SSD: Single Shot MultiBox Detector

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg

UNC Chapel Hill Zoox Inc. Google Inc. University of Michigan, Ann-Arbor

개요. 이 논문에서는 단일 심층 신경망을 사용하여 이미지에서 물체를 감지하는 방법을 소개합니다. SSD라고 명명한 우리의 접근 방식은 경계 박스의 출력 공간을 다양한 가로 세로 비율에 걸쳐 기본 상자 세트로 이산화하고 특징맵(feature map)에 위치별로 확장합니다. 예측 단계에서는 네트워크는 각 기본 상자에 있는 각 개체에 대한 점수를 생성하고 개체 모양과 더 알맞게 일치하도록 상자를 조정합니다. 또한 네트워크는 다양한 크기의 개체를 자연스럽게 처리하기 위해 해상도가 다른 여러 특징맵의 예측을 결합합니다. SSD는 제안 생성과 후속 픽셀 또는 기능 재샘플링 단계를 완전히 제거하고 모든 계산을 단일 네트워크에서 캡슐화하기 때문에 객체 제안이 필요한 방법에 비해 단순합니다. 따라서 SSD를 쉽게 교육하고 탐지 구성 요소가 필요한 시스템에 쉽게 통합할 수 있습니다. PASCAL VOC, COCO 및 ILSVRC 데이터셋에 대한 실험 결과는 SSD가 추가 객체 제안 단계를 활용하는 방법에 대한 경쟁력 있는 정확도를 가지고 있으며 훨씬 더 빠르다는 것을 확인함과 동시에 학습과 예측을 위한 통합 프레임워크를 제공합니다. 300 * 300 입력의 경우, SSD는 Nvidia Titan X의 59 FPS에서 VOC 2007 테스트에서 74.3%의 mAP를 달성하고 512 * 512 입력의 경우 SSD는 76.9%의 mAP를 달성하며 동급의 최첨단 고속 R-CNN 모델을 능가함을 증명했습니다. 다른 단일 단계 방법에 비해 SSD는 입력 이미지 크기가 작아도 정확도가 훨씬 뛰어납니다.

코드는 아래에서 확인 할 수 있습니다.

https://github.com/weiliu89/caffe/tree/ssd .

Keywords: 실시간 객체 감지, 합성곱 신경망

1. 소개

감지 시스템은 경계 상자를 가정하고, 각 상자의 픽셀 또는 기능을 다시 샘플링하고, 고품질 분류기를 적용하는 접근 방식을 변형시킨 형태입니다. 이 파이프라인은 선택 검색이 더 깊은 기능을 가진 Faster R-CNN을 기반으로 PASCAL VOC, COCO 및 ILSVRC 탐지에 대한 현재 선도적인 결과를 통해 작업한 이후 탐지 벤치마크에서 우세했습니다. 이러한 방식은 정확하기는 하

지만, 임베디드 시스템에 대해서는 계산 집약적이며, 고급 하드웨어를 사용하더라도 실시간 애플리케이션에서 활용하기에는 너무 느립니다. 탐지 속도의 정도는 보통 초당 프레임(FPS)으로 측정하며, 가장 빠른 고정밀 탐지기인 Faster R-CNN도 초당 7프레임(FPS)으로 작동합니다.

탐지 파이프라인의 각 단계를 공략하여 더 빠른 탐지기를 구축하려는 많은 시도가 있었지만(4장의 관련 작업 참조) 지금까지는 탐지 정확도를 크게 저하시키는 대가를 치르고 나서야 상당히 빠른 속도를 얻을 수 있었습니다.

이 논문에서는 경계 상자 가설을 위해 픽셀이나 특징을 다시 샘플링하지 않고 기존 접근 방식만큼 정확한 최초의 심층 네트워크 기반 객체 검출기를 소개합니다. 이 모델에서는 고정밀 검출 속도가 크게 향상되었습니다(VOC2007 테스트에서 mAP 74.3%를 달성했을 때 59 FPS). mAP 73.2%를 사용하는 R-CNN의 7 FPS 또는 mAP 63.4%를 사용하는 YOLO의 45 FPS보다 더 빠릅니다. 경계 상자 제안과 후속 픽셀 또는 기능 재샘플링 단계를 없앰으로써 속도가 근본적으로 개선됩니다. 이러한 시도를 우리가 처음한 것은 아니지만, 일련의 개선 사항을 추가함으로써 이전보다 정확도를 크게 높일 수 있었습니다.

우리의 개선점에는 경계 상자 위치의 객체 범주 및 오프셋을 예측하기 위해 작은 컨볼루션 필터를 사용하고, 다른 가로 세로 비율 탐지에 대해 별도의 예측기(필터)를 사용했으며, 이는 다중 스케일에서 탐지를 수행하기 위해 네트워크의 후기 단계부터 여러 기능 맵에 이러한 필터를 적용하는 것 모두에 포함됩니다. 이러한 수정 사항(특히 서로 다른 척도에서 예측을 위해 여러 레이어를 사용)을 사용하면 비교적 낮은 해상도 입력을 사용하여 높은 정확도를 달성할 수 있어 탐지 속도를 더욱 높일 수 있습니다. 이것의 효과는 작아 보일 수 있지만, 결과 시스템이 PASCAL VOC에 대한 실시간 감지 정확도를 YOLO의 63.4% mAP에서 SSD의 74.3% mAP로 향상시킨다는 점에 주목해야 합니다. 이는 다른 네트워크에 대한 최근의 매우 높은 인지도의 작업보다 탐지 정확도가 상대적으로 더 향상되었음을 보여줍니다. 또한 고품질 탐지 속도를 크게 향상시키면 컴퓨터 비전의 유의미한 활용 범위를 넓힐 수 있습니다.

성과에 대한 요약 :

- 우리는 이전의 싱글샷 검출기(YOLO)보다 빠르고, 실제로 명시적인 지역 제안 및 풀링(고속 R-CNN 포함)을 수행하는 느린 기술만큼 정확하며, 훨씬 더 정확한 여러 범주를 위한 싱글샷 검출기 SSD를 소개합니다.
- SSD의 핵심은 특징맵에 적용되는 작은 컨볼루션 필터를 사용하여 기본 경계 상자 고정 집합에 대한 범주 점수 및 상자 오프셋을 예측하는 것입니다.
- 높은 탐지 정확도를 달성하기 위해 우리는 다른 척도의 특징 맵에서 다른 척도의 예측을 생성하고 측면별로 예측을 명시적으로 분리합니다.
- 이러한 설계 기능은 저해상도 입력 이미지에서도 간단한 종단 간 교육 및 높은 정확도로 이어져 속도 대 정확도의 균형을 더욱 향상시킵니다.
- 실험은 PASCAL VOC, COCO 및 ILSVRC에서 평가된 다양한 입력 크기를 가진 모델에 대한 타이밍 및 정확도 분석을 포함하며 최근의 최첨단 접근 방식과 비교됩니다.

2. 단일 샷 감지기(SSD)

이번 장에서는 검출을 위해 제안된 SSD프레임워크(2.1장)와 관련 교육 방법론(2.2장)을 설명합니다. 그 후, 3장에서는 데이터셋 별 모델 세부 정보와 실험 결과를 제시합니다.

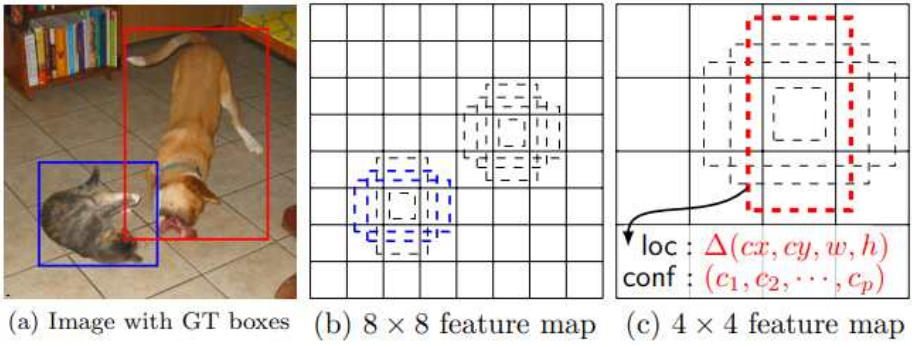

그림 1: SSD 프레임워크

(a) SSD는 훈련 중 객체별로 입력 이미지와 지상 실측 상자만 있으면 됩니다. 컨볼루션 방식으로, 우리는 축척이 다른 여러 형상 맵에서 각 위치에서 서로 다른 종횡비의 기본 상자의 작은 세트를 평가합니다(예: (b) 및 (c)의 8 * 8 및 4 * 4). 각 기본 상자에 대해 모든 객체 범주(((c1, c2, ... , cp))에 대한 형상 간격띄우기 및 신뢰도를 모두 예측합니다. 학습시에, 먼저 이러한 기본 상자를 실측 자료와 일치시킵니다. 예를 들어 기본 상자 두 개를 고양이와 일치시키고 한 개는 개와 일치시켰는데, 이 상자는 긍정으로, 나머지는 부정으로 취급됩니다. 모델 손실은 국소화 손실과 신뢰 손실(Softmax) 사이의 가중 합계입니다.

2.1 모델

SSD 접근 방식은 경계 상자의 고정 크기 집합과 해당 상자에 객체 클래스 인스턴스가 존재하는 것에 대한 점수를 생성하는 피드포워드 컨볼루션 네트워크를 기반으로 하며, 이후 최종 탐지를 생성하기 위한 최대 억제 단계가 아닙니다. 초기 네트워크 계층은 고품질 이미지 분류에 사용되는 표준 아키텍처를 기반으로 합니다(분류 계층보다 먼저 절단됨). 이를 기본 네트워크라고 부릅니다. 그런 다음 네트워크에 보조 구조를 추가하여 다음과 같은 주요 기능으로 탐지를 생성합니다.

- **탐지를 위한 다중 스케일 특징맵:** 우리는 잘린 기본 네트워크의 끝에 컨볼루션 피처 레이어를 추가합니다. 이러한 레이어는 크기가 점진적으로 감소하고 여러 척도에서 탐지를 예측할 수 있습니다. 탐지를 예측하기 위한 컨볼루션 모델은 각 형상 계층(단일 척도 형상 맵에서 작동하는 Overfeat 및 YOLO)마다 다릅니다.
- **탐지를 위한 컨볼루션 예측:** 각 기능 계층(또는 기본 네트워크의 기존 기능 계층)은 일련의 컨볼루션 필터를 사용하여 고정된 탐지 예측 세트를 생성할 수 있습니다. 이러한 정보는 그림 2의 SSD 네트워크 아키텍처 위에 나와 있습니다. p 채널이 있는 m * n 크기의 특징 계층의 경우, 잠재적 검출의 매개 변수를 예측하는 기본 요소는 범주에 대한 점수 또는 기본 상자 좌표를 기준으로 한 모양 오프셋을 생성하는 3 * 3 * p 작은 커널입니다. 커널이 적용되는 각 *n 위치에서 출력 값을 생성합니다. 경계 상자 오프셋 출력 값은 기본값을 기준으로 측정됩니다.

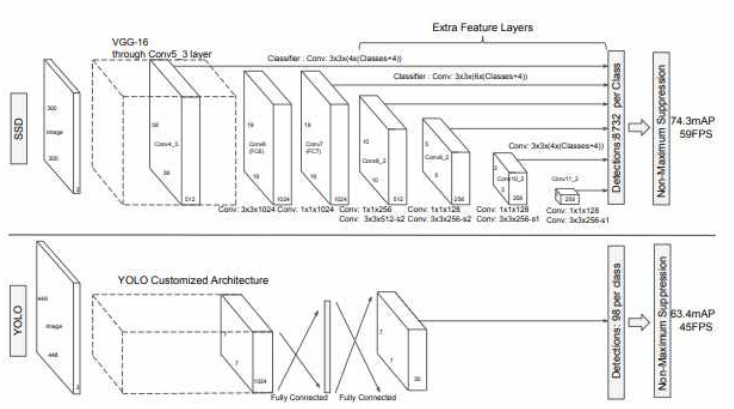

그림 2: SSD와 YOLO

SSD와 YOLO라는 두 가지 싱글샷 감지 모델 간의 비교입니다. Seagate의 SSD 모델은 여러 기능 계층을 기본 네트워크 끝에 추가하여 다양한 규모와 가로 * 세로 비율의 기본 상자에 대한 오프셋과 관련 신뢰도를 예측합니다. 입력 크기가 300 * 300인 SSD는 VOC2007 테스트에서 정확도 면에서 448 * 448 YOLO를 크게 능가하는 동시에 속도도 향상되었습니다. 상자의 배치는 각 특징맵에 상대적인 위치에 두었습니다.

디폴트박스 와 면의 비율 네트워크 상단에 있는 여러 기능 맵에 대해 기본 경계 상자 집합을 각 특징맵 셀과 연결합니다. 기본 상자는 해당 셀에 상대적인 각 상자의 위치가 고정되도록 특징맵을 컨볼루션 방식으로 타일링합니다. 각 특징맵 셀에서 셀의 기본 상자 모양과 각 상자에 클래스 인스턴스의 존재를 나타내는 클래스별 점수에 상대적인 오프셋을 예측합니다. 특히, 주어진 위치에서 각 상자 중 k개의 상자에 대해 c 클래스 점수와 원래 기본 상자 모양에 상대적인 4개의 오프셋을 계산합니다. 따라서 특징맵의 각 위치 주변에 총 $(c + 4)k$ 필터가 적용되어 mn 특징맵에 대해 $(c + 4)kmn$ 출력을 산출합니다. 기본 상자의 그림은 그림 1을 참조하십시오. 기본 상자는 고속 R-CNN에 사용되는 앵커 상자와 유사하지만, 서로 다른 해상도의 여러 특징맵에 적용합니다. 여러 특징맵에서 서로 다른 기본 상자 모양을 허용하면 가능한 출력 상자 모양의 공간을 효율적으로 이산화할 수 있습니다.

2.2 학습

SSD 훈련과 지역 제안을 사용하는 일반적인 검출기 훈련의 주요 차이점은 고정 검출기 출력 세트의 특정 출력에 실측 정보를 할당해야 한다는 것입니다. 일부 버전은 YOLO 학습 및 빠른 R-CNN 및 MultiBox의 지역 제안 단계에도 필요합니다. 이 할당이 결정되면 손실 함수와 역전파가 엔드 투 엔드(End-to-End)로 적용됩니다. 학습에는 탐지를 위한 기본 상자 및 척도 세트 선택과 하드 네거티브 마이닝 및 데이터 확대 전략도 포함됩니다.

일치 전략 훈련 중에 우리는 지상 실측 탐지에 해당하는 기본 상자를 결정하고 그에 따라 네트워크를 훈련시켜야 합니다. 각 지상 실측 정보 상자에 대해 위치, 가로 세로 비율 및 규모에 따라 다양한 기본 상자에서 선택합니다. 우리는 각 지상 진실 상자를 (MultiBox에서와 같이) 가장 좋은 jaccard 중첩이 있는 기본 상자에 일치시키는 것으로 시작합니다. MultiBox와 달리, 우리는 기본 상자를 임계값(0.5)보다 높은 jaccard 중첩을 가진 모든 지상 진실에 일치시킵니다. 이렇게 하면 학습 문제가 단순화되므로 네트워크는 최대 중첩된 기본 상자만 선택하도록 요구하지 않고 여러 개의 중복된 기본 상자에 대한 높은 점수를 예측할 수 있습니다.

학습 목표 SSD 교육 목표는 다중 상자 목표에서 파생되지만 여러 개체 범주를 처리하도록 확장됩니다. $x_{ij}^{p} = \{1, 0\}$가 i번째 기본 상자를 범주 p의 i번째 지상 진실 상자와 일치시키기 위한 지표가 될 수 있습니다. 위의 매칭 전략에서 우리는 가질 수 있습니다. $\Sigma_i x_{ij}^{p} \geq 1$ 는 전체 목표 손실 함수는 현지화 손실(loc)과 신뢰 손실($conf$)의 가중 합입니다.

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

(1)

여기서 N은 일치하는 기본 상자의 수입니다. $N = 0$이면 손실을 0으로 설정합니다. 국소화 손실은 예측 상자(l)와 지상 실측 상자(g) 파라미터 사이의 부드러운 L1 손실입니다. 고속 R-CNN과 마찬가지로 기본 경계 상자(d)의 중심(cx, cy)과 너비(w) 및 높이(h)에 대한 오프셋으로 회귀합니다.

$$\begin{aligned} L_{loc}(x,l,g) &= \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m) \\ \hat{g}_j^{cx} &= (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \\ \hat{g}_j^{w} &= \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right) \end{aligned}$$

(2)

신뢰 손실은 여러 등급의 신뢰에 대한 소프트맥스 손실입니다(c).

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

(3)

그리고 가중치 항 α는 교차 검증에 의해 1로 설정됩니다.

기본 상자에 대한 척도 및 비율을 선택 서로 다른 개체 척도를 처리하기 위해 일부 방법은 이미지를 다른 크기로 처리하고 그 후에 결과를 결합하는 것을 제안합니다. 그러나 예측을 위해 단일 네트워크에서 여러 다른 계층의 특징맵을 활용함으로써 동일한 효과를 모방하는 동시에 모든 개체 규모에서 매개 변수를 공유할 수 있습니다. 이전 연구에서는 하위 계층의 특징맵을 사용하면 하위 계층이 입력 객체의 더 미세한 세부 정보를 캡처하기 때문에 의미론적 세분화 품질을 향상시킬 수 있음을 보여주었습니다. 마찬가지로, 특징맵에서 풀링된 글로벌 컨텍스트를 추가하면 분할 결과를 원활하게 하는 데 도움이 된다는 것을 보여 주었습니다. 이러한 방법에 자극을 받아, 우리는 탐지를 위해 하부 및 상부 특징맵을 모두 사용합니다. 그림 1은 프레임워크에 사용되는 두 가지 예시 특징맵(8 * 8 및 4 * 4)을 보여줍니다. 실제로 우리는 적은 계산 오버헤드로 더 많은 것을 사용할 수 있습니다. 네트워크 내에서 서로 다른 수준의 특징맵은 서로 다른 (경험적)수용 필드 크기를 갖는 것으로 알려져 있습니다. 다행히 SSD 프레임워크 내에서 기본 상자는 각 계층의 실제 수신 필드에 해당할 필요가 없습니다. 우리는 기본 상자의 타일링을 설계하여 특정 특징맵이 객체의 특정 척도에 반응하는 법을 배우도록 합니다. 예측에 'm'특징맵을 사용하려고 합니다. 각 특징맵에 대한 기본 상자의 축척은 다음과 같이 계산됩니다.

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \quad k \in [1, m]$$

(4)

여기서 $S_{\min}$은 0.2이고 $S_{\max}$ 0.9입니다. 즉, 가장 낮은 층의 척도가 0.2이고, 가장 높은 층의 척도는 0.9이며, 그 사이의 모든 층은 일정한 간격으로 배치됩니다. 기본 상자에 서로 다른 비율을 적용하고 $a_r \in \left\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\right\}$로 표시합니다. 각 기본 상자의 너비$\left(w_k^a = s_k\sqrt{a_r}\right)$와 높이$\left(h_k^a = s_k/\sqrt{a_r}\right)$를 계산할 수 있습니다. 가로 세로 비율이 1인 경우 척도가 $s_k' = \sqrt{s_k s_{k+1}}$인 기본 상자도 추가하여 특징맵 위치당 기본 상자 6개를 생성합니다. 각 기본 상자의 중심을 $\left(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|}\right)$로 설정합니다. 여기서 $|f_k|$는 k번째 제곱 특징맵의 크기입니다.$i, j \in (0, |f_k|)$

실제로 기본 상자의 분포를 다음과 같이 설계할 수도 있습니다. 많은 특징맵의 모든 위치에서 축척과 비율이 다른 모든 기본 상자에 대한 예측을 결합함으로써 다양한 입력 개체 크기와 모양을 포함하는 다양한 예측 세트를 사용할 수 있습니다. 예를 들어, 그림 1에서 개는 4 * 4 특정맵의 기본 상자와 일치하지만 8 * 8 특징맵의 기본 상자는 일치하지 않습니다. 왜냐하면 그 상자들은 가중치가 다르고 개 상자와 일치하지 않기 때문에 훈련에 부적합하다고 여겨지기 때문입니다.

틀렸던 어려운 샘플 추출 매칭 단계 후에는 가능한 기본 상자 수가 많은 경우 대부분의 기본 상자가 음수입니다. 이것은 긍정적인 훈련 사례와 부정적인 훈련 사례 사이에 상당한 불균형을 초래합니다. 모든 부정적인 예제를 사용하는 대신 각 기본 상자에 대해 가장 높은 신뢰 손실을 사용하여 분류하고 음성과 양성의 비율이 최대 3:1이 되도록 상위 항목을 선택합니다. 우리는 이것이 더 빠른 최적화와 더 안정적인 교육으로 이어진다는 것을 발견했습니다.

데이터 증산 다양한 입력 객체 크기와 모양에 대해 모델을 더욱 견고하게 만들기 위해 각 교육 이미지는 다음 옵션 중 하나로 랜덤하게 샘플링됩니다.

- 원본 이미지 전체 사용
- 최소 jaccard가 개체와 겹치도록 패치를 샘플링
- 랜덤한 패치 표본 추출

샘플링된 각 패치의 크기는 원래 이미지 크기의 [0.1, 1]이며 가로 세로 비율은 1/2 과 2 사이입니다. 우리는 그것의 중심이 샘플링된 패치에 있다면 지상 진실 상자의 중첩된 부분을 유지합니다. 앞서 언급한 샘플링 단계 후, 샘플링된 각 패치는 고정된 크기로 크기가 조정되고 0.5의 확률로 수평 플립됩니다. 또한 이전에 설명된 것과 유사한 광도계 왜곡을 적용합니다.

3. 실험 결과

기본 망 우리의 실험은 모두 ILSVRC CLS-LOC 데이터 세트에서 사전 훈련된 VGG16을 기반으로 합니다. DeepLab-Large와 유사합니다.FOV에서는 fc6 및 fc7을 컨볼루션 레이어로 변환하고, fc6 및 fc7의 매개 변수를 하위 샘플링하고, 풀 5를 2 × 2 - s2에서 3 × 3 - s1로 변경하고, atrous 알고리즘을 사용하여 "구멍"을 채웁니다. 모든 드롭아웃 계층과 fc8 계층을 제거합니다. 초기 학습 속도 10^{-3}, 0.9 운동량, 0.0005 체중 감쇠 및 배치 크기 32로 SGD를 사용하여 결과 모델을 미세 조정합니다. 학습 속도 붕괴 정책은 데이터 세트마다 조금씩 다르며, 자세한 내용은 나중에 설명하겠습니다. 전체 학습 및 테스트 코드는 Caffe를 기반으로 하며 아래 사이트에서 볼 수 있습니다. (https://github.com/weiliu89/caffe/tree/ssd)

2.2 PASCAL VOC 2007

이 데이터 세트에서 VOC2007 테스트(4952 이미지)의 Fast R-CNN 및 Fast R-CNN과 비교합니다. 모든 방법은 사전 훈련된 동일한 VGG16 네트워크에서 미세 조정됩니다. 그림 2는 SSD300 모델의 아키텍처 세부 정보를 보여줍니다. 우리는 위치와 신뢰 모두를 예측하기 위해 conv4_3, conv7(fc7), conv8_2, conv9_2, conv10_2, conv11_2를 사용합니다. 우리는 conv4_3에 스케일 0.1로 기본 상자를 설정합니다. 우리는 "xavier" 방법을 사용하여 새로 추가된 모든 컨볼루션 레이어에 대한 매개 변수를 초기화합니다. conv4_3, conv10_2 및 conv11_2의 경우 각 형상 맵 위치에서 1/3과 3의 가로 세로 비율을 제외한 4개의 기본 상자만 연결합니다.

다른 모든 레이어에 대해서는 2.2 장에서 설명한 대로 6개의 기본 박스를 배치했습니다. conv4_3은 다른 레이어와 다른 피쳐 스케일을 가지고 있기 때문에 이전에 소개된 L2 정규화 기법을 사용하여 특징맵의 각 위치에서 피쳐 규범을 20으로 스케일링하고 역전파 중에 스케일을 학습합니다. 우리는 40k 반복에 대해 10^{-3}의 학습률을 사용한 다음 10^{-4} 및 10^{-5}로 10k 반복에 대한 훈련을 계속합니다. VOC2007 trainval에 대한 교육을 실시할 때 표 1은 저해상도 SSD300 모델이 Fast

R-CNN보다 이미 더 정확하다는 것을 보여줍니다. 더 큰 512 * 512 입력 이미지에서 SSD를 교육할 경우 Fast R-CNN보다 1.7% 더 정확합니다. SSD를 더 많은(07+12와 같은) 데이터로 교육하면 SSD300이 Faster R-CNN보다 1.1% 더 우수하고 SSD512가 3.6% 더 우수하다는 것을 알 수 있습니다. 3.4항에 설명된 대로 COCO train val 35k에 대해 훈련된 모델을 가져와서 SSD512를 사용하여 07+12 데이터 세트에서 미세 조정하면 81.6% mAP라는 최고의 결과를 얻을 수 있습니다.

두 SSD 모델의 성능을 더 자세히 이해하기 위해 탐지 분석 도구를 사용했습니다. 그림 3은 SSD가 고품질로 다양한 객체 범주(흰색 영역이 넓음)를 감지할 수 있음을 보여줍니다. 대부분의 자신있는 탐지는 정확합니다. 회수율은 약 85-90%이며, "취약"(0.1 jaccard 중복) 기준으로 훨씬 높습니다. SSD는 R-CNN에 비해 현지화 오류가 적으며, 이는 SSD가 두 개의 분리된 단계를 사용하는 대신 개체 모양을 회귀하고 개체 범주를 분류하는 방법을 직접 학습하기 때문에 개체를 더 잘 현지화할 수 있음을 나타냅니다. 그러나 SSD는 유사한 개체 범주(특히 동물의 경우)와 더 많은 혼동을 일으키는데, 이는 부분적으로 우리가 여러 범주의 위치를 공유하기 때문입니다. 그림 4는 SSD가 경계 상자 크기에 매우 민감하다는 것을 보여줍니다. 다시 말해, 소형에서 성능이 훨씬 떨어집니다.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**

표 1: PASCAL VOC2007 결과

고속 및 고속 R-CNN 모두 최소 치수가 600인 입력 영상을 사용합니다. 두 SSD 모델은 입력 크기가 다르다는 점(300 * 300 대 512 * 512)을 제외하고는 설정이 정확히 동일합니다. 입력 크기가 클수록 더 나은 결과를 얻을 수 있으며, 데이터가 많을수록 항상 도움이 됩니다. 데이터: "07": VOC2007 trainval, "07+12": VOC2007과 VOC2012 trainval의 결합. "07+12+COCO": COCO train val 35k에서 첫 학습이 다음 07+12에서 미세 조정됩니다.

물체가 클수록 더 결과가 좋습니다. 작은 물체들은 심지어 맨 위 층에 어떤 정보도 가지고 있지 않을 수 있기 때문에 이것은 놀라운 일이 아닙니다. 입력 크기(예: 300 * 300에서 512 * 512)를 늘리면 작은 물체를 탐지하는 데 도움이 될 수 있지만 아직 개선할 여지가 많습니다. 긍정적인 측면에서는 SSD가 대형 개체에서 매우 우수한 성능을 발휘한다는 것을 분명히 알 수 있습니다. 또한 특징맵 위치당 다양한 가로 세로 비율의 기본 상자를 사용하기 때문에 다양한 가로 세로 비율을 가진 물체에 매우 강력합니다.

3.2 모델 분석

SSD를 더 잘 이해하기 위해 제어된 실험을 수행하여 각 구성 요소가 성능에 어떤 영향을 미치는지 조사했습니다. 모든 실험에서 설정 또는 구성요소에 대한 지정된 변경 사항을 제외하고 동일한 설정 및 입력 크기(300 * 300)를 사용합니다.

	SSD300				
more data augmentation?		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔			✔	✔
use atrous?	✔	✔	✔		✔
VOC2007 `test` mAP	65.5	71.6	73.7	74.2	**74.3**

표 2: 다양한 설계 선택과 구성 요소가 SSD 성능에 미치는 영향

데이터 증산의 중요성 고속 및 고속 R-CNN은 원본 이미지와 수평 플립을 사용하여 학습합니다. 우리는 YOLO와 유사한 더 광범위한 샘플링 전략을 사용합니다. 표 2는 이 샘플링 전략으로 8.8%의 mAP를 개선할 수 있음을 보여줍니다. 우리의 샘플링 전략이 고속 및 고속 R-CNN에 얼마나 도움이 될지는 알 수 없지만, 설계별 객체 변환에 상대적으로 강력한 기능 풀링 단계를 사용하기 때문에 덜 도움이 될 가능성이 높습니다.

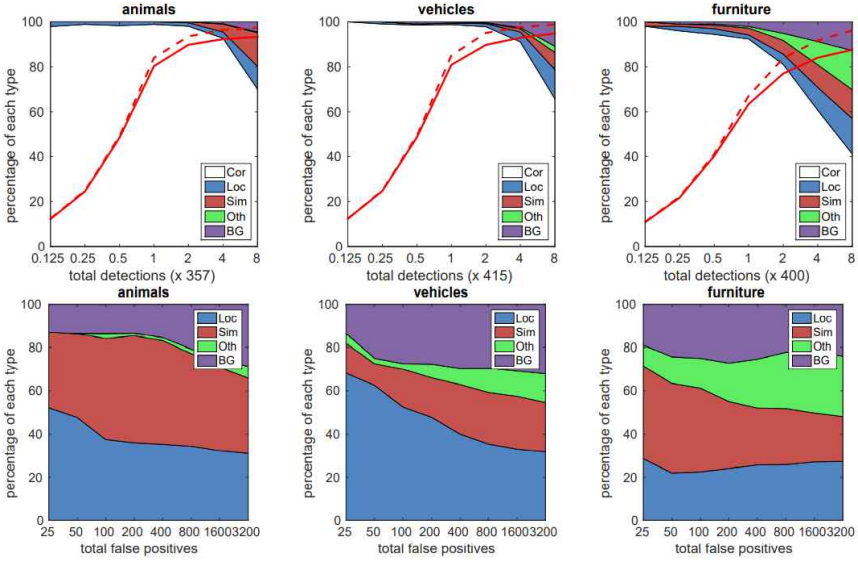

그림 3: VOC2007 테스트에서 동물, 차량 및 사물에 대한 SSD512의 성능의 시각화

위쪽 행은 잘못된 현지화(Loc), 유사한 범주와 혼동(Sim), 다른 범주와 혼동(Oth), 또는 배경(BG)으로 인해 정확함(Cor) 또는 부정확함 양성 탐지의 누적분율을 나타냅니다. 빨간색 실선은 탐지 수가 증가함에 따라 강력한 기준(0.5 jaccard 중첩)과 함께 리콜의 변화를 반영합니다. 빨간색 점선은 약한 기준(0.1 jaccard 중첩)을 사용합니다. 아래 행은 상위 순위의 잘못된 유형의 분포를 보여 줍니다.

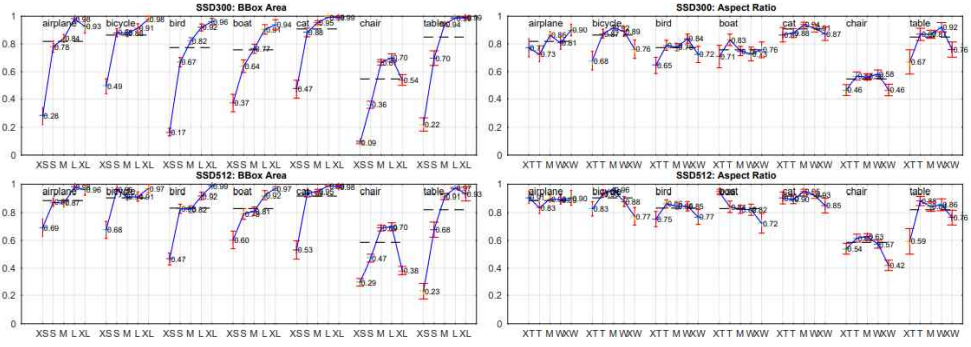

그림 4: VOC2007 테스트 세트에 대한 다양한 객체 특성의 민감도와 영향

왼쪽 그림은 범주별 BBox 면적의 효과를 나타내고 오른쪽 그림은 가로 세로 비율의 효과를 나타냅니다. 키: BBox 면적: XS=초소형, S=소형, M=중형, L=대형, XL=초대형입니다. 가로 세로 비율: XT=세로 높이/폭, T=세로 높이, M=중간, W=가로, XW=세로 폭입니다.

기본 상자 모양은 많을수록 좋다. 2.2장에 설명된 대로 기본적으로 위치당 6개의 기본 상자를 사용합니다. 가로 세로 비율이 1/3과 3인 상자를 제거하면 성능이 2.1% 더 떨어집니다. 다양한 기본 상자 모양을 사용하면 네트워크의 상자를 예측하는 작업이 쉬워집니다.

Atrous가 더 빠르다. 3장에서 설명한 대로 DeepLab-LargeFOV에 이어 서브샘플링된 VGG16의 atrous 버전을 사용했습니다. 풀5를 22초2로 유지하고 fc6와 fc7의 서브샘플링 매개 변수가 아닌 전체 VGG16을 사용하고 예측을 위해 conv53을 추가하면 결과는 거의 비슷한데 속도는 약 20% 느려졌습니다.

Prediction source layers from:						mAP use boundary boxes?		# Boxes
conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	Yes	No	
✔	✔	✔	✔	✔	✔	74.3	63.4	8732
✔	✔	✔	✔	✔		**74.6**	63.1	8764
✔	✔	✔	✔			73.8	68.4	8942
✔	✔	✔				70.7	69.2	9864
✔	✔					64.2	64.4	9025
	✔					62.4	64.0	8664

Table 3: **Effects of using multiple output layers.**

표 3: 다중 출력층을 사용했을 때의 영향.

해상도가 서로 다른 여러 출력 계층이 더 좋다. SSD의 주요 특징은 서로 다른 출력 계층에서 서로 다른 규모의 기본 상자를 사용하는 것입니다. 얻은 이점을 측정하기 위해 레이어를 점진적으로 제거하고 결과를 비교합니다. 공정한 비교를 위해 도면층을 제거할 때마다 기본 상자 타일링을 조정하여 총 상자 수를 원래 상자 수(8732)와 유사하게 유지합니다. 이 작업은 남은 층에 더 많은 크기의 상자를 쌓고 필요한 경우 상자의 크기를 조정하는 방식으로 수행됩니다. 우리는 각 설정에 대해 타일을 완전히 최적화하지 않습니다. 표 3은 74.3에서 62.4로 단조롭게 떨어지는 레이어 수에 따라 정확도가 감소하는 것을 보여줍니다. 여러 척도의 상자를 한 레이어에 쌓으면 많은 척도가 이미지 경계에 있으므로 신중하게 처리해야 합니다. 우리는 경계에 있는 상자를 무시하고 Faster R-CNN에 사용된 전략을 시도했습니다. 우리는 몇 가지 흥미로운 추세를 관찰했습니다. 예를 들어, 매우 거친 특징맵(예: conv11_2(1 * 1) 또는 conv10_2(3 * 3))을 사용하면 성능이 크게 저하됩니다. 그 이유는 가지치기 후 큰 물체를 덮을 수 있는 큰 상자가 충분하지 않기 때문일 수 있습니다. 주로 더 미세한 해상도 맵을 사용하면 충분한 수의 큰 상자가 남아 있기 때문에 성능이 다시 증가하기 시작합니다. 예측에 conv7만 사용한다면 성능이 최악이며, 이는 서로 다른 규모의 상자를 서로 다른 레이어에 분산시키는 것이 중요하다는 메시지를 강화합니다. 게다가, 우리의 예측은 ROI 풀링에 의존하지 않기 때문에, 저해상도 특징맵에서 축소 빈 문제가 발생하지 않습니다. SSD 아키텍처는 다양한 해상도의 특징맵에서 얻은 예측을 결합하여 더 낮은 해상도의 입력 이미지를 사용하는 동시에 더 빠른 R-CNN과 비슷한 정확도를 달성했습니다.

3.3 PASCAL VOC2012

SSD를 더 잘 이해하기 위해 제어된 실험을 수행하여 각 구성 요소가 성능에 어떤 영향을 미치는지 조사했습니다. 모든 실험에서 설정 또는 구성요소에 대한 지정된 변경 사항을 제외하고 동일한 설정 및 입력 크기(300 * 300)를 사용합니다. 우리는 VOC2012 trainval 및 VOC2007 trainval 및 test(21503 이미지)를 훈련에 사용하고 VOC2012 test(10991 이미지)에 대한 테스트를 사용한다는 점을 제외하고 위의 기본 VOC2007 실험에 사용된 설정과 동일한 설정을 사용합니다. 우리는 60k 반복에 대해 10^{-3}의 학습 속도로 모델을 교육한 다음 20k 반복에 대해 10^{-4}의 속도로 모델을 학습시켰습니다. 표 4에서 SSD300 및 SSD512 모델의 결과를 보여줍니다. VOC2007 테스트에서 관찰한 것과 동일한 성능 경향을 볼 수 있습니다. Seagate의 SSD300은 Fast/Fast R-CNN보다 정확도를 향상시킵니다. 교육 및 테스트 이미지 크기를 512로 늘림으로써 Fast R-CNN보다 4.5% 더 정확합니다. YOLO와 비교할 때 SSD는 여러 기능 맵의 컨볼루션 기본 상자를 사용하고 교육 중에 일치하는 전략을 사용했기 때문에 훨씬 더 정확합니다. COCO에 대해 교육을 받은 모델에서 미세 조정한 경우, Seagate SSD512는 80.0% mAP를 달성하며, 이는 Faster R-CNN보다 4.1% 높은 수치입니다.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast[6]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster[2]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
Faster[2]	07++12+COCO	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2
YOLO[5]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD300	07++12+COCO	77.5	90.2	83.3	76.3	63.0	53.6	83.8	82.8	92.0	59.7	82.7	63.5	89.3	87.6	85.9	84.3	52.6	82.5	**74.1**	**88.4**	74.2
SSD512	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
SSD512	07++12+COCO	**80.0**	**90.7**	**86.8**	**80.5**	**67.8**	**60.8**	**86.3**	**85.5**	**93.5**	**63.2**	**85.7**	**64.4**	**90.9**	**89.0**	**88.9**	**86.8**	**57.2**	**85.1**	72.8	**88.4**	**75.9**

표 4: PASCAL VOC2012 테스트 결과

고속 R-CNN은 최소 치수 600의 이미지를 사용하는 반면, YOLO의 이미지 크기는 448 * 448입니다. 데이터: "07++12": VOC2007 trainval과 test 그리고 VOC2012 trainval의 결합입니다. "07++12+COCO": COCO train val 35k에서 첫 번째 열차가 다음 07++12에서 미세 조정됩니다.

3.4 COCO

SSD 프레임워크를 더욱 검증하기 위해 우리는 COCO 데이터 세트에 대해 SSD300 및 SSD512 아키텍처를 교육했습니다. COCO의 객체는 PASCAL VOC보다 작은 경향이 있기 때문에 모든 레이어에 대해 더 작은 기본 상자를 사용합니다. 2.2절에서 언급한 전략을 따르지만, 이제 가장 작은 기본 상자의 척도는 0.2가 아니라 0.15이며, conv4_3의 기본 상자의 척도는 0.07입니다(예: 300개 이미지의 경우 21픽셀). 우리는 학습에 trainval35k를 사용합니다. 우리는 먼저 160k 반복에 대해 10^{-3}의 학습 속도로 모델을 학습시킨 다음 10^{-4}와 10^{-5} 반복에 대해 40k의 반복에 대해 학습을 계속합니다. 표 5는 test-dev2015의 결과를 보여줍니다. PASCAL VOC 데이터 세트에서 관찰한 것과 유사하게 SSD300은 mAP@0.5 및 mAP@[0.5:0.95] 모두에서 Fast R-CNN보다 우수합니다. SSD300은 ION 및 고속 R-CNN과 유사한 mAP@0.75를 가지고 있지만 mAP@0.5에서는 더 나쁩니다. Seagate의 SSD512는 이미지 크기를 512 * 512로 늘림으로써 두 기준 모두에서 Fast R-CNN보다 우수합니다. 흥미롭게도, SSD512는 mAP@0.75에서는 5.3% 더 낮지만 mAP@0.5에서는 1.2% 더 낮습니다. 또한 대형 물체에 대해서는 AP(4.8%)와 AR(4.6%)이 훨씬 우수하지만, AP(1.3%)와 AR(2.0%)은 상대적으로 덜 개선되었습니다.

Method	data	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
		0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast [6]	train	19.7	35.9	-	-	-	-	-	-	-	-	-	-
Fast [24]	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster [2]	trainval	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [24]	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster [25]	trainval	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300	trainval35k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512	trainval35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**

표 5: COCO test-dev2015 탐지 결과

작은 물체임. ION과 비교하여, 크고 작은 물체에 대한 AR 개선은 더 유사합니다(5.4% vs. 3.9%) 우리는 Fast R-CNN이 RPN 부분과 Fast R-CNN 부분 모두에서 두 가지 박스 정제 단계를 수행하기 때문에 SSD가 있는 작은 개체에서 더 경쟁력이 있다고 추측합니다. 그림 5에서는 SSD512 모델을 사용한 COCO 테스트 개발의 몇 가지 탐지 예를 보여 줍니다.

3.5 ILSVRC 예선 결과

COCO에 사용한 것과 동일한 네트워크 아키텍처를 ILSVRC DET 데이터 세트에 적용했습니다. 우리는 ILSVRC2014 DET 트레인 및 val1을 사용하여 SSD300 모델을 교육합니다. 우리는 먼저 320k 반복에 대해 10-3의 학습 속도로 모델을 교육한 다음 10-4 및 10-5 반복에 대해 40k의 반복에 대해 교육을 계속합니다. 우리는 val2 세트에서 43.4 mAP를 달성할 수 있습니다. 또한 SSD가 고품질 실시간 탐지를 위한 일반적인 프레임워크임을 검증합니다.

3.6 작은 개체의 정확도 향상을 위한 데이터 확장

Faster R-CNN에서와 같은 후속 기능 재샘플링 단계가 없으면 분석에서 보여지듯이 SSD에 대한 분류 작업이 상대적으로 어렵습니다(그림 4 참조). 2.2장에 설명된 데이터 증산 전략은 특히 PASCAL VOC와 같은 소규모 데이터 세트에서 성능을 획기적으로 향상시키는 데 도움이 됩니다. 이 전략에 의해 생성된 랜덤 작물은 "확대" 작업으로 간주될 수 있으며 더 큰 훈련 예를 생성할 수 있습니다. 더 작은 훈련 예를 만드는 "축소" 작업을 구현하기 위해 랜덤 자르기 작업을 수행하기 전에 먼저 평균값으로 채워진 원본 이미지 크기 16개의 캔버스에 이미지를 랜덤으로 배치합니다. 이 새로운 "확장" 데이터 증산 기술을 도입하여 더 많은 교육 이미지를 확보했기 때문에 교육 반복 횟수를 두 배로 늘려야 합니다. 표 6에 나와 있는 것처럼 여러 데이터셋에서 mAP가 지속적으로 2% ~ 3% 증가했습니다. 특히, 그림 6은 새로운 확장 트릭이 작은 개체의 성능을 크게 향상시킨다는 것을 보여줍니다. 이 결과는 최종 모델 정확도에 대한 데이터 확대 전략의 중요성을 강조합니다. SSD를 개선하는 또 다른 방법은 기본 박스의 타일링을 더 잘 설계하여 해당 위치와 스케일이 특징맵의 각 위치의 수용 필드에 더 잘 맞도록 하는 것입니다. 우리는 이것을 앞으로의 일을 위해 남겨둡니다.

그림 5: SSD512 모델을 사용한 COCO 테스트 개발의 탐지 예

우리는 0.6보다 높은 점수를 가진 탐지를 보여줍니다. 각 색상은 개체 범주에 해당합니다.

Method	VOC2007 test		VOC2012 test		COCO test-dev2015 trainval35k		
	07+12	07+12+COCO	07++12	07++12+COCO			
	0.5	0.5	0.5	0.5	0.5:0.95	0.5	0.75
SSD300	74.3	79.6	72.4	77.5	23.2	41.2	23.4
SSD512	76.8	81.6	74.9	80.0	26.8	46.5	27.8
SSD300*	77.2	81.2	75.8	79.3	25.1	43.1	25.8
SSD512*	**79.8**	**83.2**	**78.5**	**82.2**	**28.8**	**48.5**	**30.3**

표 6: 이미지 확장 데이터 증산 트릭을 추가하면 여러 데이터 세트에 결과가 나타남

SSD300* 및 SSD512*는 새로운 데이터 증대를 통해 교육된 모델입니다.

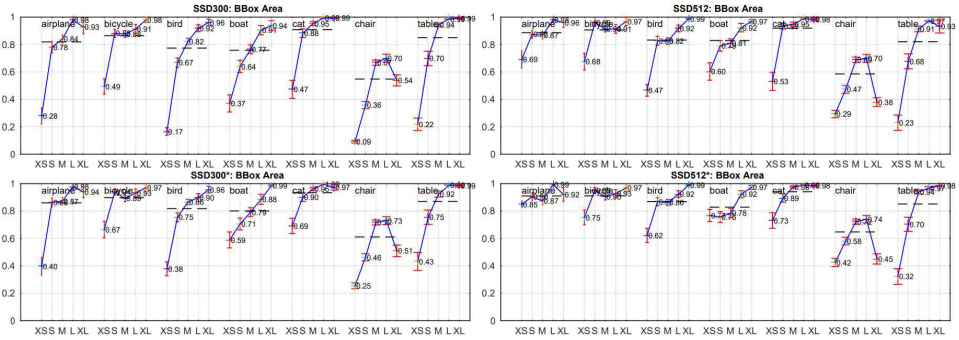

그림 6: VOC2007 테스트 세트에 대한 새로운 데이터 증강을 통한 개체 크기의 민감도 및 영향

맨 위 행은 원래 SSD300 및 SSD512 모델에 대한 범주별 BBox 영역의 효과를 나타내고, 맨 아래 행은 새로운 데이터 확대 트릭으로 훈련된 SSD300* 및 SSD512* 모델에 해당합니다. 새로운 데이터 확대 트릭이 작은 개체를 탐지하는 데 크게 도움이 된다는 것은 분명합니다.

3.7 예측 단계

우리의 방법에서 생성된 많은 박스 수를 고려할 때 추론 중에 최대가 아닌 억제(nms)를 효율적으로 수행하는 것이 필수적입니다. 0.01의 신뢰 임계값을 사용하면 대부분의 상자를 필터링할 수 있습니다. 그런 다음 클래스당 0.45의 jaccard 중첩이 있는 nms를 적용하고 이미지당 상위 200개의 탐지를 유지합니다. 이 단계는 SSD300 및 20개의 VOC 클래스에 대해 이미지당 약 1.7msec가 소요되며, 이는 새로 추가된 모든 계층에 소요되는 총 시간(2.4msec)에 가깝습니다. 우리는 Titan X와 Intel Xeon E5-2667v3@3.20GHz를 사용하는 cuDNN v4를 사용하여 배치 크기 8로 속도를 측정합니다. 표 7은 SSD, Faster R-CNN 및 YOLO의 비교를 보여줍니다. Seagate의 SSD300과 SSD512 방식은 모두 속도와 정확성 모두에서 Faster R-CNN을 능가합니다. Fast YOLO는 155 FPS에서 실행할 수 있지만 정확도는 거의 22% mAP만큼 낮습니다. 우리가 아는 한, SSD300은 70% 이상의 mAP를 달성한 최초의 실시간 방법입니다. 전송 시간의 약 80%가 기본 네트워크(이 경우 VGG16)에 사용됩니다. 따라서 더 빠른 기본 네트워크를 사용하면 속도가 더욱 향상되어 SSD512 모델을 실시간으로 만들 수 있습니다.

4. 관련 작업

이미지에서 객체 감지에는 슬라이딩 창을 기반으로 하는 방법과 지역 제안 분류를 기반으로 하는 방법 등 두 가지 클래스가 있습니다. 컨볼루션 신경망의 출현 이전에, 변형 가능한 부품 모델(DPM)과 선택적 검색이라는 두 가지 접근 방식에 대한 최첨단 기술은 비슷한 성능을 보였습니다. 그러나 선택적 검색 지역 제안과 사후 분류를 기반으로 한 컨볼루션 네트워크를 결합한 R-CNN이 가져온 극적인 개선 이후, 지역 제안 객체 감지 방법이 널리 보급되었습니다. 원래의 R-CNN 접근 방식은 다양한 방식으로 개선되었습니다. 첫 번째 접근 방식은 사후 분류의 품질과 속도를 향상시킵니다. 왜냐하면 다음과 같은 것들이 필요하기 때문입니다.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

표 7: PASCAL VOC2007 테스트 결과

70% 이상의 mAP를 달성할 수 있는 유일한 실시간 감지 방법은 SSD300입니다. 더 큰 입력 이미지를 사용함으로써 SSD512는 실시간에 가까운 속도를 유지하면서 정확도에서 모든 방법을 능가합니다.

수천 개의 이미지 결과에 대한 분류는 비용이 많이 들고 시간이 많이 소요됩니다. SPPnet은 원래의 R-CNN 접근 속도를 크게 향상시킵니다. 지역 크기와 규모에 더 강력한 공간 피라미드 풀링 레이어를 도입하고 분류 레이어가 여러 이미지 해상도에서 생성된 특징맵을 통해 계산된 피쳐를 재사용할 수 있습니다. Fast R-CNN은 학습 객관성을 위해 MultiBox에서 처음 도입된 신뢰도 및 경계 상자 회귀 분석 모두에 대한 손실을 최소화하여 모든 레이어를 종단 간 미세 조정할 수 있도록 SPPnet을 확장합니다.

두 번째 접근 방식은 심층 신경망을 사용하여 제안 생성 품질을 향상시킵니다. MultiBox와 같은 가장 최근의 작업에서, 낮은 수준의 이미지 기능을 기반으로 하는 선택적 검색 지역 제안은 별도의 심층 신경망에서 직접 생성된 제안으로 대체됩니다. 이것은 탐지 정확도를 더욱 향상시키지만 다소 복잡한 설정을 초래하여, 그들 사이의 의존성을 가진 두 개의 신경망을 훈련시켜야 합니다. 더 빠른 R-CNN은 선택적 검색 제안을 지역 제안 네트워크(RPN)에서 학습한 제안으로 대체하고, 이 두 네트워크에 대한 공유 컨볼루션 레이어와 예측 레이어를 번갈아 조정하여 RPN을 빠른 R-CNN과 통합하는 방법을 도입합니다. 이러한 방식으로 지역 제안은 중간 수준의 기능을 풀링하는 데 사용되며 최종 분류 단계는 비용이 적게 듭니다. SSD는 예측에 고정(기본) 상자 집합을 사용한다는 점에서 Faster R-CNN의 RPN(지역 제안 네트워크)과 매우 유사합니다. RPN의 앵커 상자와 유사합니다. 그러나 이러한 기능을 사용하여 기능을 풀링하고 다른 분류기를 평가하는 대신 각 상자의 각 개체 범주에 대한 점수를 동시에 생성합니다. 따라서, 우리의 접근 방식은 RPN을 Fast R-CNN과 병합하는 복잡성을 피하고, 훈련하기 쉽고, 빠르고, 다른 작업에 통합하기 쉽습니다.

우리의 접근 방식과 직접적인 관련이 있는 다른 방법 세트는 제안 단계를 모두 건너뛰고 여러 범주에 대한 경계 상자 및 신뢰도를 직접 예측합니다. 슬라이딩 창 방법의 심층 버전인 오버핏은 기본 개체 범주의 신뢰도를 알고 나면 최상위 특징맵의 각 위치에서 직접 경계 상자를 예측합니다. YOLO는 맨 위 형상도를 사용하여 여러 범주와 경계 상자(이러한 범주에 대해 공유됨)에 대한 신뢰도를 모두 예측합니다. 제안 단계는 없지만 기본 상자를 사용하기 때문에 SSD 방법이 이 범주에 속합니다. 그러나, 우리의 접근 방식은 다른 규모의 여러 특징맵에서 각 형상 위치에 서로 다른 가로 세로 비율의 기본 상자를 사용할 수 있기 때문에 기존 방법보다 더 유연합니다. 최상위 기능 맵에서 위치당 기본 상자 하나만 사용하는 경우 SSD는 OverFeat와 유사한 아키텍처를 갖게 됩니다. 최상위 기능 맵 전체를 사용하고 컨볼루션 예측기 대신 예측을 위해 완전히 연결된 계층을 추가하고 다중 가로 세로 비율을 명시적으로 고려하지 않으면 대략적으로 YOLO를 재현할 수 있습니다.

5. 결론

본 논문에서는 여러 범주에 대한 빠른 싱글샷 객체 감지기인 SSD를 소개합니다. 우리 모델의 핵심 특징은 네트워크 상단에 있는 여러 특징맵에 부착된 다중 스케일 컨볼루션 경계 상자 출력을 사용하는 것입니다. 이 표현을 통해 가능한 상자 모양의 공간을 효율적으로 모델링할 수 있습니다. 적절한 학습 전략이 주어지면 신중하게 선택한 기본 경계 상자의 수가 많을수록 성능이 향상된다는 것을 실험적으로 검증합니다. 우리는 기존 방법보다 최소 몇 배 더 많은 박스 예측 샘플링 위치, 규모 및 가로 세로 비율을 가진 SSD 모델을 구축합니다. 우리는 동일한 VGG-16 기본 아키텍처를 고려할 때 SSD가 정확도와 속도 모두에서 최첨단 물체 감지기보다 우수하다는 것을 보여줍니다. Seagate의 SSD512 모델은 PASCAL VOC 및 COCO의 정확도 측면에서 최첨단 Faster R-CNN을 능가하는 동시에 3배 더 빠릅니다. 당사의 실시간 SSD300 모델은 59 FPS로 실행되며, 이는 현재의 실시간 YOLO 대안보다 빠르며, 탐지 정확도는 월등히 뛰어납니다. 독립 실행형 유틸리티와 별개로, 우리는 단일하고 비교적 단순한 SSD 모델이 객체 감지 구성 요소를 사용하는 더 큰 시스템에 유용한 구성 요소를 제공한다고 믿습니다. 유망한 미래 방향은 비디오의 물체를 동시에 감지하고 추적하기 위해 반복적인 신경망을 사용하여 시스템의 일부로 그것의 사용을 탐구하는 것입니다.

6. 감사의 말

이 작업은 구글에서 인턴십 프로젝트로 시작되었고 UNC에서 계속되었습니다. 우리는 알렉스 토셰프에게 도움이 되는 토론을 해주셔서 감사드리며, 구글의 이미지 이해 팀과 디스트빌리프 팀에게 빚을 지고 있습니다. 우리는 또한 필립 암미라토와 패트릭 포어슨에게 도움이 되는 의견을 주셔서 감사합니다. NVIDIA가 GPU를 제공해 주셔서 감사드리며 NSF 1452851, 1446631, 1526367, 1533771의 지원에 감사드립니다.

참조

1. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV (2013)
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
4. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)
5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)
6. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
7. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)
8. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 v3 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
11. Hariharan, B., Arbela´ez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR. (2015)
12. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: Looking wider to see better. In: ILCR. (2016)
13. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. In: ICLR. (2015)
14. Howard, A.G.: Some improvements on deep convolutional neural network based image classification. arXiv preprint arXiv:1312.5402 (2013)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: NIPS. (2015)
16. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV (2015)
17. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR. (2015)
18. Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P.: A real-time algorithm for signal analysis with the help of the wavelet transform. In: Wavelets. Springer (1990) 286–297
19. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: MM. (2014)
20. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. (2010)
21. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV 2012. (2012)
22. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
23. Zhang, L., Lin, L., Liang, X., He, K.: Is faster r-cnn doing well for pedestrian detection. In: ECCV. (2016)
24. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR. (2016)
25. COCO:Common Objects in Context.(2016) [Online; accessed 25-July-2016].
26. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, de- formable part model. In: CVPR. (2008)