**SSD Single Shot MultiBox Detector**

Wei Liu1, Dragomir Anguelov2, Dumitru Erhan3, Christian Szegedy3, Scott Reed4, Cheng-Yang Fu1, Alexander C. Berg1

**Abstract**

We present a method for detecting objects in images using a single deep neural network. Our approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, COCO, and ILSVRC datasets confirm that SSD has competitive accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. For 300 × 300 input, SSD achieves 74.3% mAP1 on VOC2007 test at 59 FPS on a Nvidia Titan X and for 512 × 512 input, SSD achieves 76.9% mAP, outperforming a comparable state-of-the-art Faster R-CNN model. Compared to other single stage methods, SSD has much better accuracy even with a smaller input image size. Code is available at: <https://github.com/weiliu89/caffe/tree/ssd>.

우리는 하나의 깊은 신경망을 이용하여 이미지의 물체를 감지하는 방법을 제시합니다. NAT의 접근 방식인 SSD는 경계 상자의 출력 공간을 다양한 가로 세로 비율과 피쳐 맵 위치별 축척에 걸쳐 기본 상자 세트로 구분합니다. 예측 시 네트워크는 각 기본 상자에 있는 각 개체 범주의 존재에 대한 점수를 생성하고 개체 모양과 더 잘 일치하도록 상자에 대한 조정을 생성합니다. 또한 네트워크는 다양한 크기의 객체를 자연스럽게 처리하기 위해 해상도가 다른 여러 피쳐 맵의 예측을 결합합니다. SSD는 제안서 생성 및 후속 픽셀 또는 피쳐 재샘플링 단계를 완전히 없애고 모든 연산을 단일 네트워크에서 캡슐화하기 때문에 객체 제안이 필요한 방법에 비해 단순합니다. 따라서 SSD는 쉽게 교육할 수 있고 탐지 구성요소가 필요한 시스템에 쉽게 통합될 수 있습니다. PASCAL VOC, COCO 및 ILS VRC 데이터셋에 대한 실험 결과를 통해 SSD는 추가 객체 제안 단계를 활용하는 방법에 대한 경쟁적 정확성을 확보하고 있으며 훨씬 더 빠른 동시에 교육 및 추론을 위한 통합 프레임워크를 제공합니다. 300 × 300 입력의 경우, SSD는 Nvidia Titan X의 59 FPS에서 VOC2007 테스트에서 74.3% mAP1을 달성하고 512 × 512 입력의 경우 76.9% mAP를 달성하여 동급 최첨단의 Faster R-CNN 모델을 능가합니다. SSD는 다른 1단계 방식에 비해 입력 이미지 크기가 작아도 정확도가 월등히 높다. 코드는 https://github.com/weiliu89/caffe/tree/ssd에서 확인할 수 있습니다.

1. **Introduction**

Current state-of-the-art object detection systems are variants of the following approach: hypothesize bounding boxes, resample pixels or features for each box, and apply a highquality classifier. This pipeline has prevailed on detection benchmarks since the Selective Search work [1] through the current leading results on PASCAL VOC, COCO, and ILSVRC detection all based on Faster R-CNN[2] albeit with deeper features such as [3]. While accurate, these approaches have been too computationally intensive for embedded systems and, even with high-end hardware, too slow for real-time applications. Often detection speed for these approaches is measured in seconds per frame (SPF), and even the fastest high-accuracy detector, Faster R-CNN, operates at only 7 frames per second (FPS). There have been many attempts to build faster detectors by attacking each stage of the detection pipeline (see related work in Sec. 4), but so far, significantly increased speed comes only at the cost of significantly decreased detection accuracy.

현재 최첨단 객체 감지 시스템은 경계 상자를 가설화하고, 각 상자에 대해 픽셀 또는 피쳐를 다시 샘플링하며, 고품질 분류기를 적용하는 접근 방식의 변형입니다. PASCAL VOC, COCO 및 ILS VRC 검출에 대한 현재 최고의 결과를 통해 선택 검색 작업[1] 이후 [3]과 같은 보다 심층적인 기능을 가지고 있음에도 불구하고 이 파이프라인은 모두 Faster R-CNN[2]을 기반으로 한 검출 벤치마크에서 우세했습니다. 정확하기는 하지만, 이러한 접근 방식은 임베디드 시스템에 비해 너무 계산 집약적이었고, 하이엔드 하드웨어를 사용하더라도 실시간 애플리케이션에 비해 속도가 너무 느렸습니다. 이러한 접근법에 대한 탐지 속도는 종종 프레임당 초(SPF)로 측정되며, 가장 빠른 고정확성 검출기인 Faster R-CNN도 초당 7프레임(FPS)에서만 작동합니다. 검출 파이프라인의 각 단계를 공격하여 더 빠른 검출기를 구축하려는 시도가 많았지만(4항의 관련 작업 참조) 지금까지는 검출 정확도가 현저히 저하된 비용만을 들여 속도를 크게 높였다.

This paper presents the first deep network based object detector that does not resample pixels or features for bounding box hypotheses and and is as accurate as approaches that do. This results in a significant improvement in speed for high-accuracy detection (59 FPS with mAP 74.3% on VOC2007 test, vs. Faster R-CNN 7 FPS with mAP 73.2% or YOLO 45 FPS with mAP 63.4%). The fundamental improvement in speed comes from eliminating bounding box proposals and the subsequent pixel or feature resampling stage. We are not the first to do this (cf [4,5]), but by adding a series of improvements, we manage to increase the accuracy significantly over previous attempts. Our improvements include using a small convolutional filter to predict object categories and offsets in bounding box locations, using separate predictors (filters) for different aspect ratio detections, and applying these filters to multiple feature maps from the later stages of a network in order to perform detection at multiple scales. With these modifications—especially using multiple layers for prediction at different scales—we can achieve high-accuracy using relatively low resolution input, further increasing detection speed. While these contributions may seem small independently, we note that the resulting system improves accuracy on real-time detection for PASCAL VOC from 63.4% mAP for YOLO to 74.3% mAP for our SSD. This is a larger relative improvement in detection accuracy than that from the recent, very high-profile work on residual networks [3]. Furthermore, significantly improving the speed of high-quality detection can broaden the range of settings where computer vision is useful.

이 백서에서는 경계 상자 가설에 대해 픽셀 또는 기능을 다시 샘플링하지 않고 정확한 최초의 심층 네트워크 기반 객체 감지기를 제공합니다. 따라서 고정확도 검출 속도가 크게 향상되었습니다(VOC2007 테스트의 경우 mAP 74.3%인 FPS 59). 더 빠른 R-CNN 7 FPS(mAP 73.2% 또는 YOLO 45 FPS(mAP 63.4%). 기본적으로 속도가 향상되는 것은 경계 상자 제안과 후속 픽셀 또는 피쳐 재샘플링 단계를 제거하는 것입니다. 이 작업을 처음 수행한 것은 아니지만(cf[4,5]), 일련의 개선 사항을 추가함으로써 이전 시도보다 정확도를 크게 높일 수 있습니다. 향상된 기능에는 작은 컨볼루션 필터를 사용하여 경계 상자 위치의 객체 범주 및 오프셋을 예측하고, 서로 다른 가로 세로 비율 탐지에 대해 별도의 예측 변수(필터)를 사용하고, 이러한 필터를 네트워크의 이후 단계에서 여러 피쳐 맵에 적용하여 여러 척도로 탐지를 수행하는 것이 포함됩니다. 이러한 수정사항(특히 서로 다른 규모의 예측에 여러 레이어를 사용함)을 통해 상대적으로 낮은 해상도의 입력을 사용하여 높은 정확도를 달성하고 탐지 속도를 더욱 높일 수 있습니다. 이러한 기여도는 독립적으로 작게 보일 수 있지만, PASCAL VOC의 실시간 검출 정확도는 YOLO의 경우 63.4%에서 SSD의 경우 74.3%로 향상됩니다. 이는 잔류 네트워크에 대한 최근 매우 세간의 이목을 끄는 작업에 비해 검출 정확도가 크게 개선된 것입니다 [3]. 또한 고품질 검출 속도를 크게 개선하면 컴퓨터 비전이 유용한 설정 범위를 넓힐 수 있습니다.

We summarize our contributions as follows:

– We introduce SSD, a single-shot detector for multiple categories that is faster than the previous state-of-the-art for single shot detectors (YOLO), and significantly more accurate, in fact as accurate as slower techniques that perform explicit region proposals and pooling (including Faster R-CNN).

– The core of SSD is predicting category scores and box offsets for a fixed set of default bounding boxes using small convolutional filters applied to feature maps.

– To achieve high detection accuracy we produce predictions of different scales from feature maps of different scales, and explicitly separate predictions by aspect ratio.

– These design features lead to simple end-to-end training and high accuracy, even on low resolution input images, further improving the speed vs accuracy trade-off.

– Experiments include timing and accuracy analysis on models with varying input size evaluated on PASCAL VOC, COCO, and ILSVRC and are compared to a range of recent state-of-the-art approaches.

NAT은 다음과 같이 기여 내용을 요약합니다.

– 우리는 싱글샷 검출기(YOLO)의 이전 최신 기술보다 빠르고, 명시적인 영역 제안과 풀링을 수행하는 느린 기술(더 빠른 R-CNN 포함)만큼 정확도가 훨씬 더 높은 다중 범주를 위한 싱글샷 검출기 SSD를 도입했습니다.

– SSD의 핵심은 피쳐 맵에 적용된 작은 컨볼루션 필터를 사용하여 고정된 기본 경계 상자 집합에 대한 범주 점수 및 상자 오프셋을 예측하는 것입니다.

– 높은 탐지 정확도를 달성하기 위해 다양한 스케일의 피쳐 맵에서 다양한 스케일의 예측을 생성하고 가로 세로 비율에 따라 예측값을 명시적으로 구분합니다.

– 이러한 설계 기능은 저해상도 입력 이미지에서도 간단한 엔드 투 엔드 교육과 높은 정확도로 이어져 속도와 정확도 균형을 더욱 향상시킵니다.

– 실험에는 PASCAL VOC, COCO 및 ILSVRC에서 평가된 다양한 입력 크기를 가진 모델에 대한 타이밍 및 정확도 분석이 포함되며, 최근의 다양한 첨단 접근 방식과 비교됩니다.

1. **The Single Shot Detector (SSD)**

This section describes our proposed SSD framework for detection (Sec. 2.1) and the associated training methodology (Sec. 2.2). Afterwards, Sec. 3 presents dataset-specific model details and experimental results.

이 섹션에서는 검출에 대해 제안된 SSD 프레임워크(2.1절) 및 관련 교육 방법론(2.2절)에 대해 설명합니다. 그런 다음 3절에서는 데이터 세트별 모델 세부 정보와 실험 결과를 제공합니다.

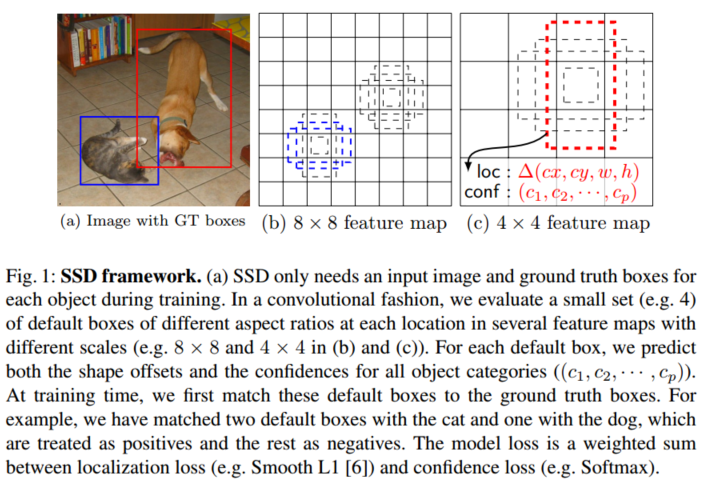


Fig. 1: SSD framework. (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8 × 8 and 4 × 4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ((c1, c2, · · · , cp)). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

그림 1: SSD 프레임워크 (a) SSD는 훈련 중 객체별로 입력 이미지 및 지상 정보 상자만 있으면 된다. 컨볼루션 방식으로, 우리는 서로 다른 척도를 가진 여러 특징 지도에서 각 위치에서 서로 다른 종횡비의 작은 세트(예: 4)를 평가한다(예: (b) 및 (c)의 8 × 8 및 4 × 4). 디폴트 박스 마다, 형상 오프셋과 모든 오브젝트 카테고리(c1, c2, ···, cp)의 신뢰도를 예측한다. 훈련 시에는 먼저 이러한 기본 상자를 지상 정보 상자와 일치시킵니다. 예를 들어, 고양이와 함께 두 개의 기본 상자와 개와 함께 한 개의 기본 상자는 양성으로 처리되고 나머지는 음성으로 처리됩니다. 모델 손실은 국산화 손실(예: 부드러운 L1 [6])과 신뢰 손실(예: 소프트맥스) 사이의 가중치 합이다.

**2.1. Model**

The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections. The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers), which we will call the base network2. We then add auxiliary structure to the network to produce detections with the following key features:

SSD 접근 방식은 고정 크기의 경계 상자 및 해당 상자에 객체 클래스 인스턴스의 존재에 대한 점수를 생성하는 피드-포워드 컨볼루션 네트워크를 기반으로 하며, 최종 탐지를 생성하기 위한 최대 억제 단계가 뒤따릅니다. 초기 네트워크 계층은 고품질 이미지 분류(분류 계층보다 먼저 잘림)에 사용되는 표준 아키텍처를 기반으로 하며, 이를 기본 네트워크라고 합니다2. 그런 다음 네트워크에 보조 구조를 추가하여 다음과 같은 주요 기능을 사용하여 탐지를 생성합니다.

**Multi-scale feature maps for detection** We add convolutional feature layers to the end of the truncated base network. These layers decrease in size progressively and allow predictions of detections at multiple scales. The convolutional model for predicting detections is different for each feature layer (cf Overfeat[4] and YOLO[5] that operate on a single scale feature map).

탐지를 위한 다중 스케일 피쳐 맵 잘린 기본 네트워크 끝에 컨볼루션 피쳐 레이어를 추가합니다. 이러한 도면층은 크기가 점진적으로 감소하며 여러 축척에서 탐지를 예측할 수 있습니다. 탐지를 예측하는 컨볼루션 모델은 각 피쳐 계층마다 다릅니다(cf 오버피트[4] 및 단일 스케일 피쳐 맵에서 작동하는 YOLO[5]).

**Convolutional predictors for detection** Each added feature layer (or optionally an existing feature layer from the base network) can produce a fixed set of detection predictions using a set of convolutional filters. These are indicated on top of the SSD network architecture in Fig. 2. For a feature layer of size m × n with p channels, the basic element for predicting parameters of a potential detection is a 3 × 3 × p small kernel that produces either a score for a category, or a shape offset relative to the default box coordinates. At each of the m × n locations where the kernel is applied, it produces an output value. The bounding box offset output values are measured relative to a default box position relative to each feature map location (cf the architecture of YOLO[5] that uses an intermediate fully connected layer instead of a convolutional filter for this step).

탐지를 위한 컨볼루션 예측 변수 추가된 각 피쳐 계층(또는 기본 네트워크의 기존 피쳐 계층)은 일련의 콘볼루션 필터를 사용하여 고정 탐지 예측 세트를 생성할 수 있습니다. 이는 그림 2의 SSD 네트워크 아키텍처 상단에 나와 있습니다. p 채널이 있는 m × n 크기의 피쳐 레이어의 경우, 잠재적 검출의 매개변수를 예측하기 위한 기본 요소는 범주에 대한 점수 또는 기본 상자 좌표에 상대적인 형상 오프셋을 생성하는 3 × 3 × p 작은 커널입니다. 커널이 적용되는 각 m × n 위치에서 출력 값을 생성합니다. 경계 상자 간격띄우기 출력 값은 각 피쳐 맵 위치에 상대적인 기본 상자 위치를 기준으로 측정됩니다(cf는 이 단계에서 컨볼루션 필터 대신 중간 완전 연결 도면층을 사용하는 YOLO[5] 아키텍처).

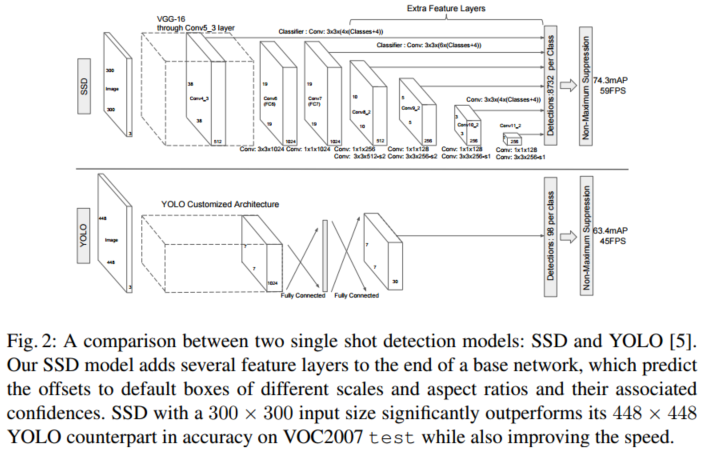


Fig. 2: A comparison between two single shot detection models: SSD and YOLO [5]. Our SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences. SSD with a 300 × 300 input size significantly outperforms its 448 × 448 YOLO counterpart in accuracy on VOC2007 test while also improving the speed.

그림 2: SSD와 YOLO의 두 싱글샷 감지 모델 비교 [5] NAT의 SSD 모델은 기본 네트워크의 끝에 여러 기능 계층을 추가하여 스케일 및 석면비가 다른 기본 상자에 대한 오프셋을 예측하고 이와 관련된 신뢰도를 예측합니다. 300 × 300 입력 크기의 SSD는 VOC2007 테스트에서 448 × 448 YOLO를 훨씬 능가하는 정확성을 자랑하며 속도도 향상시킵니다.

**Default boxes and aspect ratios** We associate a set of default bounding boxes with each feature map cell, for multiple feature maps at the top of the network. The default boxes tile the feature map in a convolutional manner, so that the position of each box relative to its corresponding cell is fixed. At each feature map cell, we predict the offsets relative to the default box shapes in the cell, as well as the per-class scores that indicate the presence of a class instance in each of those boxes. Specifically, for each box out of k at a given location, we compute c class scores and the 4 offsets relative to the original default box shape. This results in a total of (c + 4)k filters that are applied around each location in the feature map, yielding (c + 4)kmn outputs for a m × n feature map. For an illustration of default boxes, please refer to Fig. 1. Our default boxes are similar to the anchor boxes used in Faster R-CNN [2], however we apply them to several feature maps of different resolutions. Allowing different default box shapes in several feature maps let us efficiently discretize the space of possible output box shapes.

기본 상자 및 가로 세로 비율 네트워크 상단에 있는 여러 피쳐 맵을 위해 각 피쳐 맵 셀에 기본 경계 상자 집합을 연결합니다. 기본 상자는 해당 셀을 기준으로 각 상자의 위치가 고정되도록 피쳐 맵을 컨볼루션 방식으로 타일링합니다. 각 피쳐 맵 셀에서 셀의 기본 상자 모양에 상대적인 간격띄우기와 각 상자에 클래스 인스턴스가 있음을 나타내는 클래스별 점수를 예측합니다. 특히, 지정된 위치의 k에서 각 상자에 대해 원래 기본 상자 모양을 기준으로 c 클래스 점수와 4개의 오프셋을 계산합니다. 따라서 총 (c + 4)k 필터가 피쳐 맵의 각 위치 주위에 적용되어 m × n 피쳐 맵에 대해 (c + 4)kmn 출력을 산출합니다. 기본 상자의 그림은 그림 1을 참조하십시오. 기본 상자는 Faster R-CNN [2]에서 사용되는 앵커 박스와 유사하지만, 여러 기능 맵에 적용합니다. 여러 피쳐 맵에서 서로 다른 기본 상자 모양을 허용하면 가능한 출력 상자 모양의 공간을 효율적으로 구분할 수 있습니다.

* 1. **. Training**

The key difference between training SSD and training a typical detector that uses region proposals, is that ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Some version of this is also required for training in YOLO[5] and for the region proposal stage of Faster R-CNN[2] and MultiBox[7]. Once this assignment is determined, the loss function and back propagation are applied endto-end. Training also involves choosing the set of default boxes and scales for detection as well as the hard negative mining and data augmentation strategies.

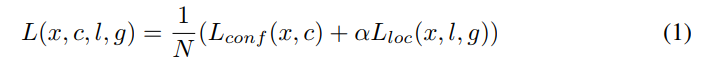
SSD 훈련과 지역 제안을 사용하는 일반적인 검출기 훈련의 주요 차이점은 접지 진실 정보를 검출기 출력 고정 세트의 특정 출력에 할당해야 한다는 것이다. 또한 일부 버전은 YOLO[5] 교육 및 Faster R-CNN[2] 및 MultiBox[7]의 지역 제안 단계에도 필요합니다. 이 할당이 결정되면 손실 함수와 후방 전파가 끝에서 끝까지 적용됩니다. 교육에는 탐지용 기본 상자 및 스케일 세트뿐만 아니라 하드 마이너스 마이닝 및 데이터 확장 전략도 함께 선택해야 합니다.

**Matching strategy** During training we need to determine which default boxes correspond to a ground truth detection and train the network accordingly. For each ground truth box we are selecting from default boxes that vary over location, aspect ratio, and scale. We begin by matching each ground truth box to the default box with the best jaccard overlap (as in MultiBox [7]). Unlike MultiBox, we then match default boxes to any ground truth with jaccard overlap higher than a threshold (0.5). This simplifies the learning problem, allowing the network to predict high scores for multiple overlapping default boxes rather than requiring it to pick only the one with maximum overlap.

일치 전략 교육 중에 지상 진실 탐지에 해당하는 기본 상자를 결정하고 그에 따라 네트워크를 교육해야 합니다. 각 접지 진실 상자에 대해 위치, 가로 세로 비율 및 축척에 따라 달라지는 기본 상자에서 선택합니다. 먼저 각 접지 진실 상자를 최상의 자카드 겹침이 있는 기본 상자에 일치시키는 것으로 시작합니다(MultiBox [7] 참조). 그런 다음 MultiBox와 달리 Jaccard가 임계값(0.5)보다 겹치는 모든 접지 진실에 기본 상자를 일치시킵니다. 이렇게 하면 네트워크가 중복되는 여러 개의 기본 상자에 대해 최대 중복되는 항목만 선택할 필요 없이 높은 점수를 예측할 수 있습니다.

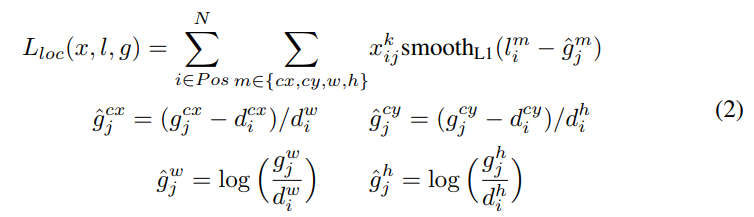
**Training objective** The SSD training objective is derived from the MultiBox objective [7,8] but is extended to handle multiple object categories. Let x p ij = {1, 0} be an indicator for matching the i-th default box to the j-th ground truth box of category p. In the matching strategy above, we can have P ixpij ≥ 1. The overall objective loss function is a weighted sum of the localization loss (loc) and the confidence loss (conf):

교육 목표 SSD 교육 목표는 MultiBox 목표[7,8]에서 도출되지만 여러 객체 범주를 처리하도록 확장됩니다. xpij = {1,0}이(가) i번째 기본 상자를 범주 p의 j번째 접지 진실 상자에 일치시키는 지표가 되도록 하십시오. 위의 매칭 전략에서, 우리는 Pixpij 1을 가질 수 있습니다. 전체적인 목표손실 함수는 국산화손실(local)과 신뢰손실(conf)의 가중치 합이다.



where N is the number of matched default boxes. If N = 0, wet set the loss to 0. The localization loss is a Smooth L1 loss [6] between the predicted box (l) and the ground truth box (g) parameters. Similar to Faster R-CNN [2], we regress to offsets for the center (cx, cy) of the default bounding box (d) and for its width (w) and height(h).

여기서 N은 일치된 기본 상자의 수입니다. N = 0이면 손실을 0으로 설정합니다. 국소화 손실은 예측 상자(l)와 지면 진실 상자(g) 매개변수 사이의 평활 L1 손실[6]입니다. 고속 R-CNN [2]과 마찬가지로 기본 경계 상자(d)의 중심(cx, cy) 및 너비(w)와 높이(h)에 대한 오프셋으로 회귀합니다.

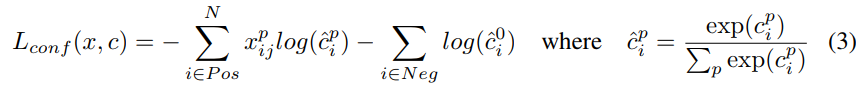


The confidence loss is the softmax loss over multiple classes confidences (c).

and the weight term α is set to 1 by cross validation.

신뢰 손실은 다중 클래스 신뢰에 대한 소프트 최대 손실입니다(c).

α는 교차 검증에 의해 1로 설정됩니다.

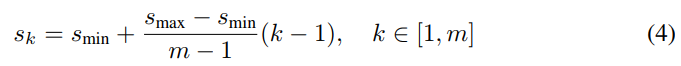


Choosing scales and aspect ratios for default boxes To handle different object scales, some methods [4,9] suggest processing the image at different sizes and combining the results afterwards. However, by utilizing feature maps from several different layers in a single network for prediction we can mimic the same effect, while also sharing parameters across all object scales. Previous works [10,11] have shown that using feature maps from the lower layers can improve semantic segmentation quality because the lower layers capture more fine details of the input objects. Similarly, [12] showed that adding global context pooled from a feature map can help smooth the segmentation results. Motivated by these methods, we use both the lower and upper feature maps for detection. Figure 1 shows two exemplar feature maps (8×8 and 4×4) which are used in the framework. In practice, we can use many more with small computational overhead.

기본 상자의 축척 및 가로 세로 비율 선택 서로 다른 객체 축척을 처리하려면 일부 방법[4,9]에서 이미지를 다른 크기로 처리하고 나중에 결과를 결합하는 것이 좋습니다. 그러나 예측에 단일 네트워크의 여러 다른 계층의 피쳐 맵을 활용하면 동일한 효과를 모방하는 동시에 모든 객체 규모에 걸쳐 매개변수를 공유할 수 있습니다. 이전 연구[10,11]에서는 하위 계층에서 피쳐 맵을 사용하면 하위 계층이 입력 객체의 보다 미세한 세부 정보를 캡처하기 때문에 의미 분할 품질을 향상시킬 수 있다는 것을 보여 주었습니다. 마찬가지로 [12]는 피쳐 맵에서 풀링된 전역 컨텍스트를 추가하는 것이 분할 결과를 원활하게 만드는 데 도움이 될 수 있음을 보여 주었습니다. 이러한 방법에 따라 우리는 하부 및 상부 피쳐 맵을 모두 사용하여 탐지합니다. 그림 1은 프레임워크에서 사용되는 두 가지 전형적인 기능 맵(8×8 및 4×4)을 보여줍니다. 실제로, 우리는 작은 컴퓨터 오버헤드로 더 많은 것을 사용할 수 있습니다.

Feature maps from different levels within a network are known to have different (empirical) receptive field sizes [13]. Fortunately, within the SSD framework, the default boxes do not necessary need to correspond to the actual receptive fields of each layer. We design the tiling of default boxes so that specific feature maps learn to be responsive to particular scales of the objects. Suppose we want to use m feature maps for prediction. The scale of the default boxes for each feature map is computed as:

네트워크 내에서 서로 다른 수준의 피쳐 맵은 수용 필드 크기가 서로 다른 것으로 알려져 있습니다 [13]. 다행히 SSD 프레임워크 내에서 기본 상자는 각 레이어의 실제 수신 필드와 일치할 필요가 없습니다. 기본 상자의 타일링을 설계하여 특정 피쳐 맵이 개체의 특정 축척에 반응하는 방법을 학습합니다. 예측에 m 피쳐 맵을 사용한다고 가정합니다. 각 피쳐 맵의 기본 상자 축척은 다음과 같이 계산됩니다.



where smin is 0.2 and smax is 0.9, meaning the lowest layer has a scale of 0.2 and the highest layer has a scale of 0.9, and all layers in between are regularly spaced. We impose different aspect ratios for the default boxes, and denote them as ar ∈ {1, 2, 3, 12, 13}. We can compute the width (wak = sk√ar) and height (hak = sk/√ar) for each default box. For the aspect ratio of 1, we also add a default box whose scale is s0 k = √sksk+1, resulting in 6 default boxes per feature map location. We set the center of each default box to (i+0.5|fk|, j+0.5|fk|), where |fk| is the size of the k-th square feature map, i, j ∈ [0, |fk|). In practice, one can also design a distribution of default boxes to best fit a specific dataset. How to design the optimal tiling is an open question as well.

여기서 smin은 0.2이고 smax는 0.9입니다. 즉, 가장 낮은 층은 0.2이고 가장 높은 층은 0.9이며 그 사이의 모든 층은 규칙적으로 간격이 지정됩니다. 기본 상자에 서로 다른 가로 세로 비율을 적용하고 이를 § {1, 2, 3, 12, 13}(으)로 나타냅니다. 각 기본 상자의 폭(hak = sk/skar)과 높이(hak = sk/skar)를 계산할 수 있습니다. 가로 세로 비율 1의 경우 축척이 s0k = kskk+1인 기본 상자도 추가하여 피쳐 맵 위치당 6개의 기본 상자가 생성됩니다. 각 기본 상자의 중심을 (i+0.5|fk|, j+0.5|fk|)로 설정합니다. 여기서 |fk|는 k번째 제곱 피쳐 맵의 크기입니다(i, j → [0, |fk|). 실제로 특정 데이터 집합에 가장 적합하도록 기본 상자의 분포를 설계할 수도 있습니다. 최적 타일을 설계하는 방법도 미해결 문제입니다.

By combining predictions for all default boxes with different scales and aspect ratios from all locations of many feature maps, we have a diverse set of predictions, covering various input object sizes and shapes. For example, in Fig. 1, the dog is matched to a default box in the 4 × 4 feature map, but not to any default boxes in the 8 × 8 feature map. This is because those boxes have different scales and do not match the dog box, and therefore are considered as negatives during training.

여러 피쳐 맵의 모든 위치에서 서로 다른 축척과 가로 세로 비율을 가진 모든 기본 상자에 대한 예측을 결합함으로써 다양한 입력 객체 크기와 모양을 포괄하는 다양한 예측 세트를 제공합니다. 예를 들어, 그림 1에서 개는 4 × 4 피쳐 맵의 기본 상자와 일치하지만 8 × 8 피쳐 맵의 기본 상자와 일치하지는 않습니다. 왜냐하면 저울이 다르고 도그박스와 맞지 않아 훈련 중 음성으로 간주되기 때문이다.

**Hard negative mining** After the matching step, most of the default boxes are negatives, especially when the number of possible default boxes is large. This introduces a significant imbalance between the positive and negative training examples. Instead of using all the negative examples, we sort them using the highest confidence loss for each default box and pick the top ones so that the ratio between the negatives and positives is at most 3:1. We found that this leads to faster optimization and a more stable training.

하드 마이너스 마이닝 일치 단계 이후 대부분의 기본 상자가 음수이며, 특히 가능한 기본 상자 수가 많은 경우에는 음수입니다. 이는 양성 및 음성 훈련 사례 간에 상당한 불균형을 초래합니다. 모든 부정적인 예를 사용하는 대신 각 기본 상자에 대해 가장 높은 신뢰 손실을 사용하여 정렬하고 상위 신뢰 손실을 선택하여 음과 양의 비율이 최대 3:1이 되도록 합니다. 우리는 이것이 더 빠른 최적화와 더 안정적인 훈련으로 이어진다는 것을 발견했습니다.

**Data augmentation** To make the model more robust to various input object sizes and shapes, each training image is randomly sampled by one of the following options:

– Use the entire original input image.

– Sample a patch so that the minimum jaccard overlap with the objects is 0.1, 0.3, 0.5, 0.7, or 0.9.

– Randomly sample a patch.

데이터 확대 다양한 입력 객체 크기와 모양에 대해 모델을 더욱 견고하게 만들기 위해 각 교육 이미지는 다음 옵션 중 하나를 사용하여 무작위로 샘플링됩니다.

– 전체 원본 입력 이미지를 사용합니다.

– 최소 자카드가 개체와 겹치도록 패치를 샘플링합니다(예: 0.1, 0.3, 0.5, 0.7 또는 0.9).

– 패치를 무작위로 추출합니다.

The size of each sampled patch is [0.1, 1] of the original image size, and the aspect ratio is between 12 and 2. We keep the overlapped part of the ground truth box if the center of it is in the sampled patch. After the aforementioned sampling step, each sampled patch is resized to fixed size and is horizontally flipped with probability of 0.5, in addition to applying some photo-metric distortions similar to those described in [14].

샘플링된 각 패치의 크기는 원래 이미지 크기의 [0.1,1]이고 가로 세로 비율은 12와 2 사이입니다. 만약 그라운드 진실 상자의 중심이 샘플링된 패치에 있다면 우리는 그라운드 진실 상자의 겹치는 부분을 보관합니다. 앞서 언급한 표본 추출 단계 후, 표본 추출된 각 패치는 고정된 크기로 크기가 조정되며 [14]에서 설명한 것과 유사한 광도 왜곡과 더불어 0.5의 확률로 수평 플립된다.

1. **Experimental Results**

**Base network** Our experiments are all based on VGG16 [15], which is pre-trained on the ILSVRC CLS-LOC dataset [16]. Similar to DeepLab-LargeFOV [17], we convert fc6 and fc7 to convolutional layers, subsample parameters from fc6 and fc7, change pool5 from 2 × 2 − s2 to 3 × 3 − s1, and use the a trous ` algorithm [18] to fill the ”holes”. We remove all the dropout layers and the fc8 layer. We fine-tune the resulting model using SGD with initial learning rate 10−3, 0.9 momentum, 0.0005 weight decay, and batch size 32. The learning rate decay policy is slightly different for each dataset, and we will describe details later. The full training and testing code is built on Caffe [19] and is open source at: <https://github.com/weiliu89/caffe/tree/ssd>.

기본 네트워크 우리의 실험은 모두 ILS VRC CLS-LOC 데이터셋[16]에서 미리 훈련된 VGG16[15]을 기반으로 합니다. DeepLab-Large와 유사FOV [17], 우리는 fc6과 fc7을 컨볼루션 레이어로 변환하고, 하위 샘플 매개변수를 fc6과 fc7에서 변환하고, 풀5를 2 × 2 - s2에서 3 × 3 - s1로 변경하고, "구멍"을 채우기 위해 적절한 알고리즘 [18]을 사용한다. 모든 드롭아웃 레이어와 fc8 레이어를 제거합니다. 초기 학습률 10-3, 0.9 모멘텀, 무게 감소 0.0005, 배치 크기 32의 SGD를 사용하여 결과 모델을 미세 조정합니다. 학습 속도 감소 정책은 데이터 세트마다 조금씩 다르며 자세한 내용은 나중에 설명하겠습니다. 전체 훈련 및 테스트 코드는 카페[19]를 기반으로 하며 https://github.com/weiliu89/caffe/tree/ssd에서 공개됩니다.

**3.1. PASCAL VOC2007**

On this dataset, we compare against Fast R-CNN [6] and Faster R-CNN [2] on VOC2007 test (4952 images). All methods fine-tune on the same pre-trained VGG16 network.

이 데이터셋에서는 VOC2007 테스트(4952 이미지)에서 Fast R-CNN[6] 및 Faster R-CNN[2]과 비교합니다. 모든 방법은 사전 교육된 동일한 VGG16 네트워크에서 미세 조정됩니다.

Figure 2 shows the architecture details of the SSD300 model. We use conv4 3, conv7 (fc7), conv8 2, conv9 2, conv10 2, and conv11 2 to predict both location and confidences. We set default box with scale 0.1 on conv4 33. We initialize the parameters for all the newly added convolutional layers with the ”xavier” method [20]. For conv4 3, conv10 2 and conv11 2, we only associate 4 default boxes at each feature map location – omitting aspect ratios of 13 and 3. For all other layers, we put 6 default boxes as described in Sec. 2.2. Since, as pointed out in [12], conv4 3 has a different feature scale compared to the other layers, we use the L2 normalization technique introduced in [12] to scale the feature norm at each location in the feature map to 20 and learn the scale during back propagation. We use the 10−3 learning rate for 40k iterations, then continue training for 10k iterations with 10−4 and 10−5. When training on VOC2007 trainval, Table 1 shows that our low resolution SSD300 model is already more accurate than Fast R-CNN. When we train SSD on a larger 512 × 512 input image, it is even more accurate, surpassing Faster R-CNN by 1.7% mAP. If we train SSD with more (i.e. 07+12) data, we see that SSD300 is already better than Faster R-CNN by 1.1% and that SSD512 is 3.6% better. If we take models trained on COCO trainval35k as described in Sec. 3.4 and fine-tuning them on the 07+12 dataset with SSD512, we achieve the best results: 81.6% mAP.

그림 2는 SSD300 모델의 아키텍처 세부 정보를 보여줍니다. 우리는 위치와 신뢰도를 모두 예측하기 위해 conv43, conv7(fc7), conv82, conv92, conv102 및 conv112를 사용합니다. 우리는 conv433에 스케일이 0.1인 기본 상자를 설정합니다. 우리는 "자비에" 방법을 사용하여 새로 추가된 모든 컨볼루션 레이어에 대한 매개변수를 초기화합니다[20]. conv43, conv102 및 conv112의 경우 각 피쳐 맵 위치에 4개의 기본 상자만 연결하고 가로 세로 비율은 13과 3입니다. 다른 모든 도면층의 경우 2.2항에서 설명한 대로 6개의 기본 상자를 배치했습니다. Conv43은 다른 도면층에 비해 피쳐 스케일이 다르기 때문에 [12]에서 소개한 L2 정규화 기법을 사용하여 피쳐 맵의 각 위치에서 피쳐 규범을 20으로 스케일링하고 역방향 전파 시 스케일을 학습합니다. 40K 반복에 10-3 학습률을 사용하고 10-4 및 10-5로 10K 반복에 대한 교육을 계속합니다. VOC2007 열차에 대한 교육에서, 표 1은 우리의 저해상도 SSD300 모델이 Fast R-CNN보다 더 정확하다는 것을 보여줍니다. 더 큰 512 × 512 입력 이미지에서 SSD를 교육했을 때, 그것은 더 정확하여 Faster R-CNN을 1.7% mAP 능가합니다. 더 많은 (07+12) 데이터로 SSD를 교육하면 SSD300이 Faster R-CNN보다 1.1%, SSD512가 3.6% 향상된 것을 알 수 있습니다. 3.4절에 설명된 것처럼 COCO train val35k에 대한 교육을 받은 모델을 SSD512로 07+12 데이터셋에서 미세 조정하면 81.6%의 mAP라는 최상의 결과를 얻을 수 있습니다.

To understand the performance of our two SSD models in more details, we used the detection analysis tool from [21]. Figure 3 shows that SSD can detect various object categories with high quality (large white area). The majority of its confident detections are correct. The recall is around 85-90%, and is much higher with “weak” (0.1 jaccard overlap) criteria. Compared to R-CNN [22], SSD has less localization error, indicating that SSD can localize objects better because it directly learns to regress the object shape and classify object categories instead of using two decoupled steps. However, SSD has more confusions with similar object categories (especially for animals), partly because we share locations for multiple categories. Figure 4 shows that SSD is very sensitive to the bounding box size. In other words, it has much worse performance on smaller objects than bigger objects. This is not surprising because those small objects may not even have any information at the very top layers. Increasing the input size (e.g. from 300×300 to 512×512) can help improve detecting small objects, but there is still a lot of room to improve. On the positive side, we can clearly see that SSD performs really well on large objects. And it is very robust to different object aspect ratios because we use default boxes of various aspect ratios per feature map location.

NAT의 두 SSD 모델의 성능을 보다 자세히 이해하기 위해 [21]의 탐지 분석 도구를 사용했습니다. 그림 3은 SSD가 고품질의 다양한 객체 범주(넓은 흰색 영역)를 감지할 수 있음을 보여줍니다. 대부분의 자신 있는 탐지가 정확합니다. 회수율은 약 85~90%로 '약함'(0.1자카드 겹침) 기준으로 훨씬 높다. R-CNN[22]에 비해 SSD의 위치 지정 오류는 적습니다. SSD는 분리된 두 단계를 사용하는 대신 개체 모양을 회귀시키고 개체 범주를 분류하는 방법을 직접 학습하기 때문에 개체를 더 잘 위치시킬 수 있습니다. 그러나 SSD는 유사한 개체 범주(특히 동물)와 혼동되는 부분이 더 많습니다. 부분적으로는 우리가 여러 범주의 위치를 공유하기 때문입니다. 그림 4는 SSD가 경계 상자 크기에 매우 민감하다는 것을 보여줍니다. 즉, 큰 물체보다 작은 물체에서 성능이 훨씬 떨어집니다. 이러한 작은 물체는 맨 위 계층에 아무런 정보도 없을 수 있기 때문에 이는 놀라운 일이 아닙니다. 입력 크기를 300×300에서 512×512로 늘리면 작은 물체의 감지 성능을 향상시킬 수 있지만 아직 개선해야 할 여지가 많습니다. 긍정적인 면에서는 SSD가 큰 물체에서 매우 우수한 성능을 발휘한다는 것을 알 수 있습니다. 또한 피쳐 맵 위치별로 다양한 가로 세로 비율의 기본 상자를 사용하기 때문에 다양한 객체 가로 세로 비율에 매우 적합합니다.

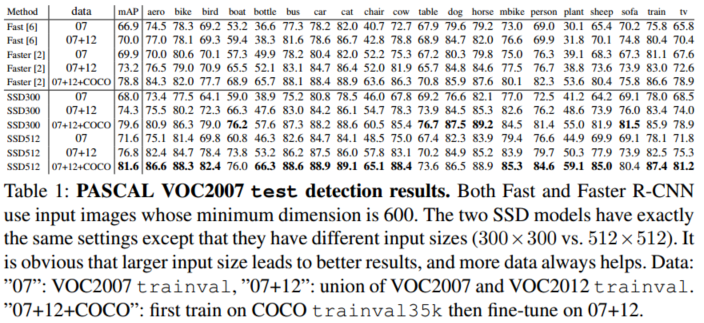


Table 1: PASCAL VOC2007 test detection results. Both Fast and Faster R-CNN use input images whose minimum dimension is 600. The two SSD models have exactly the same settings except that they have different input sizes (300×300 vs. 512×512). It is obvious that larger input size leads to better results, and more data always helps. Data: ”07”: VOC2007 trainval, ”07+12”: union of VOC2007 and VOC2012 trainval. ”07+12+COCO”: first train on COCO trainval35k then fine-tune on 07+12.

표 1: PASCAL VOC2007 테스트 검출 결과 고속 및 고속 R-CNN은 모두 최소 치수가 600인 입력 영상을 사용합니다. 두 SSD 모델은 입력 크기가 다르다는 점만 제외하면 설정이 완전히 동일합니다(300×300 대 512×512). 입력 사이즈가 클수록 결과가 좋아지는 것은 분명하며, 항상 더 많은 데이터가 도움이 됩니다. 데이터: 「07」: VOC2007 열차, 「07+12」: VOC2007 열차와 VOC2012 열차의 결합.「07+12+COCO」: COCO 열차 35k로 첫차 후, 07+12로 미세 조정.

* 1. **. Model analysis**

To understand SSD better, we carried out controlled experiments to examine how each component affects performance. For all the experiments, we use the same settings and input size (300 × 300), except for specified changes to the settings or component(s).

SSD를 더 잘 이해하기 위해 각 구성요소가 성능에 어떤 영향을 미치는지 검사하는 제어된 실험을 수행했습니다. 모든 실험에서 설정 또는 구성요소의 지정된 변경을 제외하고 동일한 설정 및 입력 크기(300 × 300)를 사용합니다.

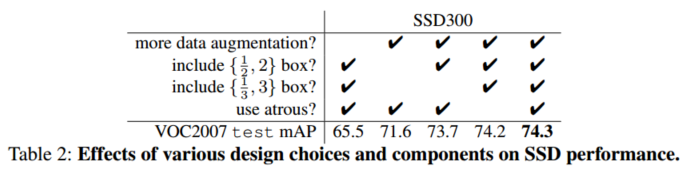


Table 2: Effects of various design choices and components on SSD performance.

표 2: 다양한 설계 선택 항목과 컴포넌트가 SSD 성능에 미치는 영향

Data augmentation is crucial. Fast and Faster R-CNN use the original image and the horizontal flip to train. We use a more extensive sampling strategy, similar to YOLO [5]. Table 2 shows that we can improve 8.8% mAP with this sampling strategy. We do not know how much our sampling strategy will benefit Fast and Faster R-CNN, but they are likely to benefit less because they use a feature pooling step during classification that is relatively robust to object translation by design.

데이터 확대가 매우 중요합니다. 빠르고 빠른 R-CNN은 원본 이미지와 수평 플립을 사용하여 훈련합니다. 우리는 YOLO[5]와 유사한 보다 광범위한 표본 추출 전략을 사용합니다. 표 2는 이 샘플링 전략으로 8.8% mAP를 개선할 수 있음을 보여줍니다. 우리는 우리의 샘플링 전략이 빠르고 빠른 R-CNN에 얼마나 도움이 될지는 모르지만, 그들은 설계에 의한 객체 변환에 상대적으로 강력한 기능 풀링 단계를 사용하기 때문에 혜택을 덜 받을 가능성이 있다.

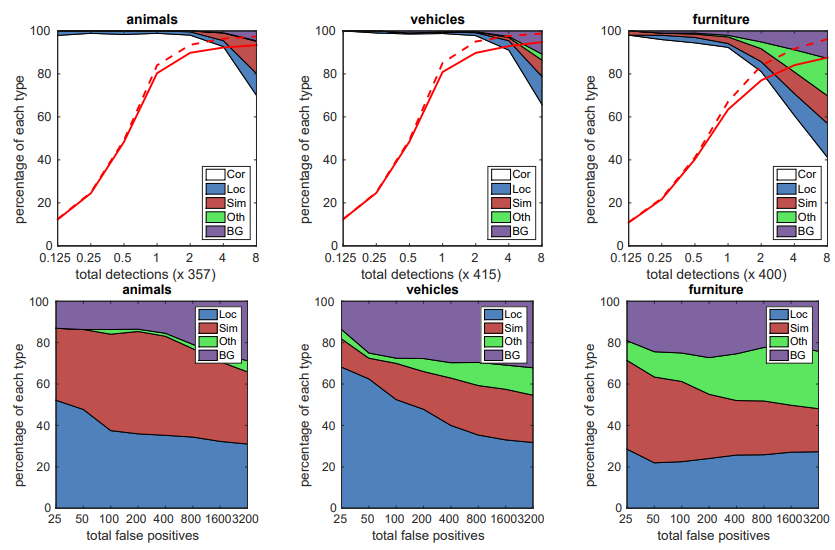


Fig. 3: **Visualization of performance for SSD512 on animals, vehicles, and furniture from VOC2007 test**. The top row shows the cumulative fraction of detections that are correct (Cor) or false positive due to poor localization (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG). The solid red line reflects the change of recall with strong criteria (0.5 jaccard overlap) as the number of detections increases. The dashed red line is using the weak criteria (0.1 jaccard overlap). The bottom row shows the distribution of top-ranked false positive types.

그림 3: VOC2007 테스트에 의한 동물, 자동차, 가구에서의 SSD512 성능 시각화 맨 위 행에는 현지화 불량(Loc), 유사한 범주(Sim), 다른 범주(Oth) 또는 배경(BG)과의 혼동으로 인해 올바른(Cor) 또는 잘못된 양성의 탐지의 누적 비율이 표시됩니다. 빨간색 실선은 검출 횟수가 증가함에 따라 강력한 기준(0.5 jaccard 중첩)을 사용하여 리콜의 변화를 나타냅니다. 빨간색 점선은 약한 기준(0.1 자카드 중첩)을 사용하고 있습니다. 맨 아래 행은 최상위 false positive 유형의 분포를 보여 줍니다.

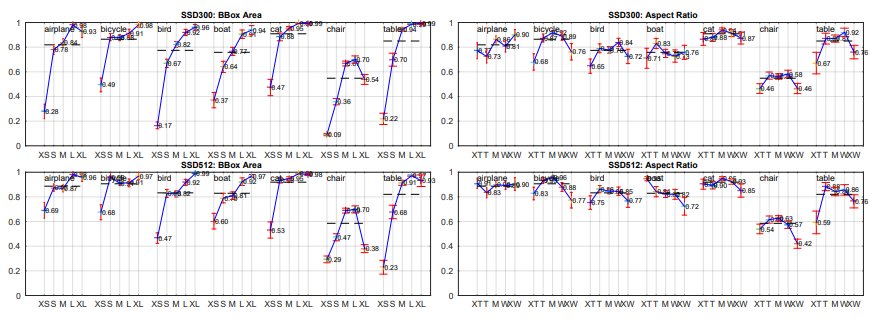


Fig. 4: **Sensitivity and impact of different object characteristics on VOC2007 test set using [21].** The plot on the left shows the effects of BBox Area per category, and the right plot shows the effect of Aspect Ratio. Key: BBox Area: XS=extra-small; S=small; M=medium; L=large; XL =extra-large. Aspect Ratio: XT=extra-tall/narrow; T=tall; M=medium; W=wide; XW =extra-wide.

그림 4: **[21]을 사용한 VOC2007 테스트 세트에 대한 다양한 물체 특성의 민감도 및 영향**. 왼쪽 그림은 범주당 BBox 면적의 효과를 나타내고 오른쪽 그림은 종횡비의 효과를 나타냅니다. 키: BBox 면적: XS=소형, S=소형, M=중형, L=대형, XL=소형. 석면비: XT=높이/좁음, T=높이, M=중간, W=폭, XW=폭.

**More default box shapes is better**. As described in Sec. 2.2, by default we use 6 default boxes per location. If we remove the boxes with 13 and 3 aspect ratios, the performance drops by 0.6%. By further removing the boxes with 12 and 2 aspect ratios, the performance drops another 2.1%. Using a variety of default box shapes seems to make the task of predicting boxes easier for the network.

기본 상자 모양이 더 좋습니다. 2.2절에 설명된 대로 기본적으로 위치당 6개의 상자를 사용합니다. 가로 세로 비율이 13과 3인 박스를 제거하면 성능이 0.6% 떨어집니다. 가로 세로 비율이 12와 2인 상자를 추가로 제거하면 성능이 2.1% 감소합니다. 다양한 기본 상자 모양을 사용하면 네트워크의 상자를 보다 쉽게 예측할 수 있습니다.

**Atrous is faster.** As described in Sec. 3, we used the atrous version of a subsampled VGG16, following DeepLab-LargeFOV [17]. If we use the full VGG16, keeping pool5 with 2 × 2 − s2 and not subsampling parameters from fc6 and fc7, and add conv5 3 for prediction, the result is about the same while the speed is about 20% slower.

아트로스가 더 빨라요. 3절에 설명된 대로 Deep Lab-Large에 이어 서브샘플링된 VGG16의 Atorus 버전을 사용했습니다.FOV [17]. fc6과 fc7의 서브샘플링 매개변수가 아닌 풀5를 2 × 2 - s2로 유지하고 conv53을 추가하여 전체 VGG16을 사용할 경우 속도는 약 20% 느리지만 결과는 거의 같다.

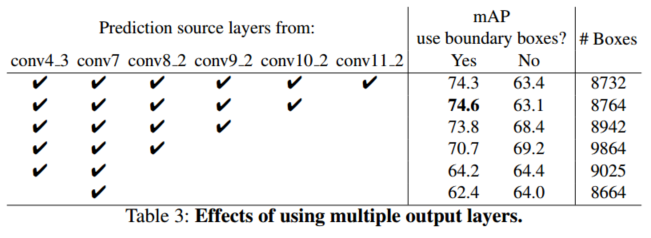


Table 3: Effects of using multiple output layers.

표 3: 복수의 출력 레이어를 사용하는 경우의 효과

**Multiple output layers at different resolutions is better.** A major contribution of SSD is using default boxes of different scales on different output layers. To measure the advantage gained, we progressively remove layers and compare results. For a fair comparison, every time we remove a layer, we adjust the default box tiling to keep the total number of boxes similar to the original (8732). This is done by stacking more scales of boxes on remaining layers and adjusting scales of boxes if needed. We do not exhaustively optimize the tiling for each setting. Table 3 shows a decrease in accuracy with fewer layers, dropping monotonically from 74.3 to 62.4. When we stack boxes of multiple scales on a layer, many are on the image boundary and need to be handled carefully. We tried the strategy used in Faster R-CNN [2], ignoring boxes which are on the boundary. We observe some interesting trends. For example, it hurts the performance by a large margin if we use very coarse feature maps (e.g. conv11 2 (1 × 1) or conv10 2 (3 × 3)). The reason might be that we do not have enough large boxes to cover large objects after the pruning. When we use primarily finer resolution maps, the performance starts increasing again because even after pruning a sufficient number of large boxes remains. If we only use conv7 for prediction, the performance is the worst, reinforcing the message that it is critical to spread boxes of different scales over different layers. Besides, since our predictions do not rely on ROI pooling as in [6], we do not have the collapsing bins problem in low-resolution feature maps [23]. The SSD architecture combines predictions from feature maps of various resolutions to achieve comparable accuracy to Faster R-CNN, while using lower resolution input images.

여러 개의 출력 계층이 서로 다른 해상도에서 더 좋습니다. SSD의 주요 기여는 출력 계층에 따라 서로 다른 스케일의 기본 상자를 사용하는 것입니다. 얻을 수 있는 이점을 측정하기 위해 단계적으로 레이어를 제거하고 결과를 비교합니다. 공정한 비교를 위해 도면층을 제거할 때마다 기본 박스 타일링을 조정하여 총 박스 수를 원본(8732)과 유사하게 유지합니다. 이는 나머지 레이어에 상자의 축척을 더 쌓고 필요한 경우 상자의 축척을 조정하여 수행됩니다. 각 설정에 대해 타일을 완전히 최적화하지는 않습니다. 표 3은 계층 수가 적을 때 정확도가 74.3에서 62.4로 단조롭게 감소하는 것을 보여줍니다. 여러 축척의 상자를 한 층에 쌓으면 이미지 경계에 많은 상자가 있으므로 주의하여 처리해야 합니다. Faster R-CNN [2]에서 사용된 전략을 사용해 경계에 있는 상자를 무시했습니다. 우리는 몇 가지 흥미로운 트렌드를 관찰합니다. 예를 들어, 매우 거친 피쳐 맵 (예: conv112 (1 × 1) 또는 conv102 (3 × 3))을 사용하면 성능이 크게 저하됩니다. 그 이유는 가지를 친 후 큰 물체를 덮을 수 있는 큰 상자가 충분하지 않기 때문일 수 있습니다. 주로 더 미세한 해상도 맵을 사용하면 충분한 수의 큰 상자를 가지치기 후에도 성능이 다시 향상되기 시작합니다. 예측에만 conv7을 사용할 경우 성능이 최악으로 나타나 서로 다른 축척의 상자를 다른 레이어에 분산시키는 것이 중요하다는 메시지를 강화합니다. 게다가, 우리의 예측은 [6]과 같이 ROI 풀링에 의존하지 않기 때문에, 저분해능 기능 맵에서는 붕괴 빈 문제가 없습니다 [23]. SSD 아키텍처는 다양한 해상도의 기능 맵에서 예측한 결과를 결합하여 낮은 해상도의 입력 이미지를 사용하면서 Faster R-CNN과 동등한 정확도를 달성합니다.

* 1. **. PASCAL VOC 2012**

We use the same settings as those used for our basic VOC2007 experiments above, except that we use VOC2012 trainval and VOC2007 trainval and test (21503 images) for training, and test on VOC2012 test (10991 images). We train the models with 10−3 learning rate for 60k iterations, then 10−4 for 20k iterations. Table 4 shows the results of our SSD300 and SSD5124 model. We see the same performance trend as we observed on VOC2007 test. Our SSD300 improves accuracy over Fast/Faster RCNN. By increasing the training and testing image size to 512×512, we are 4.5% more accurate than Faster R-CNN. Compared to YOLO, SSD is significantly more accurate, likely due to the use of convolutional default boxes from multiple feature maps and our matching strategy during training. When fine-tuned from models trained on COCO, our SSD512 achieves 80.0% mAP, which is 4.1% higher than Faster R-CNN.

VOC2012 trainval 및 VOC2007 trainval 및 test(21503 image)를 교육용으로 사용하고 VOC2012 test(10991 image)에 대한 테스트를 실시한다는 점을 제외하고는 상기 VOC2007 기본 실험에 사용한 설정과 동일한 설정을 사용합니다. 우리는 60K회 반복에 대해 10-3의 학습률로 모델을 교육한 후 20K회 반복에 대해 10-4의 학습률을 교육합니다. 표 4는 SSD300 및 SSD5124 모델의 결과를 보여줍니다. VOC2007 테스트에서 관찰한 것과 동일한 성능 추세를 볼 수 있습니다. Seagate SSD300은 Fast/Fast RCN에 비해 정확도가 향상되었으며, 교육 및 테스트 이미지 크기를 512×512로 늘림으로써 Fast R-CNN보다 4.5% 정확도가 높아졌으며, 여러 기능 맵의 컨볼루션 기본 박스와 교육 중 일치 전략이 사용되었을 가능성이 높습니다. NAT SSD512는 COCO에 대한 교육을 받은 모델에서 미세 조정했을 때 80.0%의 mAP를 달성하여 Faster R-CNN보다 4.1% 더 높습니다.

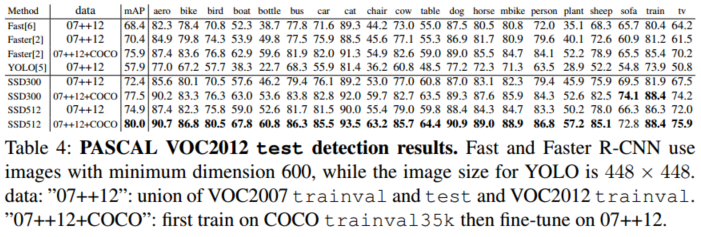


Table 4: PASCAL VOC2012 test detection results. Fast and Faster R-CNN use images with minimum dimension 600, while the image size for YOLO is 448 × 448. data: ”07++12”: union of VOC2007 trainval and test and VOC2012 trainval. ”07++12+COCO”: first train on COCO trainval35k then fine-tune on 07++12.

표 4: PASCAL VOC2012 테스트 검출 결과 고속 및 고속 R-CNN은 최소 치수 600의 이미지를 사용하는 반면 YOLO의 이미지 크기는 448 × 448입니다. 데이터: "07 + + 12": VOC2007 열차와 테스트와 VOC2012 열차의 결합. "07++12+COCO": COCO 열차 35k에서 첫 번째 열차와 07++12에서 미세 조정.

* 1. **. COCO**

To further validate the SSD framework, we trained our SSD300 and SSD512 architectures on the COCO dataset. Since objects in COCO tend to be smaller than PASCAL VOC, we use smaller default boxes for all layers. We follow the strategy mentioned in Sec. 2.2, but now our smallest default box has a scale of 0.15 instead of 0.2, and the scale of the default box on conv4 3 is 0.07 (e.g. 21 pixels for a 300 × 300 image)

SSD 프레임워크를 추가로 검증하기 위해 COCO 데이터셋에 대한 SSD300 및 SSD512 아키텍처를 교육했습니다. COCO의 객체는 PASCAL VOC보다 작은 경향이 있기 때문에 모든 레이어에 대해 작은 기본 박스를 사용합니다. 2.2절에 언급된 전략을 따르지만, 이제 최소 기본 상자의 배율은 0.2가 아닌 0.15이며, conv43의 기본 상자의 배율은 0.07입니다(예: 300 × 300 이미지의 경우 21픽셀).

We use the trainval35k [24] for training. We first train the model with 10−3 learning rate for 160k iterations, and then continue training for 40k iterations with 10−4 and 40k iterations with 10−5. Table 5 shows the results on test-dev2015. Similar to what we observed on the PASCAL VOC dataset, SSD300 is better than Fast R-CNN in both mAP@0.5 and mAP@[0.5:0.95]. SSD300 has a similar mAP@0.75 as ION [24] and Faster R-CNN [25], but is worse in mAP@0.5. By increasing the image size to 512 × 512, our SSD512 is better than Faster R-CNN [25] in both criteria. Interestingly, we observe that SSD512 is 5.3% better in mAP@0.75, but is only 1.2% better in mAP@0.5. We also observe that it has much better AP (4.8%) and AR (4.6%) for large objects, but has relatively less improvement in AP (1.3%) and AR (2.0%) for small objects. Compared to ION, the improvement in AR for large and small objects is more similar (5.4% vs. 3.9%). We conjecture that Faster R-CNN is more competitive on smaller objects with SSD because it performs two box refinement steps, in both the RPN part and in the Fast R-CNN part. In Fig. 5, we show some detection examples on COCO test-dev with the SSD512 model.

우리는 열차 발 35k[24]를 훈련에 사용합니다. 우선 160k 반복에 대해 10-3 학습률로 모델을 교육한 후 10-4, 10-5에 대해 40k 반복 교육을 계속합니다. 표 5는 test-dev2015에 대한 결과를 보여준다. PASCAL VOC 데이터 집합에서 관찰한 것과 마찬가지로 SSD300은 mAP@0.5와 mAP@[0.5:0.95] 모두에서 Fast R-CNN보다 우수합니다. SSD300의 mAP@0.75는 ION[24] 및 Faster R-CNN[25]과 유사하지만 mAP@0.5에서는 더 나쁩니다. 이미지 크기를 512 × 512로 늘림으로써 SSD512는 두 기준 모두에서 Faster R-CNN[25]보다 우수합니다. 흥미로운 점은 SSD512가 mAP@0.75에서는 5.3% 향상되었지만 mAP@0.5에서는 1.2% 향상되었다는 점입니다. 또한 큰 물체는 AP(4.8%)와 AR(4.6%)이 훨씬 낫지만 작은 물체는 AP(1.3%)와 AR(2.0%)이 상대적으로 덜 개선되는 것으로 관찰된다. ION과 비교했을 때 크고 작은 물체의 AR 개선은 (3.9%) 더 유사합니다. Faster R-CNN은 RPN 부분과 Fast R-CNN 부분 모두에서 두 가지 박스 미세화 단계를 수행하기 때문에 SSD로 작은 객체에서 더 경쟁력이 있을 것으로 추측됩니다. 그림 5에서는 SSD512 모델을 사용한 COCO 테스트 개발의 몇 가지 검출 예를 보여줍니다.

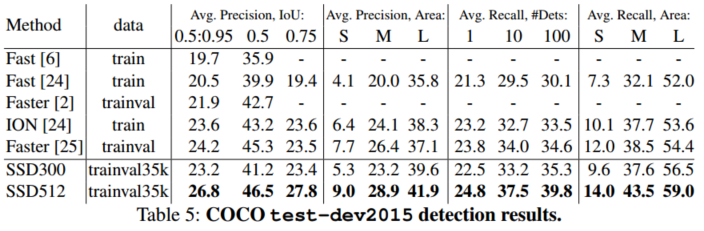


Table 5: COCO test-dev2015 detection results.표 5: COCO test-dev2015 검출 결과.

* 1. **. Preliminary ILSVRC results**

We applied the same network architecture we used for COCO to the ILSVRC DET dataset [16]. We train a SSD300 model using the ILSVRC2014 DET train and val1 as used in [22]. We first train the model with 10−3 learning rate for 320k iterations, and then continue training for 80k iterations with 10−4 and 40k iterations with 10−5. We can achieve 43.4 mAP on the val2 set [22]. Again, it validates that SSD is a general framework for high quality real-time detection.

COCO에 사용한 것과 동일한 네트워크 아키텍처를 ILSVRC DET 데이터셋에 적용했습니다 [16]. [22]에 사용된 대로 ILSVRC2014 DET Train 및 val1을 사용하여 SSD300 모델을 교육합니다. 우선 320,000회 반복 학습률 10-3으로 모델을 교육한 후 10-4회 반복 학습률 80,000회, 10-5회 반복 학습률 40,000회 교육을 계속합니다. 우리는 val2 세트[22]에서 43.4 mAP를 달성할 수 있습니다. SSD가 고품질 실시간 감지를 위한 일반적인 프레임워크임을 다시 한 번 확인합니다.

* 1. **. Data Augmentation for Small Object Accuracy**

Without a follow-up feature resampling step as in Faster R-CNN, the classification task for small objects is relatively hard for SSD, as demonstrated in our analysis (see Fig. 4). The data augmentation strategy described in Sec. 2.2 helps to improve the performance dramatically, especially on small datasets such as PASCAL VOC. The random crops generated by the strategy can be thought of as a ”zoom in” operation and can generate many larger training examples. To implement a ”zoom out” operation that creates more small training examples, we first randomly place an image on a canvas of 16× of the original image size filled with mean values before we do any random crop operation. Because we have more training images by introducing this new ”expansion” data augmentation trick, we have to double the training iterations. We have seen a consistent increase of 2%-3% mAP across multiple datasets, as shown in Table 6. In specific, Figure6 shows that the new augmentation trick significantly improves the performance on small objects. This result underscores the importance of the data augmentation strategy for the final model accuracy.

Faster R-CNN에서와 같이 후속 기능 재샘플링 단계가 없으면 분석 결과처럼 소형 객체에 대한 분류 작업이 SSD에 상대적으로 어렵습니다(그림 4 참조). 2.2절에 설명된 데이터 확장 전략은 특히 PASCAL VOC와 같은 소규모 데이터셋에서 성능을 획기적으로 향상하는 데 도움이 됩니다. 전략에 의해 생성된 랜덤 작물은 "줌인" 작업으로 간주할 수 있으며 더 큰 교육 사례를 생성할 수 있습니다. 더 작은 교육 예를 만드는 "줌 아웃" 작업을 구현하기 위해, 랜덤 자르기 작업을 수행하기 전에 먼저 평균 값으로 채워진 원래 이미지 크기의 16배 크기의 캔버스에 이미지를 무작위로 배치합니다. 이 새로운 "확장" 데이터 확장 트릭을 도입하여 교육 이미지가 더 많아지기 때문에 교육 반복 횟수를 두 배로 늘려야 합니다. 표 6과 같이 여러 데이터셋에서 mAP가 지속적으로 2%-3% 증가했습니다. 특히, 그림 6은 새로운 증강 트릭이 작은 물체의 성능을 크게 향상시킨다는 것을 보여줍니다. 이 결과는 최종 모델 정확도를 위한 데이터 확대 전략의 중요성을 강조합니다.

An alternative way of improving SSD is to design a better tiling of default boxes so that its position and scale are better aligned with the receptive field of each position on a feature map. We leave this for future work.

SSD를 개선하는 또 다른 방법은 기본 상자의 타일링을 설계하여 위치 및 배율이 피쳐 맵에서 각 위치의 수용 필드에 더 잘 정렬되도록 하는 것입니다. 이건 앞으로의 일을 위해서 남겨두는 거야

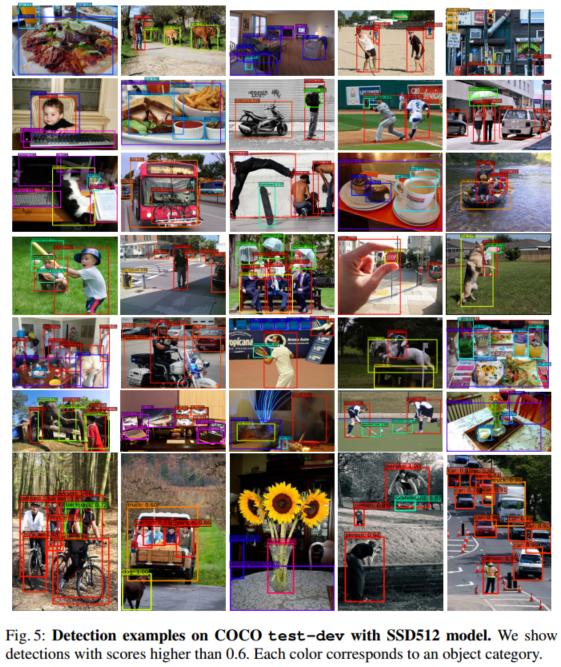


Fig. 5: Detection examples on COCO test-dev with SSD512 model. We show detections with scores higher than 0.6. Each color corresponds to an object category.

그림 5: SSD512 모델에서의 COCO 테스트 개발의 검출 예 우리는 0.6보다 높은 점수를 가진 탐지를 보여준다. 각 색상은 오브젝트 카테고리에 대응합니다.

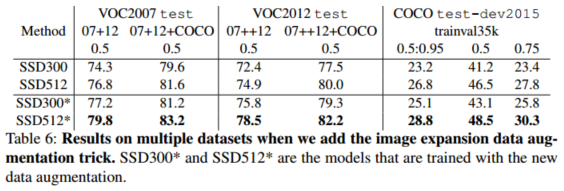


Table 6: Results on multiple datasets when we add the image expansion data augmentation trick. SSD300\* and SSD512\* are the models that are trained with the new data augmentation.

표 6: 이미지 확장 데이터 증가 트릭을 추가했을 때 여러 데이터 세트에 대한 결과 SSD300\* 및 SSD512\*는 새로운 데이터 증강을 위한 교육을 받은 모델입니다.

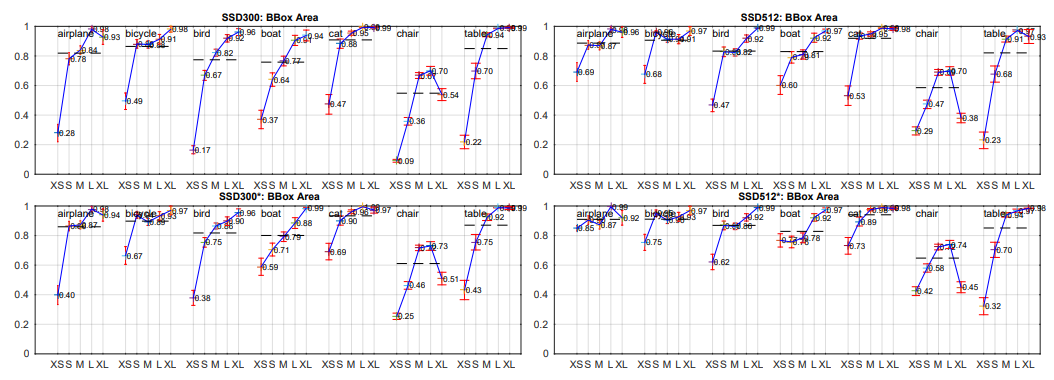


Fig. 6: Sensitivity and impact of object size with new data augmentation on VOC2007 test set using [21]. The top row shows the effects of BBox Area per category for the original SSD300 and SSD512 model, and the bottom row corresponds to the SSD300\* and SSD512\* model trained with the new data augmentation trick. It is obvious that the new data augmentation trick helps detecting small objects significantly.

그림 6: [21]을 사용한 VOC2007 테스트 세트에 대한 새로운 데이터 확대에 따른 객체 크기의 민감도와 영향. 위쪽 행은 원본 SSD300 및 SSD512 모델에 대한 범주당 BBox Area의 효과를 나타내고 아래쪽 행은 새로운 데이터 증강 기술로 교육받은 SSD300\* 및 SSD512\* 모델에 해당합니다. 새로운 데이터 증강 트릭이 작은 물체를 감지하는 데 큰 도움이 된다는 것은 명백합니다.

* 1. **. Inference time**

Considering the large number of boxes generated from our method, it is essential to perform non-maximum suppression (nms) efficiently during inference. By using a confidence threshold of 0.01, we can filter out most boxes. We then apply nms with jaccard overlap of 0.45 per class and keep the top 200 detections per image. This step costs about 1.7 msec per image for SSD300 and 20 VOC classes, which is close to the total time (2.4 msec) spent on all newly added layers. We measure the speed with batch size 8 using Titan X and cuDNN v4 with Intel Xeon [E5-2667v3@3.20GHz](mailto:E5-2667v3@3.20GHz).

당사의 방법에서 생성되는 많은 수의 상자를 고려할 때, 추론 시 최대값이 아닌 억제(nms)를 효율적으로 수행하는 것이 필수적입니다. 0.01의 신뢰 임계값을 사용하여 대부분의 상자를 필터링할 수 있습니다. 그런 다음 클래스당 0.45의 자카드 겹침이 있는 nms를 적용하고 이미지당 상위 200개의 탐지 결과를 유지합니다. SSD300의 경우 이미지당 약 1.7msec, VOC 클래스 20개로 새로 추가된 모든 레이어에 소요된 총 시간(2.4msec)에 가깝다. Intel Xeon E5-2667v3@3.20GHz의 Titan X 및 cuDNN v4를 사용하여 배치 크기 8로 속도를 측정합니다.

Table 7 shows the comparison between SSD, Faster R-CNN[2], and YOLO[5]. Both our SSD300 and SSD512 method outperforms Faster R-CNN in both speed and accuracy. Although Fast YOLO[5] can run at 155 FPS, it has lower accuracy by almost 22% mAP. To the best of our knowledge, SSD300 is the first real-time method to achieve above 70% mAP. Note that about 80% of the forward time is spent on the base network (VGG16 in our case). Therefore, using a faster base network could even further improve the speed, which can possibly make the SSD512 model real-time as well.

표 7은 SSD, 고속 R-CNN[2] 및 YOLO[5] 간의 비교를 보여줍니다. SSD300과 SSD512 모두 속도와 정확도에서 Faster R-CNN을 능가합니다. Fast YOLO[5]는 155 FPS에서 실행될 수 있지만 정확도는 거의 22% mAP만큼 낮습니다. 우리가 아는 한 SSD300은 70% mAP 이상을 달성하는 최초의 실시간 방법입니다. 전진 시간의 약 80%가 기본 네트워크(VGG16의 경우)에서 소비됩니다. 따라서 보다 빠른 베이스 네트워크를 사용하면 속도가 더욱 향상되어 SSD512 모델을 실시간으로 만들 수 있습니다.

1. **Related Work**

There are two established classes of methods for object detection in images, one based on sliding windows and the other based on region proposal classification. Before the advent of convolutional neural networks, the state of the art for those two approaches – Deformable Part Model (DPM) [26] and Selective Search [1] – had comparable performance. However, after the dramatic improvement brought on by R-CNN [22], which combines selective search region proposals and convolutional network based post-classification, region proposal object detection methods became prevalent.

이미지에는 슬라이딩 창을 기반으로 하는 방법과 지역 제안 분류를 기반으로 하는 두 가지 객체 감지 방법이 있습니다. 컨볼루션 신경 네트워크가 등장하기 전에는 두 가지 접근 방식인 변형 가능 부품 모델(DPM)[26]과 선택 검색[1]에 대한 최첨단 기술이 비슷한 성능을 보였습니다. 그러나, 선택적인 검색 지역 제안과 컨볼루션 네트워크 기반의 사후 분류를 결합한 R-CNN[22]이 가져온 극적인 개선 이후, 지역 제안 객체 감지 방법이 보편화되었습니다.

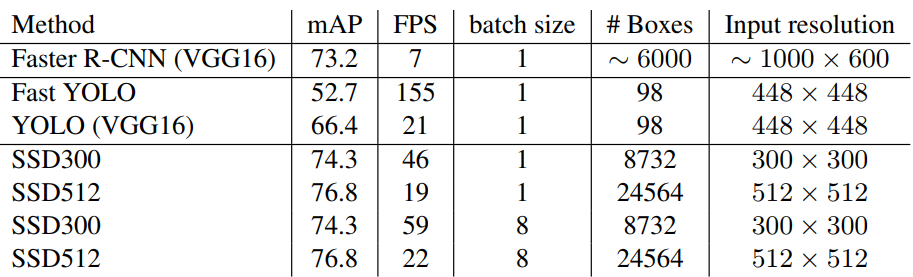


Table 7: Results on Pascal VOC2007 test. SSD300 is the only real-time detection method that can achieve above 70% mAP. By using a larger input image, SSD512 outperforms all methods on accuracy while maintaining a close to real-time speed.

표 7: Pascal VOC2007 테스트 결과 SSD300은 70% 이상의 mAP를 달성할 수 있는 유일한 실시간 감지 방법입니다. SSD512는 더 큰 입력 이미지를 사용함으로써 실시간에 가까운 속도를 유지하면서 정확도 면에서 모든 방법을 능가한다.

The original R-CNN approach has been improved in a variety of ways. The first set of approaches improve the quality and speed of post-classification, since it requires the classification of thousands of image crops, which is expensive and time-consuming. SPPnet [9] speeds up the original R-CNN approach significantly. It introduces a spatial pyramid pooling layer that is more robust to region size and scale and allows the classification layers to reuse features computed over feature maps generated at several image resolutions. Fast R-CNN [6] extends SPPnet so that it can fine-tune all layers end-toend by minimizing a loss for both confidences and bounding box regression, which was first introduced in MultiBox [7] for learning objectness.

원래의 R-CNN 접근 방식은 다양한 방식으로 개선되었습니다. 첫 번째 접근 방식은 수천 개의 이미지 크롭을 분류해야 하기 때문에 사후 분류의 품질과 속도를 향상시키며, 비용과 시간이 많이 소요됩니다. SPPnet[9]은(는) 기존의 R-CNN 접근 방식을 상당히 가속화합니다. 영역 크기와 규모에 보다 강력한 공간 피라미드 풀링 레이어를 도입하여 분류 레이어가 여러 이미지 해상도에서 생성된 피쳐 맵을 통해 계산된 피쳐를 재사용할 수 있습니다. Fast R-CNN[6]은 SPPnet을 확장하여 객관성 학습을 위해 MultiBox[7]에 처음 소개된 신뢰도 및 바운딩 박스 회귀 분석의 손실을 최소화함으로써 모든 계층을 엔드 투 엔드로 미세 조정할 수 있도록 합니다.

The second set of approaches improve the quality of proposal generation using deep neural networks. In the most recent works like MultiBox [7,8], the Selective Search region proposals, which are based on low-level image features, are replaced by proposals generated directly from a separate deep neural network. This further improves the detection accuracy but results in a somewhat complex setup, requiring the training of two neural networks with a dependency between them. Faster R-CNN [2] replaces selective search proposals by ones learned from a region proposal network (RPN), and introduces a method to integrate the RPN with Fast R-CNN by alternating between finetuning shared convolutional layers and prediction layers for these two networks. This way region proposals are used to pool mid-level features and the final classification step is less expensive. Our SSD is very similar to the region proposal network (RPN) in Faster R-CNN in that we also use a fixed set of (default) boxes for prediction, similar to the anchor boxes in the RPN. But instead of using these to pool features and evaluate another classifier, we simultaneously produce a score for each object category in each box. Thus, our approach avoids the complication of merging RPN with Fast R-CNN and is easier to train, faster, and straightforward to integrate in other tasks.

두 번째 접근법은 깊은 신경망을 이용한 제안 생성의 품질을 향상시킵니다. MultiBox[7,8]와 같은 가장 최근의 작업에서, 낮은 수준의 이미지 기능을 기반으로 하는 선택적 검색 영역 제안은 별도의 심층 신경망에서 직접 생성된 제안으로 대체됩니다. 이는 검출 정확도를 더욱 향상시키지만 설정이 다소 복잡해 두 개의 신경망이 서로 종속되어 있는 훈련을 필요로 합니다. 빠른 R-CNN [2]은 지역 제안 네트워크(RPN)에서 학습한 검색 제안을 선별적으로 대체하고, 이 두 네트워크에 대한 정밀 조정 공유 콘볼루션 계층과 예측 계층을 번갈아 가며 RPN을 Fast R-CNN과 통합하는 방법을 도입합니다. 이러한 방식으로 지역 제안을 사용하여 중간 수준의 피쳐를 풀링하고 최종 분류 단계는 비용이 덜 듭니다. NAT SSD는 RPN의 앵커 박스와 유사한 고정 세트의 (기본) 박스도 예측에 사용한다는 점에서 Faster R-CN의 지역 제안 네트워크(RPN)와 매우 유사합니다. 그러나 이러한 기능을 사용하여 피쳐를 풀링하고 다른 분류자를 평가하는 대신 각 상자의 각 개체 범주에 대한 점수를 동시에 생성합니다. 따라서 NAT의 접근 방식은 RPN을 Fast R-CNN과 병합하는 복잡성을 방지하고 다른 작업에 보다 쉽고 빠르게 통합하기 쉽습니다.

Another set of methods, which are directly related to our approach, skip the proposal step altogether and predict bounding boxes and confidences for multiple categories directly. OverFeat [4], a deep version of the sliding window method, predicts a bounding box directly from each location of the topmost feature map after knowing the confidences of the underlying object categories. YOLO [5] uses the whole topmost feature map to predict both confidences for multiple categories and bounding boxes (which are shared for these categories). Our SSD method falls in this category because we do not have the proposal step but use the default boxes. However, our approach is more flexible than the existing methods because we can use default boxes of different aspect ratios on each feature location from multiple feature maps at different scales. If we only use one default box per location from the topmost feature map, our SSD would have similar architecture to OverFeat [4]; if we use the whole topmost feature map and add a fully connected layer for predictions instead of our convolutional predictors, and do not explicitly consider multiple aspect ratios, we can approximately reproduce YOLO [5].

NAT의 접근 방식과 직접적인 관련이 있는 또 다른 방법 집합은 제안 단계를 모두 건너뛰고 여러 범주에 대한 경계 상자 및 신뢰도를 직접 예측합니다. 슬라이딩 윈도우 방법의 심층 버전인 OverFeat [4]는 기본 객체 범주의 신뢰도를 알고 나면 최상위 피쳐 맵의 각 위치에서 직접 경계 상자를 예측합니다. YOLO [5]는 전체 최상위 피쳐 맵을 사용하여 (이러한 범주에 대해 공유되는) 여러 범주와 경계 상자에 대한 신뢰도를 모두 예측합니다. 제안 단계는 없지만 기본 상자를 사용하기 때문에 SSD 방식이 이 범주에 속합니다. 그러나 다양한 축척의 여러 피쳐 맵에서 각 피쳐 위치에 가로 세로 비율이 다른 기본 상자를 사용할 수 있기 때문에 기존 방법보다 유연합니다. 최상위 피쳐 맵에서 위치당 하나의 기본 상자만 사용하는 경우 SSD의 아키텍처는 OverFeat [4]와 유사합니다. 최상위 피쳐 맵을 사용하고 컨볼루션 예측 변수 대신 완전히 연결된 레이어를 추가하고 여러 가로 세로 비율을 명시적으로 고려하지 않으면 대략적으로 YOL을 재현할 수 있습니다.O [5].

1. **Conclusions**

This paper introduces SSD, a fast single-shot object detector for multiple categories. A key feature of our model is the use of multi-scale convolutional bounding box outputs attached to multiple feature maps at the top of the network. This representation allows us to efficiently model the space of possible box shapes. We experimentally validate that given appropriate training strategies, a larger number of carefully chosen default bounding boxes results in improved performance. We build SSD models with at least an order of magnitude more box predictions sampling location, scale, and aspect ratio, than existing methods [5,7]. We demonstrate that given the same VGG-16 base architecture, SSD compares favorably to its state-of-the-art object detector counterparts in terms of both accuracy and speed. Our SSD512 model significantly outperforms the state-of-theart Faster R-CNN [2] in terms of accuracy on PASCAL VOC and COCO, while being 3× faster. Our real time SSD300 model runs at 59 FPS, which is faster than the current real time YOLO [5] alternative, while producing markedly superior detection accuracy.

이 백서에서는 여러 범주의 고속 싱글샷 객체 감지기인 SSD에 대해 소개합니다. NAT 모델의 주요 특징은 네트워크 상단에 있는 여러 피쳐 맵에 연결된 다중 스케일 컨볼루션 경계 상자 출력을 사용하는 것입니다. 이 표현을 통해 가능한 상자 모양의 공간을 효율적으로 모델링할 수 있습니다. 적절한 교육 전략이 주어졌을 때, 신중하게 선택된 많은 수의 기본 경계 상자가 성능 향상으로 이어진다는 것을 실험적으로 검증한다. NAT은 기존 방법보다 최소한 박스 예측 샘플링 위치, 척도 및 가로 세로 비율이 더 큰 SSD 모델을 구축합니다[5,7]. 동일한 VGG-16 기반 아키텍처를 고려할 때 SSD는 정확도와 속도 면에서 최신 객체 감지기에 비해 우수합니다. 당사의 SSD512 모델은 PASCAL VOC 및 COCO의 정확성 측면에서 최신 Faster R-CNN[2]을 크게 능가하는 동시에 3배 더 빠릅니다. 실시간 SSD300 모델은 59FPS로 실행되는데, 이는 현재 실시간 YOLO[5] 모델보다 빠르면서도 검출 정확도가 현저하게 뛰어납니다.

Apart from its standalone utility, we believe that our monolithic and relatively simple SSD model provides a useful building block for larger systems that employ an object detection component. A promising future direction is to explore its use as part of a system using recurrent neural networks to detect and track objects in video simultaneously.

독립형 유틸리티 외에도, NAT의 비교적 단순한 단일 SSD 모델은 객체 감지 구성요소를 사용하는 대규모 시스템에 유용한 구성 요소를 제공합니다. 미래 유망한 방향은 영상의 물체를 동시에 감지하고 추적하기 위해 반복되는 신경망을 사용하는 시스템의 일부로서 그것의 사용을 탐구하는 것이다.

1. **Acknowledgement**

This work was started as an internship project at Google and continued at UNC. We would like to thank Alex Toshev for helpful discussions and are indebted to the Image Understanding and DistBelief teams at Google. We also thank Philip Ammirato and Patrick Poirson for helpful comments. We thank NVIDIA for providing GPUs and acknowledge support from NSF 1452851, 1446631, 1526367, 1533771.

이 작업은 구글에서 인턴십 프로젝트로 시작되어 UNC에서 계속 진행되었으며, 알렉스 토셰프(Alex Toshev)의 도움에 감사드리며 구글의 이미지 이해 및 디스트빌리프(Image Understanding and DistBelief) 팀에게 은혜를 입고 있습니다. 우리는 또한 필립 아미라토와 패트릭 푸어슨에게 도움을 준 것에 감사한다. GPU를 제공해 주신 NVIDIA에 감사드리며 NSF 1452851, 1446631, 1526367, 1533771의 지원을 인정합니다.