 있습니다. 객체 제안 방법의 종합적인 조사 및 비교는 [19], [20], [21]에서 확인할 수 있습니다. 널리 사용되는 객체 제안 방법에는 그룹화 슈퍼픽셀에 기반한 방법이 포함된다(예:선택 검색 [4], CPMC [22], MCG [23]) 및 슬라이딩 윈도우에 기반한 검색(예: 윈도우의 객체성 [24], EdgeBoxes [6]). 객체 제안 방법은 검출기와 독립적인 외부 모듈로 채택되었다(예: 선택적 검색 [4] 객체 검출기, RCNN [5], 고속 R-CNN [2]).

**Deep Networks for Object Detection.** The R-CNN method [5] trains CNNs end-to-end to classify the proposal regions into object categories or background. R-CNN mainly plays as a classifier, and it does not predict object bounds (except for refining by bounding box regression). Its accuracy depends on the performance of the region proposal module (see comparisons in [20]). Several papers have proposed ways of using deep networks for predicting object bounding boxes [25], [9], [26], [27]. In the OverFeat method [9], a fully-connected layer is trained to predict the box coordinates for the localization task that assumes a single object. The fully-connected layer is then turned into a convolutional layer for detecting multiple classspecific objects. The MultiBox methods [26], [27] generate region proposals from a network whose last fully-connected layer simultaneously predicts multiple class agnostic boxes, generalizing the “singlebox” fashion of OverFeat. These class-agnostic boxes are used as proposals for R-CNN [5]. The MultiBox proposal network is applied on a single image crop or multiple large image crops (e.g., 224×224), in contrast to our fully convolutional scheme. MultiBox does not share features between the proposal and detection networks. We discuss OverFeat and MultiBox in more depth later in context with our method. Concurrent with our work, the DeepMask method [28] is developed for learning segmentation proposals.

개체 탐지를 위한 심층 네트워크입니다. R-CNN 방법[5]은 제안 영역을 객체 범주나 배경으로 분류하도록 CNN을 전체적으로 훈련시킵니다. R-CNN은 주로 분류자로 재생되며 개체 경계를 예측하지 않습니다(바운딩 상자 회귀 분석으로 미세 조정되는 것 제외). 정확도는 지역 제안 모듈의 성능에 따라 달라집니다(20]의 비교 참조). 여러 논문에서 객체 경계 상자 예측에 심층 네트워크를 사용하는 방법을 제안했습니다 [25], [9], [26], [27]. OverFeat 방법[9]에서는 완전히 연결된 레이어가 단일 객체를 가정하는 위치화 작업에 대한 상자 좌표를 예측하도록 교육됩니다. 그런 다음 완전히 연결된 레이어가 여러 클래스별 객체를 탐지하기 위한 컨볼루션 레이어로 바뀝니다. MultiBox 메서드 [26], [27]는 마지막으로 완전히 연결된 계층이 동시에 여러 클래스 구분 상자를 예측하여 OverFeat의 "단일 상자" 스타일을 일반화하는 네트워크에서 영역 제안을 생성합니다. 이 클래스 구분 상자는 R-CNN[5]에 대한 제안으로 사용됩니다. MultiBox 제안 네트워크는 단일 이미지 자르기 또는 여러 개의 대형 이미지 자르기(예: 224×224)에 적용되며, 이는 완전한 컨볼루션 구성과 대조됩니다. MultiBox는 제안 네트워크와 탐지 네트워크 간에 기능을 공유하지 않습니다. OverFeat 및 MultiBox에 대해서는 나중에 NAT의 방법에 대해 자세히 설명합니다. 당사의 작업과 동시에 학습 세분화 제안을 위해 DeepMask 방법[28]이 개발되었습니다.

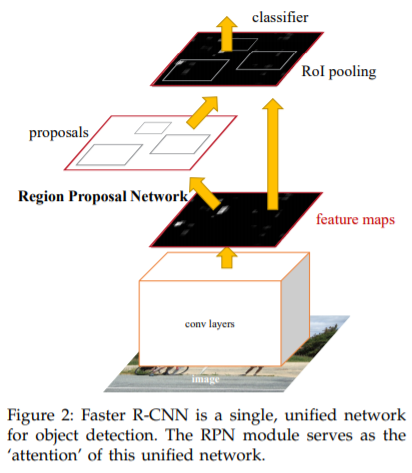


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the ‘attention’ of this unified network.

그림 2: 고속 R-CNN은 객체 검출을 위한 단일 통합 네트워크입니다. RPN 모듈은 이 통합 네트워크의 '주의' 역할을 합니다.

Shared computation of convolutions [9], [1], [29], [7], [2] has been attracting increasing attention for efficient, yet accurate, visual recognition. The OverFeat paper [9] computes convolutional features from an image pyramid for classification, localization, and detection. Adaptively-sized pooling (SPP) [1] on shared convolutional feature maps is developed for efficient region-based object detection [1], [30] and semantic segmentation [29]. Fast R-CNN [2] enables end-to-end detector training on shared convolutional features and shows compelling accuracy and speed.

[9], [1], [29], [7], [2]에 대한 공유 계산은 효율적이면서도 정확한 시각적 인식을 위해 점점 더 관심을 끌고 있습니다. OverFeat 문서[9]는 분류, 위치 측정 및 탐지를 위해 이미지 피라미드에서 컨볼루션 피쳐를 계산합니다. 효율적인 지역 기반 객체 감지 [1], [30] 및 의미 세분화를 위해 공유된 컨볼루션 피쳐 맵의 적응형 크기 풀링(SPP)[1]이 개발되었다[29]. 빠른 R-CNN [2]을 통해 공유된 컨볼루션 기능에 대한 엔드 투 엔드 디텍터 교육이 가능하며 뛰어난 정확도와 속도를 보여줍니다.

**3. FASTER R-CNN**

Our object detection system, called Faster R-CNN, is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector [2] that uses the proposed regions. The entire system is a single, unified network for object detection (Figure 2). Using the recently popular terminology of neural networks with ‘attention’ [31] mechanisms, the RPN module tells the Fast R-CNN module where to look. In Section 3.1 we introduce the designs and properties of the network for region proposal. In Section 3.2 we develop algorithms for training both modules with features shared.

Faster R-CNN이라고 불리는 우리의 물체 감지 시스템은 두 개의 모듈로 구성되어 있습니다. 첫 번째 모듈은 영역을 제안하는 완전 콘볼루션 네트워크이며, 두 번째 모듈은 제안된 영역을 사용하는 고속 R-CNN 검출기 [2]입니다. 전체 시스템은 객체 감지를 위한 단일 통합 네트워크입니다(그림 2). 최근 '주의'[31] 메커니즘으로 널리 알려진 신경 네트워크 용어를 사용하여 RPN 모듈은 Fast R-CNN 모듈에게 어디를 찾아야 하는지 알려줍니다. 섹션 3.1에서는 지역 제안을 위한 네트워크의 설계와 특성을 소개합니다. 섹션 3.2에서는 기능이 공유된 두 모듈을 교육하기 위한 알고리즘을 개발합니다.

**3.1 Region Proposal Networks**

A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score.3 We model this process with a fully convolutional network [7], which we describe in this section. Because our ultimate goal is to share computation with a Fast R-CNN object detection network [2], we assume that both nets share a common set of convolutional layers. In our experiments, we investigate the Zeiler and Fergus model [32] (ZF), which has 5 shareable convolutional layers and the Simonyan and Zisserman model [3] (VGG-16), which has 13 shareable convolutional layers.

RPN(Region Proposal Network)은 (모든 크기의) 이미지를 입력으로 사용하고 각각 객체성 점수가 있는 직사각형 객체 제안 세트를 출력합니다.3 우리는 이 절에서 설명하는 완전한 컨볼루션 네트워크[7]를 사용하여 이 프로세스를 모델링합니다. 우리의 궁극적인 목표는 고속 R-CNN 객체 감지 네트워크[2]와 계산을 공유하는 것이기 때문에, 우리는 두 네트가 공통의 컨볼루션 레이어를 공유한다고 가정합니다. 실험에서는 5개의 공유 가능한 컨볼루션 레이어가 있는 Zeiler 및 Fergus 모델[32](ZF)과 13개의 공유 가능한 컨볼루션 레이어가 있는 Simonyan 및 Ziserman 모델[3](VG-16)을 조사한다.

To generate region proposals, we slide a small network over the convolutional feature map output by the last shared convolutional layer. This small network takes as input an n × n spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature (256-d for ZF and 512-d for VGG, with ReLU [33] following). This feature is fed into two sibling fullyconnected layers—a box-regression layer (reg) and a box-classification layer (cls). We use n = 3 in this paper, noting that the effective receptive field on the input image is large (171 and 228 pixels for ZF and VGG, respectively). This mini-network is illustrated at a single position in Figure 3 (left). Note that because the mini-network operates in a sliding-window fashion, the fully-connected layers are shared across all spatial locations. This architecture is naturally implemented with an n×n convolutional layer followed by two sibling 1 × 1 convolutional layers (for reg and cls, respectively).

지역 제안을 생성하기 위해 마지막 공유 컨볼루션 계층에 의한 컨볼루션 피쳐 맵 출력 위로 작은 네트워크를 밀어넣습니다. 이 작은 네트워크는 입력 컨볼루션 피쳐 맵의 입력 n × n 공간 창을 받아들입니다. 각 슬라이딩 윈도우는 저차원 기능(ZF의 경우 256-d, VGG의 경우 512-d, 다음과 같이 ReLU [33])으로 매핑됩니다. 이 기능은 완전히 연결된 두 개의 형제 계층, 즉 상자-회귀 계층(reg)과 상자 분류 계층(cls)에 공급됩니다. 본 문서에서는 입력 이미지의 유효 수신 필드가 크다는 점에 주목하여 n = 3을 사용합니다(ZF 및 VGG의 경우 각각 164픽셀과 228픽셀). 이 미니 네트워크는 그림 3(왼쪽)의 단일 위치에서 설명됩니다. 미니 네트워크는 슬라이딩 창 방식으로 작동하므로 완전히 연결된 레이어는 모든 공간 위치에서 공유됩니다. 이 아키텍처는 자연스럽게 n×n convolution layer에 이어 두 개의 형제 1 × 1 convolution layer (각각 reg와 cls에 대해)로 구현됩니다.

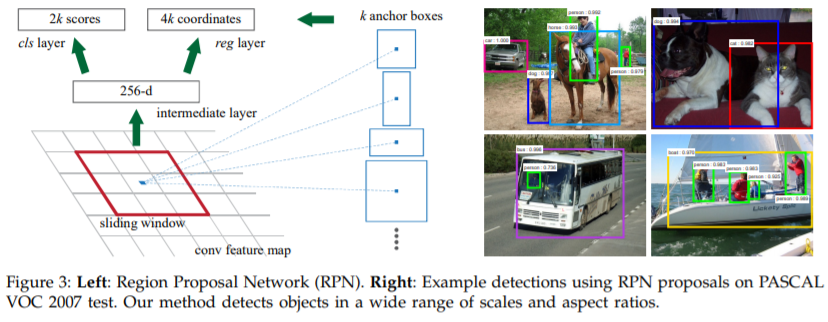


Figure 3: Left: Region Proposal Network (RPN). Right: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

그림 3: 왼쪽: 지역 프로포절 네트워크(RPN) 오른쪽: PASCAL VOC 2007 테스트에서 RPN 제안을 사용한 검출 예. 우리의 방법은 광범위한 축척과 종횡비에서 물체를 감지한다.

**3.1.1. Anchors**

At each sliding-window location, we simultaneously predict multiple region proposals, where the number of maximum possible proposals for each location is denoted as k. So the reg layer has 4k outputs encoding the coordinates of k boxes, and the cls layer outputs 2k scores that estimate probability of object or not object for each proposal4. The k proposals are parameterized relative to k reference boxes, which we call anchors. An anchor is centered at the sliding window in question, and is associated with a scale and aspect ratio (Figure 3, left). By default we use 3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position. For a convolutional feature map of a size W × H (typically ∼2,400), there are W Hk anchors in total.

각 슬라이딩 윈도우 위치에서 여러 지역 제안을 동시에 예측하며, 각 위치에 대해 가능한 최대 제안 수가 표시됩니다. 따라서 레지스트리 계층은 k 상자의 좌표를 인코딩하는 4k 출력을 가지며, cls 계층은 각 제안에 대해 객체 또는 객체가 아닌 확률을 추정하는 2k 점수를 출력합니다4. k 제안서는 우리가 앵커라고 부르는 k 참조 박스에 상대적인 매개변수를 가집니다. 앵커는 해당 슬라이딩 윈도우의 중앙에 위치하며 축척 및 가로 세로 비율과 관련이 있습니다(그림 3, 왼쪽). 기본적으로 3개의 축척과 3개의 가로 세로비를 사용하며 각 슬라이딩 위치에서 k = 9개의 앵커를 산출합니다. 크기 W × H (일반적으로 ~ 2,400)의 컨볼루션 특성 맵의 경우, 총 W Hk 앵커가 있습니다.

**Translation-Invariant Anchors**

An important property of our approach is that it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. If one translates an object in an image, the proposal should translate and the same function should be able to predict the proposal in either location. This translation-invariant property is guaranteed by our method5. As a comparison, the MultiBox method [27] uses k-means to generate 800 anchors, which are not translation invariant. So MultiBox does not guarantee that the same proposal is generated if an object is translated.

우리의 접근법의 중요한 특성은 앵커와 앵커에 관련된 제안을 계산하는 함수 측면에서 모두 변환 불변이라는 것입니다. 이미지에서 객체를 번역하는 경우 제안서를 번역해야 하며 두 위치 모두에서 동일한 기능이 제안을 예측할 수 있어야 합니다. 이 번역 불변 속성은 당사의 방법 5에 의해 보장됩니다. 이에 비해 MultiBox 방법[27]은 k-평균을 사용하여 변환 불변량이 아닌 800개의 앵커를 생성합니다. 따라서 MultiBox는 객체를 번역할 때 동일한 제안이 생성된다는 것을 보장하지 않습니다.

The translation-invariant property also reduces the model size. MultiBox has a (4 + 1) × 800-dimensional fully-connected output layer, whereas our method has a (4 + 2) × 9-dimensional convolutional output layer in the case of k = 9 anchors. As a result, our output layer has 2.8 × 104 parameters (512 × (4 + 2) × 9 for VGG-16), two orders of magnitude fewer than MultiBox’s output layer that has 6.1 × 106 parameters (1536 × (4 + 1) × 800 for GoogleNet [34] in MultiBox [27]). If considering the feature projection layers, our proposal layers still have an order of magnitude fewer parameters than MultiBox6. We expect our method to have less risk of overfitting on small datasets, like PASCAL VOC.

변환 불변 속성을 사용하면 모델 크기도 줄어듭니다. MultiBox는 (4 + 1) × 800차원 완전 연결 출력 계층을 가지고 있는 반면, 우리의 방법은 (4 + 2) × 9차원 컨볼루션 출력 계층을 가지고 있습니다. 결과적으로, 우리의 출력 계층은 2.8 × 104 매개변수 (512 × (4 + 2) × VGG-16의 경우)를 가집니다. 이는 6.1 × 106 매개변수를 가진 MultiBox의 출력 계층 (1536 × (4 + 1) × 800 in MultiBox [27]의 [34]보다 2배 작은 크기입니다. 피쳐 투영 레이어를 고려하더라도 제안 레이어는 MultiBox6보다 훨씬 작은 매개변수를 가집니다. PASCAL VOC와 같은 소규모 데이터셋에서 과다 적합 위험이 줄어들 것으로 예상합니다.

**Multi-Scale Anchors as Regression References**

Our design of anchors presents a novel scheme for addressing multiple scales (and aspect ratios). As shown in Figure 1, there have been two popular ways for multi-scale predictions. The first way is based on image/feature pyramids, e.g., in DPM [8] and CNNbased methods [9], [1], [2]. The images are resized at multiple scales, and feature maps (HOG [8] or deep convolutional features [9], [1], [2]) are computed for each scale (Figure 1(a)). This way is often useful but is time-consuming. The second way is to use sliding windows of multiple scales (and/or aspect ratios) on the feature maps. For example, in DPM [8], models of different aspect ratios are trained separately using different filter sizes (such as 5×7 and 7×5). If this way is used to address multiple scales, it can be thought of as a “pyramid of filters” (Figure 1(b)). The second way is usually adopted jointly with the first way [8].

우리의 앵커 설계는 다중 척도(및 가로 세로 비율)를 다루기 위한 새로운 계획을 제시합니다. 그림 1에서 볼 수 있듯이, 다단계 예측에는 두 가지 방법이 있습니다. 첫 번째 방법은 이미지/기능 피라미드를 기반으로 합니다. 예를 들어, DPM[8] 및 CNN 기반 방법[9], [1], [2]. 영상은 여러 척도로 크기가 조정되며, 피쳐 맵(HOG[8] 또는 딥 컨볼루션 피쳐[9], [1], [2])이 각 척도에 대해 계산됩니다(그림 1(a)). 이 방법은 종종 유용하지만 시간이 많이 걸립니다. 두 번째 방법은 피쳐 맵에서 다중 축척(및/또는 가로 세로 비율)의 슬라이딩 창을 사용하는 것입니다. 예를 들어 DPM[8]에서는 가로 세로 비율이 서로 다른 모델이 서로 다른 필터 크기(5×7 및 7×5 등)를 사용하여 별도로 교육됩니다. 이 방법을 사용하여 여러 척도를 처리할 경우 "필터의 피라미드"라고 생각할 수 있습니다(그림 1(b)) 두 번째 방법은 일반적으로 첫 번째 방법과 공동으로 채택된다[8].

As a comparison, our anchor-based method is built on a pyramid of anchors, which is more cost-efficient. Our method classifies and regresses bounding boxes with reference to anchor boxes of multiple scales and aspect ratios. It only relies on images and feature maps of a single scale, and uses filters (sliding windows on the feature map) of a single size. We show by experiments the effects of this scheme for addressing multiple scales and sizes (Table 8).

그에 비해, 우리의 앵커 기반 방식은 앵커 피라미드 위에 구축되어 있어 비용 효율적입니다. 이 방법은 다중 척도 및 가로 세로 비율의 앵커 박스를 기준으로 경계 박스를 분류하고 회귀 분석합니다. 단일 축척의 이미지 및 피쳐 맵에만 의존하며 단일 크기의 필터( 피쳐 맵의 슬라이딩 창)를 사용합니다. 여러 척도와 크기를 다루기 위한 이 계획의 효과를 실험으로 보여 줍니다(표 8).

Because of this multi-scale design based on anchors, we can simply use the convolutional features computed on a single-scale image, as is also done by the Fast R-CNN detector [2]. The design of multiscale anchors is a key component for sharing features without extra cost for addressing scales.

앵커에 기반한 이러한 멀티스케일 설계로 인해 Fast R-CNN 검출기[2]에 의해 수행되는 것처럼 단일 스케일 이미지에서 계산되는 컨볼루션 기능을 간단히 사용할 수 있다. 멀티스케일 앵커 설계는 축척 문제를 해결하기 위한 추가 비용 없이 기능을 공유할 수 있는 핵심 구성요소입니다.

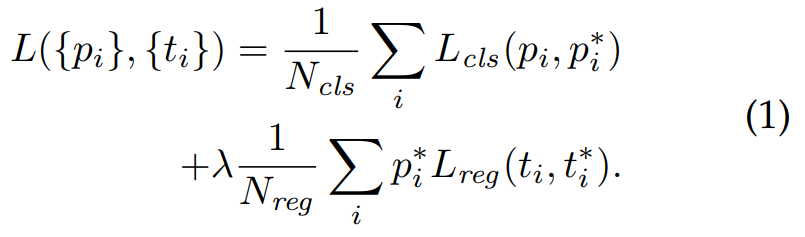
**3.1.2 Loss Function**

For training RPNs, we assign a binary class label (of being an object or not) to each anchor. We assign a positive label to two kinds of anchors: (i) the anchor/anchors with the highest Intersection-overUnion (IoU) overlap with a ground-truth box, or (ii) an anchor that has an IoU overlap higher than 0.7 with any ground-truth box. Note that a single ground-truth box may assign positive labels to multiple anchors. Usually the second condition is sufficient to determine the positive samples; but we still adopt the first condition for the reason that in some rare cases the second condition may find no positive sample. We assign a negative label to a non-positive anchor if its IoU ratio is lower than 0.3 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective.

교육용 RPN의 경우 각 앵커에 (개체인지 아닌지에 대한) 이진 클래스 레이블을 지정합니다. 우리는 두 종류의 앵커에 양의 레이블을 할당합니다. (i) 가장 높은 교차점-오버 유니언(Intersection-over Union, IoU)이 지면-진실 상자와 겹치는 앵커 또는 (ii) 모든 지면-진실 상자와 0.7 이상 겹치는 앵커입니다. 단일 접지 진실 상자가 여러 앵커에 양의 레이블을 할당할 수 있습니다. 일반적으로 두 번째 조건은 양성 검체를 확인하는 데 충분하지만, 드문 경우 두 번째 조건에서는 양성 검체가 발견되지 않을 수 있기 때문에 여전히 첫 번째 조건을 채택하고 있습니다. 비양성 앵커의 IoU 비율이 모든 접지-진실 박스에 대해 0.3 미만일 경우 음의 라벨을 할당합니다. 긍정적이거나 부정적이지 않은 앵커는 교육 목표에 기여하지 않습니다.

With these definitions, we minimize an objective function following the multi-task loss in Fast R-CNN [2]. Our loss function for an image is defined as:

이러한 정의를 통해, 우리는 Fast R-CNN [2]에서 다중 작업 손실을 따르는 객관적 함수를 최소화합니다. 이미지에 대한 손실 함수는 다음과 같이 정의됩니다.



Here, i is the index of an anchor in a mini-batch and pi is the predicted probability of anchor i being an object. The ground-truth label p∗i is 1 if the anchor is positive, and is 0 if the anchor is negative. ti is a vector representing the 4 parameterized coordinates of the predicted bounding box, and t∗i is that of the ground-truth box associated with a positive anchor. The classification loss Lcls is log loss over two classes (object vs. not object). For the regression loss, we use Lreg (ti, t∗i) = R(ti − t∗i) where R is the robust loss function (smooth L1) defined in [2]. The term p∗i Lreg means the regression loss is activated only for positive anchors (p∗i = 1) and is disabled otherwise (p∗i = 0). The outputs of the cls and reg layers consist of {pi} and {ti} respectively.

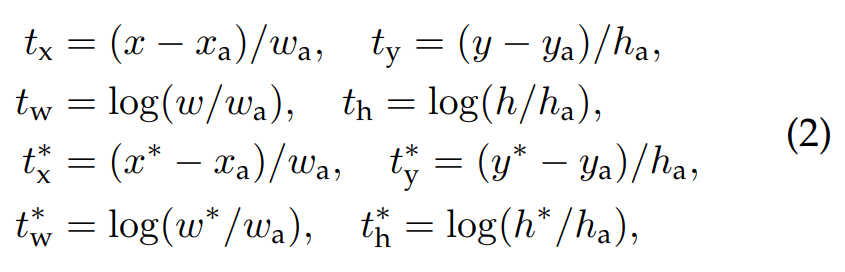
여기서 i는 미니 배치에서 앵커의 지수이고 pi는 물체가 될 수 있는 예측 확률입니다. 접지-진실 라벨 pii는 앵커가 양수이면 1이고, 앵커가 음수이면 0이며, t isi는 예측한 경계상자의 4개 매개변수 좌표를 나타내는 벡터이며, tii는 양의 앵커와 연관된 접지-진실상자의 것이다. 분류 손실 Lcls는 두 클래스에 대한 로그 손실입니다(개체 대 개체가 아님). 회귀손실의 경우, 우리는 Lreg (ti, tii) = R(ti - tii)를 사용한다. 여기서 R은 [2]에 정의된 강건한 손실 함수(평활한 L1)이다. pii Lreg라는 용어는 회귀 손실이 양의 앵커(p onlyi = 1)에 대해서만 활성화되고 그렇지 않으면 비활성화됨을 의미합니다(p∗i = 0). cls 및 레지스트리 계층의 출력은 각각 {pi} 및 {ti}(으)로 구성됩니다.

The two terms are normalized by Ncls and Nreg and weighted by a balancing parameter λ. In our current implementation (as in the released code), the cls term in Eqn.(1) is normalized by the mini-batch size (i.e., Ncls = 256) and the reg term is normalized by the number of anchor locations (i.e., Nreg ∼ 2, 400). By default we set λ = 10, and thus both cls and reg terms are roughly equally weighted. We show by experiments that the results are insensitive to the values of λ in a wide range (Table 9). We also note that the normalization as above is not required and could be simplified.

두 항은 Ncls와 Nreg에 의해 정규화되고 균형 매개변수 λ에 의해 가중됩니다. 현재 구현(공개된 코드와 같이)에서 Eqn(1)의 cls 항은 최소 크기(즉, Ncls = 256)에 의해 정규화되며, 정규 항은 앵커 위치 수(즉, Nreg ~ 2,400)에 의해 정규화된다. 기본적으로 λ = 10으로 설정되므로 cls 항과 reg 항 모두 거의 동일하게 가중치가 적용됩니다. 실험을 통해 그 결과가 넓은 범위의 λ 값에 무감각하다는 것을 알 수 있습니다(표 9). 또한 위와 같은 정규화가 필요하지 않으며 단순화될 수 있다는 점에 유의합니다.

For bounding box regression, we adopt the parameterizations of the 4 coordinates following [5]:

경계 상자 회귀 분석의 경우 [5]에 따라 4개 좌표에 대한 모수화를 채택합니다.



where x, y, w, and h denote the box’s center coordinates and its width and height. Variables x, xa, and x∗ are for the predicted box, anchor box, and groundtruth box respectively (likewise for y, w, h). This can be thought of as bounding-box regression from an anchor box to a nearby ground-truth box.

여기서 x, y, w 및 h는 상자의 중심 좌표와 너비와 높이를 나타냅니다. 변수 x, xa 및 x³는 각각 예측 상자, 앵커 상자 및 접지 진실 상자(y, w, h와 같음)에 대한 값입니다. 이것은 앵커 박스에서 가까운 지면-진실 박스로의 경계 상자 회귀로 간주할 수 있습니다.

Nevertheless, our method achieves bounding-box regression by a different manner from previous RoIbased (Region of Interest) methods [1], [2]. In [1], [2], bounding-box regression is performed on features pooled from arbitrarily sized RoIs, and the regression weights are shared by all region sizes. In our formulation, the features used for regression are of the same spatial size (3 × 3) on the feature maps. To account for varying sizes, a set of k bounding-box regressors are learned. Each regressor is responsible for one scale and one aspect ratio, and the k regressors do not share weights. As such, it is still possible to predict boxes of various sizes even though the features are of a fixed size/scale, thanks to the design of anchors.

그럼에도 불구하고, 우리의 방법은 이전의 RoI 기반 (관심 영역) 방법[1], [2]과는 다른 방법으로 경계 상자 회귀 분석을 달성한다. [1], [2]에서 경계 상자 회귀는 임의 크기 RoI에서 풀링된 피쳐에 대해 수행되며 회귀 가중치는 모든 영역 크기에 의해 공유됩니다. 우리의 공식화에서, 회귀에 사용되는 특징들은 피쳐 맵에서 공간 크기(3 × 3)가 같습니다. 다양한 크기를 고려하기 위해 k 경계 상자 회귀기 집합을 학습합니다. 각 회귀 분석기는 척도와 가로 세로 비율을 하나씩 담당하며, K 회귀 분석기는 가중치를 공유하지 않습니다. 이와 같이 앵커 설계 덕분에 고정된 크기/스케일이라도 다양한 크기의 박스 예측이 가능하다.

**3.1.3 Training RPNs**

The RPN can be trained end-to-end by backpropagation and stochastic gradient descent (SGD) [35]. We follow the “image-centric” sampling strategy from [2] to train this network. Each mini-batch arises from a single image that contains many positive and negative example anchors. It is possible to optimize for the loss functions of all anchors, but this will bias towards negative samples as they are dominate. Instead, we randomly sample 256 anchors in an image to compute the loss function of a mini-batch, where the sampled positive and negative anchors have a ratio of up to 1:1. If there are fewer than 128 positive samples in an image, we pad the mini-batch with negative ones.

RPN은 역전파 및 확률적 구배 하강(SGD)에 의해 엔드 투 엔드 훈련될 수 있다[35]. 우리는 [2]의 "이미지 중심" 샘플링 전략을 따라 이 네트워크를 교육합니다. 각 미니 배치는 많은 긍정 및 부정 예제 앵커가 포함된 단일 이미지에서 발생합니다. 모든 앵커의 손실 함수에 대해 최적화할 수 있지만, 이는 지배적이기 때문에 음수 표본으로 치우치게 됩니다. 대신 이미지에서 임의로 256개의 앵커를 샘플링하여 샘플링된 양의 앵커와 음의 앵커의 비율이 최대 1:1인 미니 배치의 손실 함수를 계산합니다. 이미지에 양성 샘플이 128개 미만이면 미니배치에 음성 샘플로 패딩을 합니다.

We randomly initialize all new layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. All other layers (i.e., the shared convolutional layers) are initialized by pretraining a model for ImageNet classification [36], as is standard practice [5]. We tune all layers of the ZF net, and conv3 1 and up for the VGG net to conserve memory [2]. We use a learning rate of 0.001 for 60k mini-batches, and 0.0001 for the next 20k mini-batches on the PASCAL VOC dataset. We use a momentum of 0.9 and a weight decay of 0.0005 [37]. Our implementation uses Caffe [38].

표준 편차가 0.01인 0-평균 가우스 분포에서 가중치를 그려 모든 새 레이어를 랜덤하게 초기화합니다. 다른 모든 레이어(즉, 공유 컨볼루션 레이어)는 표준 프랙티스 [5]와 같이 ImageNet 분류에 대한 모델을 사전 교육하여 초기화됩니다. 우리는 메모리를 보존하기 위해 ZF 네트워크의 모든 계층과 VGG 네트워크를 위해 conv31 이상을 튜닝합니다 [2]. PASCAL VOC 데이터 집합의 60k 미니 배치에는 0.001의 학습률을 사용하고, 이후 20k 미니 배치에는 0.0001의 학습률을 사용합니다. 운동량은 0.9이고 무게는 0.0005[37]입니다. 우리의 구현은 카페[38]를 사용합니다.

**3.2 Sharing Features for RPN and Fast R-CNN**

Thus far we have described how to train a network for region proposal generation, without considering the region-based object detection CNN that will utilize these proposals. For the detection network, we adopt Fast R-CNN [2]. Next we describe algorithms that learn a unified network composed of RPN and Fast R-CNN with shared convolutional layers (Figure 2).

지금까지 우리는 이러한 제안을 활용할 지역 기반 객체 감지 CNN을 고려하지 않고 지역 제안 생성을 위한 네트워크를 교육하는 방법을 설명했습니다. 탐지 네트워크를 위해, 우리는 Fast R-CNN[2]을 채택합니다. 다음으로, 공유 컨볼루션 레이어가 있는 RPN 및 Fast R-CNN으로 구성된 유니파이드 네트워크를 학습하는 알고리즘을 설명합니다(그림 2).

Both RPN and Fast R-CNN, trained independently, will modify their convolutional layers in different ways. We therefore need to develop a technique that allows for sharing convolutional layers between the two networks, rather than learning two separate networks. We discuss three ways for training networks with features shared:

독립적으로 훈련된 RPN과 Fast R-CNN은 모두 다른 방식으로 컨볼루션 레이어를 수정합니다. 따라서 우리는 두 개의 별도 네트워크를 배우는 것보다 두 네트워크 간에 컨볼루션 레이어를 공유할 수 있는 기술을 개발해야 합니다. 우리는 기능이 공유된 네트워크를 교육하는 세 가지 방법에 대해 논의합니다.



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

표 1: ZF 망을 사용하여 각 앵커에 대해 학습된 평균 제안 크기(s = 600의 경우).

(i) Alternating training. In this solution, we first train RPN, and use the proposals to train Fast R-CNN. The network tuned by Fast R-CNN is then used to initialize RPN, and this process is iterated. This is the solution that is used in all experiments in this paper.

(i) 교대 교육. 이 솔루션에서는 먼저 RPN을 교육하고 제안서를 사용하여 Fast R-CNN을 교육합니다. 그런 다음 Fast R-CNN에 의해 튜닝된 네트워크를 사용하여 RPN을 초기화하고 이 프로세스를 반복합니다. 이것은 본 논문의 모든 실험에 사용되는 용액입니다.

(ii) Approximate joint training. In this solution, the RPN and Fast R-CNN networks are merged into one network during training as in Figure 2. In each SGD iteration, the forward pass generates region proposals which are treated just like fixed, pre-computed proposals when training a Fast R-CNN detector. The backward propagation takes place as usual, where for the shared layers the backward propagated signals from both the RPN loss and the Fast R-CNN loss are combined. This solution is easy to implement. But this solution ignores the derivative w.r.t. the proposal boxes’ coordinates that are also network responses, so is approximate. In our experiments, we have empirically found this solver produces close results, yet reduces the training time by about 25-50% comparing with alternating training. This solver is included in our released Python code.

(ii) 대략적인 합동 훈련 이 솔루션에서 RPN과 Fast R-CNN 네트워크는 그림 2와 같이 교육 중에 하나의 네트워크로 병합됩니다. 각 SGD 반복에서 전진 통과는 Fast R-CNN 검출기를 훈련할 때 고정된 사전 계산 제안과 동일하게 취급되는 지역 제안을 생성한다. 후방 전파는 공유 계층의 경우 RPN 손실과 고속 R-CNN 손실의 후방 전파 신호가 결합되는 일반적인 방식으로 수행됩니다. 이 솔루션은 구현하기 쉽습니다. 그러나 이 솔루션은 네트워크 응답이기도 한 제안 상자의 좌표를 무시하기 때문에 대략적입니다. 우리의 실험에서 우리는 이 해결사가 가까운 결과를 만들어내면서도 교대로 훈련하는 것에 비해 훈련 시간을 약 25-50% 단축시킨다는 것을 경험적으로 발견했습니다. 이 해결사는 우리가 출시한 Python 코드에 포함되어 있습니다.

(iii) Non-approximate joint training. As discussed above, the bounding boxes predicted by RPN are also functions of the input. The RoI pooling layer [2] in Fast R-CNN accepts the convolutional features and also the predicted bounding boxes as input, so a theoretically valid backpropagation solver should also involve gradients w.r.t. the box coordinates. These gradients are ignored in the above approximate joint training. In a non-approximate joint training solution, we need an RoI pooling layer that is differentiable w.r.t. the box coordinates. This is a nontrivial problem and a solution can be given by an “RoI warping” layer as developed in [15], which is beyond the scope of this paper.

(iii) 비 근사적 공동 훈련 위에서 설명한 것처럼 RPN이 예측한 경계 상자도 입력의 함수입니다. Fast R-CNN의 RoI 풀링 레이어 [2]는 컨볼루션 특성 및 예측 경계 상자를 입력으로 받아들이므로 이론적으로 유효한 백프로파전 해결사는 박스 좌표의 구배도 포함해야 한다. 위의 대략적인 공동 훈련에서는 이러한 구배가 무시됩니다. 대략적이지 않은 공동 교육 솔루션에서는 박스 좌표를 구분할 수 있는 RoI 풀링 레이어가 필요합니다. 이것은 사소한 문제가 아니며 [15]에서 개발한 것처럼 "RoI warping" 계층에 의해 해결책이 제시될 수 있습니다. 이는 본 논문의 범위를 벗어난 것입니다.

**4-Step Alternating Training.** In this paper, we adopt a pragmatic 4-step training algorithm to learn shared features via alternating optimization. In the first step, we train the RPN as described in Section 3.1.3. This network is initialized with an ImageNet-pre-trained model and fine-tuned end-to-end for the region proposal task. In the second step, we train a separate detection network by Fast R-CNN using the proposals generated by the step-1 RPN. This detection network is also initialized by the ImageNet-pre-trained model. At this point the two networks do not share convolutional layers. In the third step, we use the detector network to initialize RPN training, but we fix the shared convolutional layers and only fine-tune the layers unique to RPN. Now the two networks share convolutional layers. Finally, keeping the shared convolutional layers fixed, we fine-tune the unique layers of Fast R-CNN. As such, both networks share the same convolutional layers and form a unified network. A similar alternating training can be run for more iterations, but we have observed negligible improvements.

4단계 교대 교육. 본 문서에서는 실용적인 4단계 교육 알고리즘을 채택하여 대체 최적화를 통해 공유 기능을 학습합니다. 첫 번째 단계에서는 3.1.3절에 설명된 대로 RPN을 교육합니다. 이 네트워크는 ImageNet의 사전 교육 모델을 사용하여 초기화되며 지역 제안 작업에 맞게 엔드 투 엔드 미세 조정됩니다. 두 번째 단계에서는 1단계 RPN에서 생성된 제안을 사용하여 Fast R-CNN에 의한 별도의 탐지 네트워크를 교육합니다. 이 탐지 네트워크는 ImageNet의 사전 교육 모델에서도 초기화됩니다. 이 시점에서 두 네트워크는 컨볼루션 레이어를 공유하지 않습니다. 세 번째 단계에서는 검출기 네트워크를 사용하여 RPN 교육을 초기화하지만 공유 컨볼루션 레이어를 수정하고 RPN 고유의 레이어만 미세 조정합니다. 이제 두 네트워크는 컨볼루션 레이어를 공유합니다. 마지막으로 공유 콘볼루션 레이어를 고정시켜 Fast R-CNN 고유의 레이어를 미세 조정하여 두 네트워크 모두 동일한 콘볼루션 레이어를 공유하고 통합 네트워크를 형성합니다. 유사한 교대 교육을 더 반복할 수 있지만, 우리는 사소한 개선 사항을 관찰했습니다.

**3.3 Implementation Details**

We train and test both region proposal and object detection networks on images of a single scale [1], [2]. We re-scale the images such that their shorter side is s = 600 pixels [2]. Multi-scale feature extraction (using an image pyramid) may improve accuracy but does not exhibit a good speed-accuracy trade-off [2]. On the re-scaled images, the total stride for both ZF and VGG nets on the last convolutional layer is 16 pixels, and thus is ∼10 pixels on a typical PASCAL image before resizing (∼500×375). Even such a large stride provides good results, though accuracy may be further improved with a smaller stride.

우리는 단일 척도의 이미지[1], [2]에서 지역 제안과 객체 감지 네트워크 모두를 교육하고 테스트합니다. 영상의 더 짧은 면이 s = 600 픽셀이 되도록 영상의 스케일을 재조정합니다 [2]. 멀티스케일 피쳐 추출(이미지 피라미드 사용)은 정확도를 향상시킬 수 있지만 속도 정확도 균형이 잘 잡히지 않습니다 [2]. 재스케일링된 이미지에서 마지막 컨볼루션 레이어의 ZF 및 VGG 네트의 총 보폭은 16픽셀이므로 크기 조정 전 일반적인 PASCAL 이미지에서 최대 10픽셀입니다(~500×375). 이렇게 큰 보폭이라도 더 작은 보폭으로 정확도가 더 향상될 수 있지만 좋은 결과를 얻을 수 있습니다.

For anchors, we use 3 scales with box areas of 1282, 2562, and 5122 pixels, and 3 aspect ratios of 1:1, 1:2, and 2:1. These hyper-parameters are not carefully chosen for a particular dataset, and we provide ablation experiments on their effects in the next section. As discussed, our solution does not need an image pyramid or filter pyramid to predict regions of multiple scales, saving considerable running time. Figure 3 (right) shows the capability of our method for a wide range of scales and aspect ratios. Table 1 shows the learned average proposal size for each anchor using the ZFnet. We note that our algorithm allows predictions that are larger than the underlying receptive field. Such predictions are not impossible—one may still roughly infer the extent of an object if only the middle of the object is visible.

앵커의 경우 박스 면적 1282, 2562, 5122 픽셀의 3개의 스케일과 1:1, 1:2, 2:1의 3개의 가로 세로 비율을 사용합니다. 이러한 하이퍼 파라미터는 특정 데이터 세트에 대해 신중하게 선택되지 않으며, 다음 섹션에서 이러한 파라미터의 효과에 대한 절제 실험을 제공합니다. 논의한 바와 같이, NAT 솔루션은 여러 척도의 영역을 예측하기 위해 이미지 피라미드나 필터 피라미드가 필요하지 않으므로 실행 시간이 상당히 절약됩니다. 그림 3(오른쪽)은 광범위한 척도 및 가로 세로 비율에 대한 NAT 방법의 기능을 보여줍니다. 표 1은 ZFnet을 사용하여 각 앵커에 대해 학습된 평균 제안 크기를 보여줍니다. 우리는 우리의 알고리즘이 기본적인 수용 영역보다 더 큰 예측을 허용한다는 것에 주목합니다. 이러한 예측은 불가능하지 않다. 물체의 중간만 보일 경우 물체의 정도를 대략적으로 추론할 수 있다.

The anchor boxes that cross image boundaries need to be handled with care. During training, we ignore all cross-boundary anchors so they do not contribute to the loss. For a typical 1000 × 600 image, there will be roughly 20000 (≈ 60 × 40 × 9) anchors in total. With the cross-boundary anchors ignored, there are about 6000 anchors per image for training. If the boundary-crossing outliers are not ignored in training, they introduce large, difficult to correct error terms in the objective, and training does not converge. During testing, however, we still apply the fully convolutional RPN to the entire image. This may generate crossboundary proposal boxes, which we clip to the image boundary.

이미지 경계를 교차하는 앵커 박스는 주의하여 처리해야 합니다. 교육 중에는 모든 교차 경계 앵커를 무시하여 손실의 원인이 되지 않습니다. 일반적인 1000 × 600 이미지의 경우, 총 약 20000 (→ 60 × 40 × 9)개의 앵커가 있습니다. 크로스 바운더리 앵커가 무시된 상태에서, 이미지 당 약 6000개의 앵커가 교육을 위해 있습니다. 훈련에서 경계 교차 특이치를 무시하지 않으면 목표에서 큰 오차항을 수정하기 어렵고 훈련이 수렴되지 않습니다. 그러나 테스트하는 동안 전체 이미지에 완전한 컨볼루션 RPN을 적용합니다. 그러면 교차 경계 제안 상자가 생성되어 이미지 경계에 클리핑됩니다.

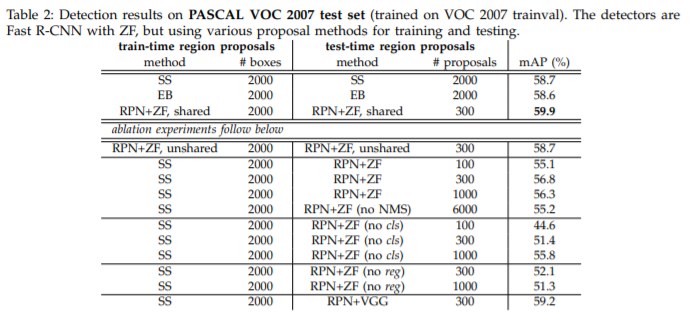


Table 2: Detection results on PASCAL VOC 2007 test set (trained on VOC 2007 trainval). The detectors are Fast R-CNN with ZF, but using various proposal methods for training and testing.

표 2: PASCAL VOC 2007 테스트 세트의 검출 결과(VOC 2007 trainval에 관한 트레이닝). 검출기는 ZF를 사용하는 고속 R-CNN이지만 교육 및 테스트에 다양한 제안 방법을 사용합니다.

Some RPN proposals highly overlap with each other. To reduce redundancy, we adopt non-maximum suppression (NMS) on the proposal regions based on their cls scores. We fix the IoU threshold for NMS at 0.7, which leaves us about 2000 proposal regions per image. As we will show, NMS does not harm the ultimate detection accuracy, but substantially reduces the number of proposals. After NMS, we use the top-N ranked proposal regions for detection. In the following, we train Fast R-CNN using 2000 RPN proposals, but evaluate different numbers of proposals at test-time.

일부 RPN 제안은 서로 매우 겹칩니다. 중복을 줄이기 위해, 우리는 제안서 영역에 대한 비최대 억제(NMS: Non-maximum suppression)를 채택합니다. NMS의 IoU 임계값을 0.7로 수정하여 이미지당 2000개 정도의 제안 영역을 남깁니다. 우리가 보여줄 것처럼, NMS는 궁극적인 검출 정확성을 해치지 않지만 제안 횟수를 상당히 줄여줍니다. NMS 이후, 우리는 상위 N순위 제안 영역을 탐지에 사용합니다. 다음에서는 2000 RPN 제안서를 사용하여 Fast R-CNN을 교육하지만, 테스트 시 제안 건수는 다르게 평가합니다.

**4 EXPERIMENTS**

**4.1 Experiments on PASCAL VOC**

We comprehensively evaluate our method on the PASCAL VOC 2007 detection benchmark [11]. This dataset consists of about 5k trainval images and 5k test images over 20 object categories. We also provide results on the PASCAL VOC 2012 benchmark for a few models. For the ImageNet pre-trained network, we use the “fast” version of ZF net [32] that has 5 convolutional layers and 3 fully-connected layers, and the public VGG-16 model7 [3] that has 13 convolutional layers and 3 fully-connected layers. We primarily evaluate detection mean Average Precision (mAP), because this is the actual metric for object detection (rather than focusing on object proposal proxy metrics).

우리는 PASCAL VOC 2007 검출 벤치마크에 대한 우리의 방법을 포괄적으로 평가한다[11]. 이 데이터 세트는 약 5k의 열차 이미지와 20개의 객체 범주에 걸친 5k의 테스트 이미지로 구성됩니다. PASCAL VOC 2012 벤치마크에 대한 결과도 제공합니다. ImageNet의 사전 교육 네트워크를 위해, 우리는 5개의 컨볼루션 레이어와 3개의 완전히 연결된 레이어를 가진 ZFnet[32]의 "빠른" 버전과 13개의 컨볼루션 레이어와 3개의 완전히 연결된 레이어를 가진 공용 VGG-16 모델 7[3]을 사용합니다. mAP(평균 정밀도)는 개체 제안 프록시 메트릭에 초점을 맞추는 것이 아니라 개체 감지를 위한 실제 메트릭이기 때문에 주로 탐지 평균을 평가합니다.

Table 2 (top) shows Fast R-CNN results when trained and tested using various region proposal methods. These results use the ZF net. For Selective Search (SS) [4], we generate about 2000 proposals by the “fast” mode. For EdgeBoxes (EB) [6], we generate the proposals by the default EB setting tuned for 0.7 IoU. SS has an mAP of 58.7% and EB has an mAP of 58.6% under the Fast R-CNN framework. RPN with Fast R-CNN achieves competitive results, with an mAP of 59.9% while using up to 300 proposals8. Using RPN yields a much faster detection system than using either SS or EB because of shared convolutional computations; the fewer proposals also reduce the region-wise fully-connected layers’ cost (Table 5).

표 2(상단)는 다양한 지역 제안 방법을 사용하여 교육 및 테스트했을 때의 빠른 R-CNN 결과를 보여줍니다. 이러한 결과는 ZF 네트워크를 사용합니다. 선택적 검색(SS) [4]의 경우 "빠른" 모드로 약 2000개의 제안을 생성합니다. EdgeBoxes (EB)의 경우 [6] 0.7 IoU로 튜닝된 기본 EB 설정으로 제안서를 생성합니다. SS는 58.7%의 mAP를 가지며, EB는 Fast R-CNN 프레임워크에서 58.6%의 mAP를 갖습니다. RPN(Fast R-CNN 포함)은 최대 300개의 제안서를 사용하면서 59.9%의 mAP로 경쟁력 있는 결과를 달성합니다8. RPN을 사용하면 SS 또는 EB를 사용하는 것보다 훨씬 빠른 탐지 시스템을 얻을 수 있습니다. 공유된 컨볼루션 계산으로 인해 더 적은 수의 제안이 지역별로 완전히 연결된 계층의 비용을 절감할 수 있습니다(표 5).

**Ablation Experiments on RPN.** To investigate the behavior of RPNs as a proposal method, we conducted several ablation studies. First, we show the effect of sharing convolutional layers between the RPN and Fast R-CNN detection network. To do this, we stop after the second step in the 4-step training process. Using separate networks reduces the result slightly to 58.7% (RPN+ZF, unshared, Table 2). We observe that this is because in the third step when the detectortuned features are used to fine-tune the RPN, the proposal quality is improved.

RPN에 대한 절제 실험. 제안 방법으로서 RPN의 행동을 조사하기 위해, 우리는 몇 가지 절제 연구를 수행했다. 먼저, RPN과 Fast R-CNN 검출 네트워크 간의 컨볼루션 레이어 공유 효과를 보여줍니다. 이를 위해 우리는 4단계 교육 과정의 두 번째 단계를 마치고 중단합니다. 별도의 네트워크를 사용하면 결과가 58.7%로 약간 감소합니다(RPN+ZF, 공유되지 않음, 표 2). 세 번째 단계에서 RPN 미세 조정을 위해 검출된 기능을 사용할 때 제안 품질이 향상되기 때문인 것으로 관찰됩니다.

Next, we disentangle the RPN’s influence on training the Fast RCNN detection network. For this purpose, we train a Fast R-CNN model by using the 2000 SS proposals and ZF net. We fix this detector and evaluate the detection mAP by changing the proposal regions used at test-time. In these ablation experiments, the RPN does not share features with the detector.

다음으로, Fast RCN 탐지 네트워크 교육에 대한 RPN의 영향력을 제거합니다. 이를 위해 2000 SS 제안서와 ZF net를 사용하여 Fast R-CNN 모델을 교육합니다. 우리는 이 검출기를 고치고 테스트 시간에 사용되는 제안 영역을 변경하여 검출 mAP를 평가합니다. 이러한 절제 실험에서 RPN은 디텍터와 기능을 공유하지 않습니다.

Replacing SS with 300 RPN proposals at test-time leads to an mAP of 56.8%. The loss in mAP is because of the inconsistency between the training/testing proposals. This result serves as the baseline for the following comparisons.

SS를 테스트 시점에 300 RPN 제안으로 대체하면 56.8%의 mAP를 얻을 수 있습니다. mAP의 손실은 교육/테스트 제안 간의 불일치 때문입니다. 이 결과는 다음과 같은 비교의 기준이 됩니다.

Somewhat surprisingly, the RPN still leads to a competitive result (55.1%) when using the top-ranked 100 proposals at test-time, indicating that the topranked RPN proposals are accurate. On the other extreme, using the top-ranked 6000 RPN proposals (without NMS) has a comparable mAP (55.2%), suggesting NMS does not harm the detection mAP and may reduce false alarms.

놀랍게도, RPN은 여전히 테스트 시점에 상위 100개 제안서를 사용할 때 경쟁 결과(55.1%)로 이어져 상위 RPN 제안이 정확하다는 것을 알 수 있다. 반면, 최상위 6000 RPN 제안(NMS 미포함)을 사용하면 유사한 mAP(55.2%)가 있어 NMS가 탐지 mAP를 해치지 않으며 잘못된 경보를 줄일 수 있음을 시사한다.

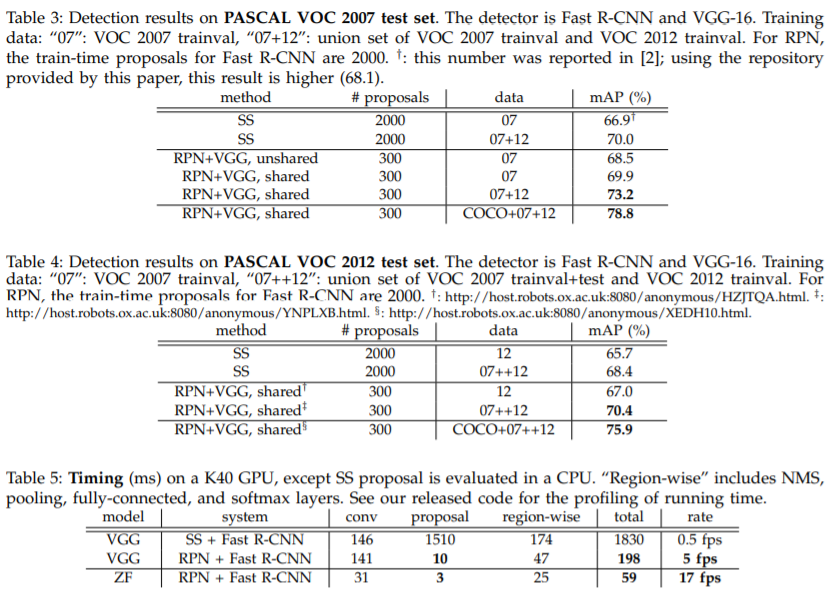


Table 3: Detection results on PASCAL VOC 2007 test set. The detector is Fast R-CNN and VGG-16. Training data: “07”: VOC 2007 trainval, “07+12”: union set of VOC 2007 trainval and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. † : this number was reported in [2]; using the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

표 3: PASCAL VOC 2007 테스트 세트의 검출 결과 검출기는 Fast R-CNN 및 VGG-16입니다. 훈련 데이터: "07": VOC 2007 열차, "07+12": VOC 2007 열차와 VOC 2012 열차의 조합 세트. RPN의 경우, Fast R-CNN의 열차 시간 제안은 2000입니다. † : 이 수치는 [2]에 기재되어 있습니다.이 문서에서 제공하는 저장소를 사용하면 이 결과가 더 높아집니다(68.1).

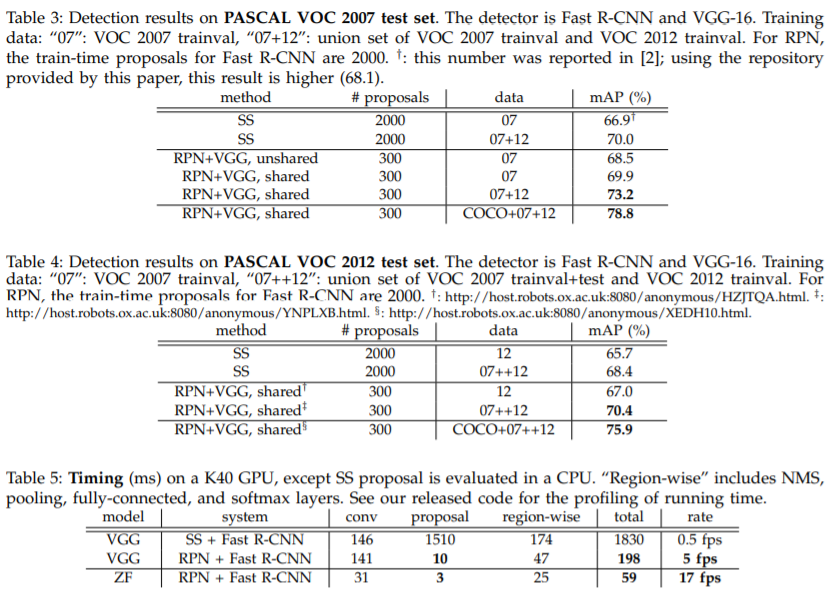


Table 4: Detection results on PASCAL VOC 2012 test set. The detector is Fast R-CNN and VGG-16. Training data: “07”: VOC 2007 trainval, “07++12”: union set of VOC 2007 trainval+test and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000.

표 4: PASCAL VOC 2012 테스트 세트의 검출 결과 검출기는 Fast R-CNN 및 VGG-16입니다. 트레이닝 데이터: "07": VOC 2007 훈련, "07++12": VOC 2007 trainval+test와 VOC 2012 trainval의 조합 세트. RPN의 경우, Fast R-CNN의 열차 시간 제안은 2000입니다.

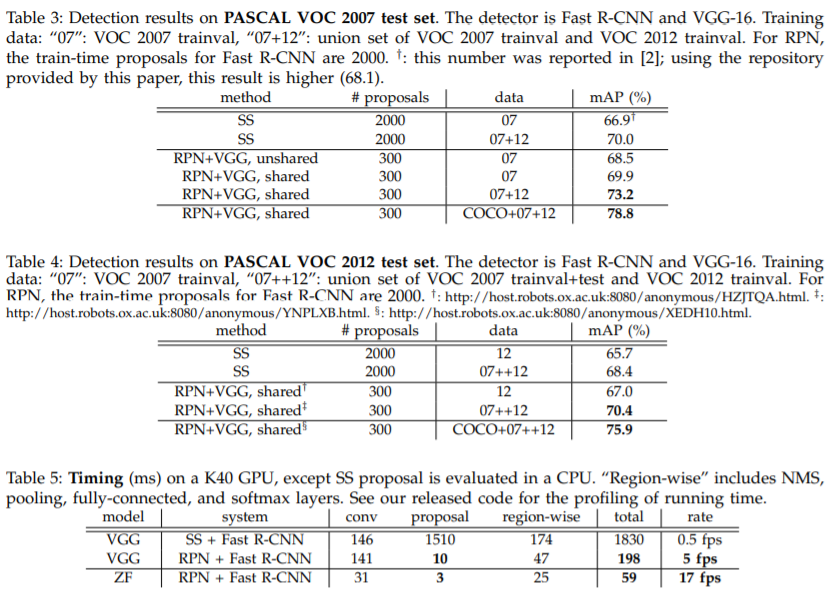


Table 5: Timing (ms) on a K40 GPU, except SS proposal is evaluated in a CPU. “Region-wise” includes NMS, pooling, fully-connected, and softmax layers. See our released code for the profiling of running time.

표 5: K40 GPU에서의 타이밍(ms). 단, SS 프로포절은 CPU로 평가됩니다.「지역별」에는, NMS, 풀링, 완전 접속, 및 softmax 레이어가 포함됩니다. 실행 시간에 대한 프로파일링은 공개된 코드를 참조하십시오.

Next, we separately investigate the roles of RPN’s cls and reg outputs by turning off either of them at test-time. When the cls layer is removed at testtime (thus no NMS/ranking is used), we randomly sample N proposals from the unscored regions. The mAP is nearly unchanged with N = 1000 (55.8%), but degrades considerably to 44.6% when N = 100. This shows that the cls scores account for the accuracy of the highest ranked proposals.

다음으로, 테스트 시 RPN의 cls와 reg 출력 중 하나를 꺼서 역할을 별도로 조사합니다. 테스트 시 cls 레이어를 제거하면(따라서 NMS/순위 지정이 사용되지 않음), 점수가 없는 영역에서 N개의 제안을 랜덤하게 추출합니다. mAP는 N = 1000(55.8%)으로 거의 변경되지 않지만 N = 100일 경우 44.6%로 크게 감소합니다. 이것은 cls 점수가 가장 높은 순위의 제안의 정확성을 설명하고 있음을 보여준다.

On the other hand, when the reg layer is removed at test-time (so the proposals become anchor boxes), the mAP drops to 52.1%. This suggests that the highquality proposals are mainly due to the regressed box bounds. The anchor boxes, though having multiple scales and aspect ratios, are not sufficient for accurate detection.

반면, 테스트 시점에 레지스트리 레이어를 제거하면(따라서 제안서가 앵커 박스가 됨) mAP는 52.1%로 떨어집니다. 따라서 고품질 제안이 주로 회귀된 상자 경계에 기인한다는 것을 알 수 있습니다. 앵커 박스는 척도와 가로 세로 비율이 여러 개이지만 정확한 탐지에 충분하지 않습니다.

We also evaluate the effects of more powerful networks on the proposal quality of RPN alone. We use VGG-16 to train the RPN, and still use the above detector of SS+ZF. The mAP improves from 56.8% (using RPN+ZF) to 59.2% (using RPN+VGG). This is a promising result, because it suggests that the proposal quality of RPN+VGG is better than that of RPN+ZF. Because proposals of RPN+ZF are competitive with SS (both are 58.7% when consistently used for training and testing), we may expect RPN+VGG to be better than SS. The following experiments justify this hypothesis.

우리는 또한 보다 강력한 네트워크가 RPN의 제안 품질에 미치는 영향만을 평가합니다. 우리는 RPN을 훈련하기 위해 VGG-16을 사용하지만 여전히 위의 SS+ZF 검출기를 사용합니다. mAP는 56.8%(RPN+ZF 사용)에서 59.2%(RPN+VGG 사용)로 개선됩니다. 이는 RPN+VGG의 제안품질이 RPN+ZF보다 우수하다는 것을 시사하기 때문에 유망한 결과이다. RPN+ZF의 제안은 SS와 경쟁력이 있기 때문에(훈련과 테스트에 지속적으로 사용될 경우 둘 다 58.7%임), RPN+VGG가 SS보다 나을 것으로 예상할 수 있다. 다음 실험들이 이 가설을 정당화한다.

**Performance of VGG-16.** Table 3 shows the results of VGG-16 for both proposal and detection. Using RPN+VGG, the result is 68.5% for unshared features, slightly higher than the SS baseline. As shown above, this is because the proposals generated by RPN+VGG are more accurate than SS. Unlike SS that is predefined, the RPN is actively trained and benefits from better networks. For the feature-shared variant, the result is 69.9%—better than the strong SS baseline, yet with nearly cost-free proposals. We further train the RPN and detection network on the union set of PASCAL VOC 2007 trainval and 2012 trainval. The mAP is 73.2%. Figure 5 shows some results on the PASCAL VOC 2007 test set. On the PASCAL VOC 2012 test set (Table 4), our method has an mAP of 70.4% trained on the union set of VOC 2007 trainval+test and VOC 2012 trainval. Table 6 and Table 7 show the detailed numbers.

VGG-16의 성능. 표 3은 제안과 탐지에 대한 VGG-16의 결과를 보여준다. RPN+VGG를 사용하면 공유되지 않은 기능의 경우 68.5%로 SS 기준보다 약간 높습니다. 위와 같이 RPN+VGG에서 생성된 제안서가 SS보다 정확하기 때문이다. 미리 정의된 SS와 달리 RPN은 적극적으로 교육되어 있으며 더 나은 네트워크를 통해 이점을 얻을 수 있습니다. 기능 공유 모델의 경우 69.9%의 결과가 나와 강력한 SS 기준보다 우수하면서도 거의 비용 부담 없는 제안서를 제시합니다. PASCAL VOC 2007 열차 발사와 2012 열차 발사의 조합 세트에 대한 RPN 및 검출 네트워크를 추가로 교육합니다. mAP는 73.2%입니다. 그림 5는 PASCAL VOC 2007 테스트 세트에 대한 일부 결과를 보여줍니다. PASCAL VOC 2012 테스트 세트(표 4)에서, 우리의 방법은 VOC 2007 열차 발+ 테스트와 VOC 2012 열차 발 연합 세트에 대해 70.4%의 mAP를 훈련하였습니다. 표 6과 표 7은 상세한 수치를 보여준다.

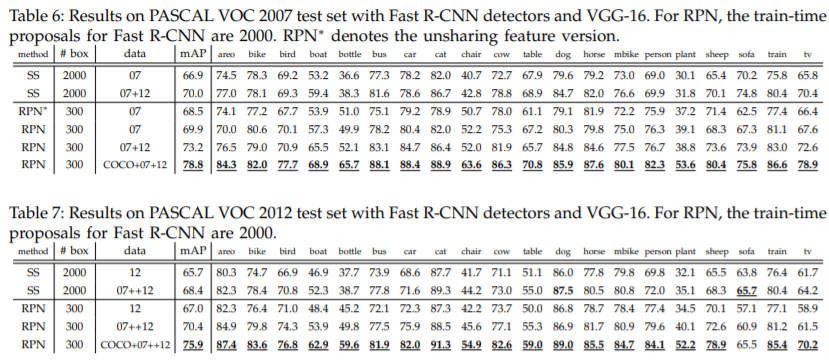


Table 6: Results on PASCAL VOC 2007 test set with Fast R-CNN detectors and VGG-16. For RPN, the train-time proposals for Fast R-CNN are 2000. RPN∗ denotes the unsharing feature version.

표 6: Fast R-CNN 검출기와 VGG-16을 사용한PASCAL VOC 2007 테스트 세트의 결과 RPN의 경우, Fast R-CNN의 열차 시간 제안은 2000입니다. RPN'은 공유 해제 기능 버전을 나타냅니다.

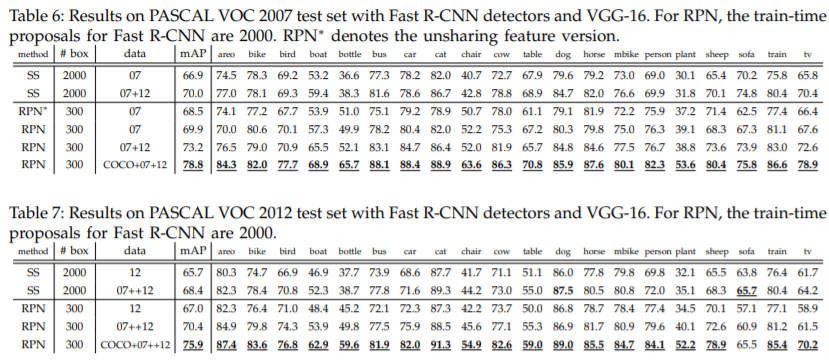


Table 7: Results on PASCAL VOC 2012 test set with Fast R-CNN detectors and VGG-16. For RPN, the train-time proposals for Fast R-CNN are 2000.

표 7: 고속 R-CNN 검출기와 VGG-16을 사용한 PASCAL VOC 2012 테스트 세트의 결과. RPN의 경우, Fast R-CNN의 열차 시간 제안은 2000입니다.

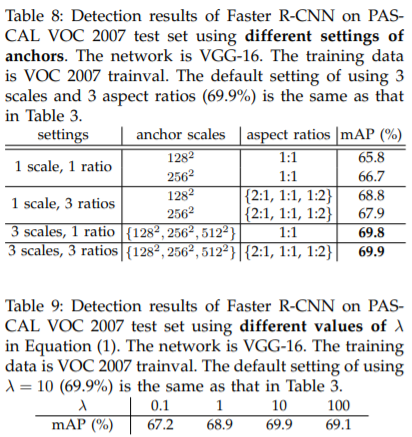


Table 8: Detection results of Faster R-CNN on PASCAL VOC 2007 test set using different settings of anchors. The network is VGG-16. The training data is VOC 2007 trainval. The default setting of using 3 scales and 3 aspect ratios (69.9%) is the same as that in Table 3.

표 8: 다양한 앵커 설정을 사용한PASCAL VOC 2007 테스트 세트에서의 고속 R-CNN 검출 결과 네트워크는 VGG-16입니다. 훈련 데이터는 VOC 2007 훈련 데이터입니다. 3 스케일 및 3 종횡비(69.9%)를 사용하는 기본 설정은 표 3과 동일합니다.

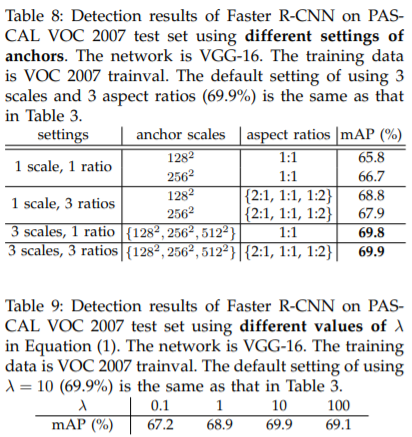


Table 9: Detection results of Faster R-CNN on PASCAL VOC 2007 test set using different values of λ in Equation (1). The network is VGG-16. The training data is VOC 2007 trainval. The default setting of using λ = 10 (69.9%) is the same as that in Table 3.

표 9: 식 (1)의 다른 θ 값을 사용한 PASCAL VOC 2007 테스트 세트의 고속 R-CNN 검출 결과. 네트워크는 VGG-16입니다. 훈련 데이터는 VOC 2007 훈련 데이터입니다. § = 10(69.9%)을 사용하는 기본 설정은 표 3과 동일하다.

In Table 5 we summarize the running time of the entire object detection system. SS takes 1-2 seconds depending on content (on average about 1.5s), and Fast R-CNN with VGG-16 takes 320ms on 2000 SS proposals (or 223ms if using SVD on fully-connected layers [2]). Our system with VGG-16 takes in total 198ms for both proposal and detection. With the convolutional features shared, the RPN alone only takes 10ms computing the additional layers. Our regionwise computation is also lower, thanks to fewer proposals (300 per image). Our system has a frame-rate of 17 fps with the ZF net.

표 5에서는 전체 객체 감지 시스템의 실행 시간을 요약합니다. SS는 콘텐츠에 따라 1-2초(평균 약 1.5초)가 소요되며, VGG-16이 탑재된 Fast R-CNN은 2000 SS 제안 시 320ms(또는 완전히 연결된 계층에서 SVD를 사용하는 경우 223ms)가 소요됩니다[2]. 우리 VGG-16 시스템은 제안과 탐지에 총 198ms가 소요됩니다. 컨볼루션 기능을 공유한 경우 RPN만 해도 추가 레이어를 계산하는 데 10ms밖에 걸리지 않습니다. 제안 건수가 적어 지역별 계산도 낮아졌습니다(이미지당 300건). 우리 시스템은 17fps의 프레임률과 ZF 망을 가지고 있습니다.

Sensitivities to Hyper-parameters. In Table 8 we investigate the settings of anchors. By default we use 3 scales and 3 aspect ratios (69.9% mAP in Table 8). If using just one anchor at each position, the mAP drops by a considerable margin of 3-4%. The mAP is higher if using 3 scales (with 1 aspect ratio) or 3 aspect ratios (with 1 scale), demonstrating that using anchors of multiple sizes as the regression references is an effective solution. Using just 3 scales with 1 aspect ratio (69.8%) is as good as using 3 scales with 3 aspect ratios on this dataset, suggesting that scales and aspect ratios are not disentangled dimensions for the detection accuracy. But we still adopt these two dimensions in our designs to keep our system flexible.

하이퍼 파라미터에 대한 민감도입니다. 표 8에서는 앵커의 설정을 조사합니다. 기본적으로 척도 3개와 가로 세로 비율 3개를 사용합니다(표 8의 69.9% mAP). 각 위치에서 하나의 앵커만 사용할 경우 mAP는 3~4%의 상당한 차이로 떨어집니다. mAP는 3개의 척도(1개의 가로 세로 비율) 또는 3개의 가로 세로 비율(1개의 척도)을 사용하는 경우 더 높으며, 회귀 참조로 여러 크기의 앵커를 사용하는 것이 효과적인 솔루션임을 보여 줍니다. 1 가로 세로 비율(69.8%)로 3개의 축척만 사용하는 것이 이 데이터 집합에서 3개의 가로 세로 비율로 3개의 축척을 사용하는 것과 같으므로 검출 정확도에 대해 축척과 가로 세로 비율이 분리되지 않습니다. 하지만 우리는 여전히 우리의 시스템을 유연하게 유지하기 위해 이 두 가지 차원을 디자인에 채택하고 있습니다.

In Table 9 we compare different values of λ in Equation (1). By default we use λ = 10 which makes the two terms in Equation (1) roughly equally weighted after normalization. Table 9 shows that our result is impacted just marginally (by ∼ 1%) when λ is within a scale of about two orders of magnitude (1 to 100). This demonstrates that the result is insensitive to λ in a wide range.

표 9에서는 등식 (1)에서 서로 다른 λ 값을 비교합니다. 기본적으로 λ = 10을 사용하여 정규화 후 방정식 (1)의 두 항에 거의 동일한 가중치를 부여합니다. 표 9는 λ이 약 2차 규모(1~100) 내에 있을 때 결과에 약간의 영향(~1%)이 미치는 것을 보여준다. 이는 결과가 넓은 범위에서 λ에 민감하지 않음을 나타냅니다.

Analysis of Recall-to-IoU. Next we compute the recall of proposals at different IoU ratios with groundtruth boxes. It is noteworthy that the Recall-to-IoU metric is just loosely [19], [20], [21] related to the ultimate detection accuracy. It is more appropriate to use this metric to diagnose the proposal method than to evaluate it.

리콜 대 IoU 분석. 다음으로 우리는 지면 진실 상자를 사용하여 서로 다른 IoU 비율로 제안 리콜을 계산합니다. Recall-to-IoU 메트릭은 최종 검출 정확도와 관련성이 거의 없는[19],[20], [21]에 불과합니다. 제안 방법을 평가하는 것보다 진단하는 데 이 메트릭을 사용하는 것이 더 적절합니다.

In Figure 4, we show the results of using 300, 1000, and 2000 proposals. We compare with SS and EB, and the N proposals are the top-N ranked ones based on the confidence generated by these methods. The plots show that the RPN method behaves gracefully when the number of proposals drops from 2000 to 300. This explains why the RPN has a good ultimate detection mAP when using as few as 300 proposals. As we analyzed before, this property is mainly attributed to the cls term of the RPN. The recall of SS and EB drops more quickly than RPN when the proposals are fewer.

그림 4에서는 300, 1000, 2000 제안서를 사용한 결과를 보여줍니다. SS 및 EB와 비교하며, 이러한 방법으로 창출된 신뢰도를 바탕으로 N개 제안은 상위 N개 제안입니다. 그림은 제안 수가 2000개에서 300개로 감소하면 RPN 방법이 정상적으로 동작함을 보여 줍니다. 따라서 RPN이 300개 미만의 제안서를 사용할 때 최적의 최종 탐지 맵을 갖는 이유가 설명됩니다. 앞에서 분석했듯이, 이 속성은 주로 RPN의 cls 기간에 기인합니다. SS와 EB의 리콜은 제안이 적을 때 RPN보다 더 빠르게 감소합니다.

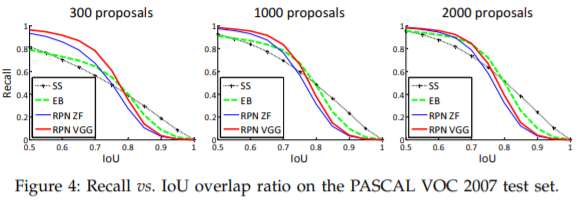


Figure 4: Recall vs. IoU overlap ratio on the PASCAL VOC 2007 test set

그림 4: 리콜과 PASCAL VOC 2007 테스트 세트의 IoU 오버랩비

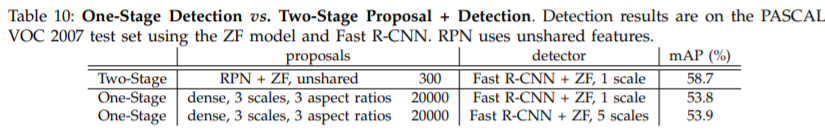


Table 10: One-Stage Detection vs. Two-Stage Proposal + Detection. Detection results are on the PASCAL VOC 2007 test set using the ZF model and Fast R-CNN. RPN uses unshared features.

표 10: 1단계 검출과 2단계 제안 + 검출. 검출 결과는 ZF 모델과 Fast R-CNN을 사용한PASCAL VOC 2007 테스트세트에 기재되어 있습니다.RPN은 비공유 기능을 사용합니다.

**One-Stage Detection vs. Two-Stage Proposal + Detection.** The OverFeat paper [9] proposes a detection method that uses regressors and classifiers on sliding windows over convolutional feature maps. OverFeat is a one-stage, class-specific detection pipeline, and ours is a two-stage cascade consisting of class-agnostic proposals and class-specific detections. In OverFeat, the region-wise features come from a sliding window of one aspect ratio over a scale pyramid. These features are used to simultaneously determine the location and category of objects. In RPN, the features are from square (3×3) sliding windows and predict proposals relative to anchors with different scales and aspect ratios. Though both methods use sliding windows, the region proposal task is only the first stage of Faster RCNN—the downstream Fast R-CNN detector attends to the proposals to refine them. In the second stage of our cascade, the region-wise features are adaptively pooled [1], [2] from proposal boxes that more faithfully cover the features of the regions. We believe these features lead to more accurate detections.

1단계 탐지 대 2단계 제안 + 탐지 OverFeat 논문[9]은 컨볼루션 피쳐 맵을 통해 슬라이딩 윈도우에 회귀기와 분류기를 사용하는 검출 방법을 제안합니다. OverFeat는 클래스별 1단계 탐지 파이프라인이며, 클래스별 제안과 클래스별 탐지로 구성된 2단계 계단식입니다. OverFeat에서 영역별 피쳐는 축척 피라미드 위의 한 가로 세로 비율의 슬라이딩 윈도우에서 가져옵니다. 이러한 기능은 객체의 위치와 범주를 동시에 결정하는 데 사용됩니다. RPN에서 기능은 사각형(3×3) 슬라이딩 윈도우에서 제공되며 척도 및 가로 세로 비율이 서로 다른 앵커와 관련된 제안을 예측합니다. 두 방법 모두 슬라이딩 창을 사용하지만 영역 제안 작업은 Faster RNN의 첫 번째 단계일 뿐입니다. 즉, 다운스트림 Fast R-CNN 디텍터가 제안 내용을 구체화합니다. 계단식의 두 번째 단계에서는 지역별 특성이 지역 특성을 보다 충실하게 다루는 제안 상자에서 [1], [2] 적응적으로 통합된다. 이러한 기능은 보다 정확한 탐지로 이어집니다.

To compare the one-stage and two-stage systems, we emulate the OverFeat system (and thus also circumvent other differences of implementation details) by one-stage Fast R-CNN. In this system, the “proposals” are dense sliding windows of 3 scales (128, 256, 512) and 3 aspect ratios (1:1, 1:2, 2:1). Fast R-CNN is trained to predict class-specific scores and regress box locations from these sliding windows. Because the OverFeat system adopts an image pyramid, we also evaluate using convolutional features extracted from 5 scales. We use those 5 scales as in [1], [2].

1단계와 2단계 시스템을 비교하기 위해 1단계 Fast R-CNN으로 OverFeat 시스템을 에뮬레이트하여 구현 세부사항의 다른 차이점을 회피하며, 이 시스템에서 "제안"은 3단계(128, 256, 512)와 3단계(1:1, 1:2, 2:1)의 조밀한 슬라이딩 윈도우입니다. Fast R-CNN은 이러한 슬라이딩 윈도우에서 클래스별 점수를 예측하고 상자 위치를 회귀시키도록 교육됩니다. OverFeat 시스템은 이미지 피라미드를 채택하기 때문에 5개의 척도에서 추출한 컨볼루션 피쳐를 사용하여 평가합니다. [1], [2]와 같이 5가지 척도를 사용합니다.

Table 10 compares the two-stage system and two variants of the one-stage system. Using the ZF model, the one-stage system has an mAP of 53.9%. This is lower than the two-stage system (58.7%) by 4.8%. This experiment justifies the effectiveness of cascaded region proposals and object detection. Similar observations are reported in [2], [39], where replacing SS region proposals with sliding windows leads to ∼6% degradation in both papers. We also note that the onestage system is slower as it has considerably more proposals to process.

표 10은 2단계 시스템과 1단계 시스템의 두 변형을 비교합니다. ZF 모델을 사용하면 1단계 시스템의 mAP는 53.9%입니다. 이는 2단계 시스템(58.7%)보다 4.8% 낮은 수준이다. 이 실험은 계단식 지역 제안과 객체 탐지의 효과를 정당화한다. [2], [39]에도 유사한 관찰 결과가 보고되어 있는데, SS 지역 제안을 슬라이딩 윈도우로 대체하면 두 논문에서 모두 ~6%의 저하가 초래된다. 또한 1단계 시스템은 처리해야 할 제안사항이 상당히 많기 때문에 속도가 느립니다.

**4.2 Experiments on MS COCO**

We present more results on the Microsoft COCO object detection dataset [12]. This dataset involves 80 object categories. We experiment with the 80k images on the training set, 40k images on the validation set, and 20k images on the test-dev set. We evaluate the mAP averaged for IoU ∈ [0.5 : 0.05 : 0.95] (COCO’s standard metric, simply denoted as mAP@[.5, .95]) and mAP@0.5 (PASCAL VOC’s metric).

Microsoft COCO 개체 감지 데이터셋에 대한 추가 결과를 제공합니다[12]. 이 데이터 집합에는 80개의 개체 범주가 포함됩니다. 교육 세트의 80,000개 이미지, 검증 세트의 40,000개 이미지, 테스트 개발 세트의 20,000개 이미지로 실험합니다. IoU → [0.5 : 0.05 : 0.95](COCO의 표준 메트릭, mAP@[.5,.95]로 표시됨) 및 mAP@0.5(PASCAL VOC의 메트릭)에 대한 mAP 평균을 평가한다.

There are a few minor changes of our system made for this dataset. We train our models on an 8-GPU implementation, and the effective mini-batch size becomes 8 for RPN (1 per GPU) and 16 for Fast R-CNN (2 per GPU). The RPN step and Fast R-CNN step are both trained for 240k iterations with a learning rate of 0.003 and then for 80k iterations with 0.0003. We modify the learning rates (starting with 0.003 instead of 0.001) because the mini-batch size is changed. For the anchors, we use 3 aspect ratios and 4 scales (adding 642), mainly motivated by handling small objects on this dataset. In addition, in our Fast R-CNN step, the negative samples are defined as those with a maximum IoU with ground truth in the interval of [0, 0.5), instead of [0.1, 0.5) used in [1], [2]. We note that in the SPPnet system [1], the negative samples in [0.1, 0.5) are used for network fine-tuning, but the negative samples in [0, 0.5) are still visited in the SVM step with hard-negative mining. But the Fast R-CNN system [2] abandons the SVM step, so the negative samples in [0, 0.1) are never visited. Including these [0, 0.1) samples improves mAP@0.5 on the COCO dataset for both Fast R-CNN and Faster R-CNN systems (but the impact is negligible on PASCAL VOC).

이 데이터 세트에 대해 변경된 시스템이 몇 가지 있습니다. 당사는 8-GPU 구현에 대한 모델을 교육하고 있으며, 유효 미니 배치 크기는 RPN(GPU당 1개)의 경우 8개, Fast R-CNN(GPU당 2개)의 경우 16개가 됩니다. RPN 단계와 Fast R-CNN 단계는 모두 0.003의 학습률로 240k 반복에 대해 교육된 후 0.0003의 80k 반복에 대해 교육됩니다. 미니배치 크기가 변경되어 학습율(0.001이 아닌 0.003으로 시작)을 수정합니다. 앵커의 경우 3개의 가로 세로 비율과 4개의 축척(642 추가)을 사용하며, 주로 이 데이터 집합의 작은 객체를 처리하는 데 동기를 부여합니다. 또한, Fast R-CNN 단계에서 음성 샘플은 [1], [2]에서 사용된 [0.1, 0.5) 대신 [0, 0.5] 간격의 접지 진리를 가진 최대 IoU를 가진 샘플로 정의됩니다. SPPnet 시스템 [1]에서 네트워크 미세조정에 [0.1, 0.5)의 음성 샘플이 사용되지만, SVM 단계에서 하드 음성 마이닝으로 음성 샘플이 계속 방문됩니다. 그러나 Fast R-CNN 시스템 [2]은(는) SVM 단계를 포기하므로 [0, 0.1]의 음수 샘플은 방문한 적이 없습니다. 이러한 [0, 0.1] 샘플을 포함하면 COCO 데이터 집합의 mAP@0.5가 Fast R-CNN 및 Faster R-CNN 시스템 모두에서 향상됩니다(PASCAL VOC에는 영향이 미미함).

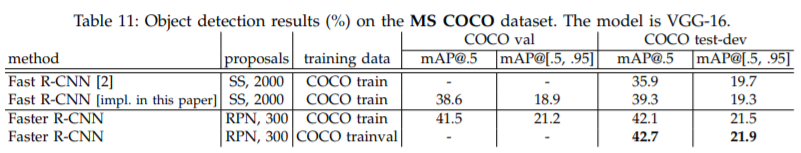


Table 11: Object detection results (%) on the MS COCO dataset. The model is VGG-16.

표 11: MS COCO 데이터 세트의 객체 검출 결과(%) 모델은 VGG-16입니다.

The rest of the implementation details are the same as on PASCAL VOC. In particular, we keep using 300 proposals and single-scale (s = 600) testing. The testing time is still about 200ms per image on the COCO dataset.

나머지 구현 세부 정보는 PASCAL VOC와 동일합니다. 특히 300개의 제안서와 단일 규모(s = 600) 테스트를 계속 사용하고 있습니다. 테스트 시간은 여전히 COCO 데이터셋의 이미지당 약 200ms입니다.

In Table 11 we first report the results of the Fast R-CNN system [2] using the implementation in this paper. Our Fast R-CNN baseline has 39.3% mAP@0.5 on the test-dev set, higher than that reported in [2]. We conjecture that the reason for this gap is mainly due to the definition of the negative samples and also the changes of the mini-batch sizes. We also note that the mAP@[.5, .95] is just comparable.

표 11에서는 먼저 본 문서의 구현을 사용하여 Fast R-CNN 시스템 [2]의 결과를 보고합니다. 당사의 Fast R-CNN 기준선은 테스트-개발 세트에서 [3]에 보고된 것보다 높은 39.3% mAP@0.5를 가지고 있습니다. 이러한 격차의 원인은 주로 음수 검체의 정의와 미니 배치 크기의 변화 때문일 것으로 추측됩니다. 또한 mAP@[.5,.95]도 비교할 수 있습니다.

Next we evaluate our Faster R-CNN system. Using the COCO training set to train, Faster R-CNN has 42.1% mAP@0.5 and 21.5% mAP@[.5, .95] on the COCO test-dev set. This is 2.8% higher for mAP@0.5 and 2.2% higher for mAP@[.5, .95] than the Fast RCNN counterpart under the same protocol (Table 11). This indicates that RPN performs excellent for improving the localization accuracy at higher IoU thresholds. Using the COCO trainval set to train, Faster RCNN has 42.7% mAP@0.5 and 21.9% mAP@[.5, .95] on the COCO test-dev set. Figure 6 shows some results on the MS COCO test-dev set.

다음으로 Faster R-CNN 시스템을 평가합니다. COCO 훈련 세트를 사용하여 Faster R-CNN은 COCO 테스트 개발 세트에서 mAP@0.5 및 21.5% mAP@[.5,.95]를 획득합니다. 이는 mAP@0.5의 경우 2.8%, mAP@[.5,.95]의 경우 동일한 프로토콜의 Fast RCN보다 2.2% 더 높습니다(표 11). 이는 RPN이 높은 IoU 임계값에서 지역화 정확도를 향상시키는 데 탁월하다는 것을 나타냅니다. 훈련하기 위해 COCO 열차 발판을 사용하는 Faster RCN은 COCO 테스트 개발 세트에서 mAP@0.5 및 21.9% mAP@[.5,.95]를 가집니다. 그림 6은 MS COCO 테스트 개발 세트에 대한 몇 가지 결과를 보여줍니다.

**Faster R-CNN in ILSVRC & COCO 2015 competitions** We have demonstrated that Faster R-CNN benefits more from better features, thanks to the fact that the RPN completely learns to propose regions by neural networks. This observation is still valid even when one increases the depth substantially to over 100 layers [18]. Only by replacing VGG-16 with a 101-layer residual net (ResNet-101) [18], the Faster R-CNN system increases the mAP from 41.5%/21.2% (VGG16) to 48.4%/27.2% (ResNet-101) on the COCO val set. With other improvements orthogonal to Faster RCNN, He et al. [18] obtained a single-model result of 55.7%/34.9% and an ensemble result of 59.0%/37.4% on the COCO test-dev set, which won the 1st place in the COCO 2015 object detection competition. The same system [18] also won the 1st place in the ILSVRC 2015 object detection competition, surpassing the second place by absolute 8.5%. RPN is also a building block of the 1st-place winning entries in ILSVRC 2015 localization and COCO 2015 segmentation competitions, for which the details are available in [18] and [15] respectively.

ILSVRC & COCO 2015 대회에서 R-CNN 속도 향상 RPN이 신경망에 의해 지역을 제안하는 방법을 완전히 학습한 덕분에 R-CNN 속도가 더 나은 기능의 혜택을 더 많이 받는다는 것을 입증했습니다. 이러한 관찰은 100개 이상의 층으로 깊이를 상당히 증가시키는 경우에도 여전히 유효하다[18]. 오직 VGG-16을 101층 잔류망(ResNet-101)[18]으로 대체함으로써, 고속 R-CNN 시스템은 COCO 밸브 세트의 mAP를 41.5%/21.2%(VGG16)에서 48.4%/27.2%(ResNet-101)로 증가시킵니다. 더 빠른 RCN과 직결되는 다른 개선사항과 함께, He 등. [18] COCO 2015 객체 감지 대회에서 1위를 차지한 COCO 테스트-개발 세트에서 55.7%/34.9%의 단일 모델 결과와 59.0%/37.4%의 앙상블 결과를 얻었습니다. 같은 시스템[18]도 ILSVRC 2015 객체 감지 대회에서 1위를 차지하여 절대 8.5% 차이로 2위를 앞질렀다. 또한 RPN은 ILSVRC 2015 현지화 및 COCO 2015 세분화 경쟁에서 1위를 차지한 참가자의 구성 요소이며, 자세한 내용은 각각 [18]과 [15]에서 확인할 수 있다.

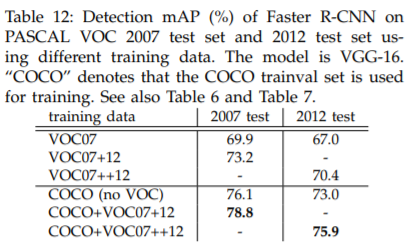


Table 12: Detection mAP (%) of Faster R-CNN on PASCAL VOC 2007 test set and 2012 test set using different training data. The model is VGG-16. “COCO” denotes that the COCO trainval set is used for training. See also Table 6 and Table 7.

표 12: PASCAL VOC 2007 테스트 세트 및 2012 테스트 세트에서의 고속 R-CNN 검출 mAP(%) 모델은 VGG-16입니다.「COCO」는, COCO 트레인밸 세트가 트레이닝에 사용되고 있는 것을 나타냅니다. 표 6 및 표 7을 참조하십시오.

**4.3 From MS COCO to PASCAL VOC**

Large-scale data is of crucial importance for improving deep neural networks. Next, we investigate how the MS COCO dataset can help with the detection performance on PASCAL VOC.

대규모 데이터는 심층 신경망을 개선하는 데 매우 중요하다. 다음으로, MS COCO 데이터 세트가 PASCAL VOC의 검출 성능에 어떻게 도움이 되는지 조사합니다.

As a simple baseline, we directly evaluate the COCO detection model on the PASCAL VOC dataset, without fine-tuning on any PASCAL VOC data. This evaluation is possible because the categories on COCO are a superset of those on PASCAL VOC. The categories that are exclusive on COCO are ignored in this experiment, and the softmax layer is performed only on the 20 categories plus background. The mAP under this setting is 76.1% on the PASCAL VOC 2007 test set (Table 12). This result is better than that trained on VOC07+12 (73.2%) by a good margin, even though the PASCAL VOC data are not exploited.

PASCAL VOC 데이터를 세부적으로 조정하지 않고 PASCAL VOC 데이터셋에서 COCO 탐지 모델을 직접 평가합니다. 이러한 평가는 COCO에 대한 범주가 PASCAL VOC에 대한 범주보다 상위 집합이기 때문에 가능합니다. 이 실험에서는 COCO에만 적용되는 범주가 무시되며, 소프트맥스 레이어는 20개 카테고리 + 배경에서만 수행됩니다. 이 설정에 따른 mAP는 PASCAL VOC 2007 테스트 세트의 76.1%입니다(표 12). 이 결과는 PASCAL VOC 데이터가 악용되지 않더라도 VOC07+12(73.2%)에서 상당한 차이로 교육된 결과보다 우수합니다.

Then we fine-tune the COCO detection model on the VOC dataset. In this experiment, the COCO model is in place of the ImageNet-pre-trained model (that is used to initialize the network weights), and the Faster R-CNN system is fine-tuned as described in Section 3.2. Doing so leads to 78.8% mAP on the PASCAL VOC 2007 test set. The extra data from the COCO set increases the mAP by 5.6%. Table 6 shows that the model trained on COCO+VOC has the best AP for every individual category on PASCAL VOC 2007. Similar improvements are observed on the PASCAL VOC 2012 test set (Table 12 and Table 7). We note that the test-time speed of obtaining these strong results is still about 200ms per image.

그런 다음 VOC 데이터 집합에서 COCO 탐지 모델을 미세 조정합니다. 이 실험에서, COCO 모델은 ImageNet의 사전 훈련된 모델(네트워크 가중치를 초기화하기 위해 사용됨)을 대신하며, 더 빠른 R-CNN 시스템은 섹션 3.2에 설명된 대로 미세 조정됩니다. 이렇게 하면 PASCAL VOC 2007 테스트 세트에서 78.8%의 mAP가 생성됩니다. COCO 세트의 추가 데이터는 mAP를 5.6% 증가시킵니다. 표 6은 COCO+VOC에 대해 훈련된 모델이 PASCAL VOC 2007의 모든 개별 범주에 대해 최고의 AP를 가지고 있음을 보여줍니다. PASCAL VOC 2012 테스트 세트에서도 유사한 개선이 관찰됩니다(표 12 및 표 7). 이러한 강력한 결과를 얻기 위한 테스트 시간 속도는 이미지당 약 200ms입니다.

**5. CONCLUSION**

We have presented RPNs for efficient and accurate region proposal generation. By sharing convolutional features with the downstream detection network, the region proposal step is nearly cost-free. Our method enables a unified, deep-learning-based object detection system to run at near real-time frame rates. The learned RPN also improves region proposal quality and thus the overall object detection accuracy

효율적이고 정확한 지역 제안 생성을 위한 RPN을 제시하였습니다. 컨볼루션 기능을 다운스트림 탐지 네트워크와 공유하여 지역 제안 단계를 거의 비용 부담 없이 진행할 수 있습니다. NAT의 방법을 사용하면 통합 딥러닝 기반 객체 감지 시스템을 거의 실시간 프레임률로 실행할 수 있습니다. 또한 학습된 RPN은 지역 제안 품질을 향상시켜 전체 객체 감지 정확도를 향상시킵니다.



Figure 5: Selected examples of object detection results on the PASCAL VOC 2007 test set using the Faster R-CNN system. The model is VGG-16 and the training data is 07+12 trainval (73.2% mAP on the 2007 test set). Our method detects objects of a wide range of scales and aspect ratios. Each output box is associated with a category label and a softmax score in [0, 1]. A score threshold of 0.6 is used to display these images. The running time for obtaining these results is 198ms per image, including all steps.

그림 5: Faster R-CNN 시스템을 사용한PASCAL VOC 2007 테스트 세트에서의 물체 검출 결과의 일부 예 모델은 VGG-16으로 트레이닝 데이터는 07+12 트레벌(2007년 테스트 세트의 73.2%)입니다. 우리의 방법은 광범위한 축척과 가로 세로 비율을 가진 물체를 감지한다. 각 출력 상자는 [0, 1]의 카테고리 라벨 및 소프트맥스 점수와 관련지어집니다. 점수 임계값 0.6은 이러한 이미지를 표시하는 데 사용됩니다. 이러한 결과를 얻기 위한 실행 시간은 모든 단계를 포함하여 이미지당 198ms입니다.