마스크 R-CNN

Kaiming He Georgia Gkioxari Piotr Dollar Ross Girshick

페이스북 AI 리서치(FAIR)

추상적인

우리는 개념적으로 단순하고 유연하며 일반적인
객체 인스턴스 분할을 위한 프레임워크. 우리의 접근
이미지의 물체를 효율적으로 감지하는 동시에
각 입장에 대해 고품질 분할 마스크를 생성합니다. Mask R-CNN이라는 방법
은 더 빠르게 확장됩니다.
R-CNN에서 객체 마스크를 예측하기 위한 분기를 추가하여
경계 상자 인식을 위해 기존 분기와 병렬입니다. Mask R-CNN은 훈련이 간단
하고 적은 양만 추가합니다.
5fps에서 실행되는 Faster R-CNN에 대한 오버헤드. 더구나,
Mask R-CNN은 다른 작업으로 일반화하기 쉽습니다. 예를 들어 동일한 프레
임워크에서 사람의 자세를 추정할 수 있습니다.
COCO 제품군의 세 트랙 모두에서 최고의 결과를 보여줍니다.
인스턴스 분할, 경계 상자 객체 감지 및 사람 키포인트 감지를 비롯한 여러 문
제를 해결합니다. 종소리와 휘파람이 없으면 Mask R-CNN은 다음을 포함한
모든 작업에서 기존의 모든 단일 모델 항목을 능가합니다.

COCO 2016 챌린지 우승자. 우리는 우리의 단순하고
효과적인 접근 방식은 견고한 기준선과 도움이 될 것입니다.
인스턴스 수준 인식에서 향후 연구를 용이하게 합니다. 암호
https://github.com/ 에서 사용할 수 있습니다.
페이스북리서치/디텍트론.

1. 소개

비전 커뮤니티는 단기간에 객체 감지 및 의미론적 분할 결과를 빠르게 개선
했습니다. 대부분 이러한 발전을 주도했습니다.

Fast/Faster R CNN [12, 36] 및 FCN(Fully Convolutional Network)
[30] 과 같은 강력한 기준 시스템에 의해
각각 객체 감지 및 의미론적 분할을 위한 프레임워크입니다. 이러한 방법은 개
념적으로 직관적입니다.
빠른 훈련 및 추론 시간과 함께 유연성과 견고성을 제공합니다. 이 작업에서
우리의 목표는 개발하는 것입니다.
인스턴스 분할을 위한 프레임워크를 비교적 가능하게 합니다.

인스턴스 세분화는 다음이 필요하기 때문에 어렵습니다.
이미지의 모든 물체를 정확하게 감지하는 동시에
각 인스턴스를 정확하게 분할합니다. 따라서 결합한다
개체 감지의 고전적인 컴퓨터 비전 작업의 요소. 목표는 경계 상자를 사용하
여 개별 개체를 분류하고 각 개체를 지역화하는 것입니다.

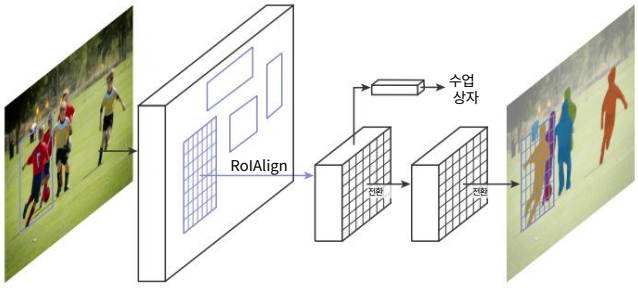

그림 1. 인스턴스 분할을 위한 Mask R-CNN 프레임워크.

각 픽셀을 다음으로 분류하는 것이 목표인 세분화
입장에서 대상을 구별하지 않고 고정된 범주 집합입니다.1 이를 감안할 때 복
잡한 방법을 기대할 수 있습니다.
좋은 결과를 얻으려면 필요합니다. 그러나 우리는 그것을 보여줍니다
놀랍도록 간단하고 유연하며 빠른 시스템은
이전의 최첨단 인스턴스 분할 결과.

Mask R-CNN이라고 하는 우리의 방법은 Faster R-CNN을 확장합니다.
[36] 분할 마스크 예측을 위한 분기 추가
각 관심 영역(RoI)에서 분류 및 경계 상자 회귀를 위한 기존 분기와 병렬로(그
림 1). 마스크 브랜치는 작은 FCN이 적용된

각 RoI에 대해 픽셀 대 픽셀 방식으로 분할 마스크를 예측합니다. Mask R-
CNN은 구현이 간단하고
Faster R-CNN 프레임워크가 주어졌을 때 훈련
다양한 유연한 아키텍처 설계. 추가적으로,
마스크 분기는 약간의 계산 오버헤드만 추가합니다.
빠른 시스템과 빠른 실험을 가능하게 합니다.

원칙적으로 Mask R-CNN은 직관적인 확장입니다.
더 빠른 R-CNN, 아직 마스크 분기를 올바르게 구성
좋은 결과를 위해 중요합니다. 가장 중요한 것은 Faster R CNN은 네트워크
입력과 출력 간의 픽셀 대 픽셀 정렬을 위해 설계되지 않았다는 것입니다. 이것
은 에서 가장 분명하다.
RoIPool [18, 12], 인스턴스를 처리할 때의 사실상 핵심 연산이 대략적인 공
간 양자화를 수행하는 방법
특징 추출을 위해 오정렬을 수정하기 위해 RoIAlign이라고 하는 단순하고 양
자화가 없는 레이어를 제안합니다.
정확한 공간 위치를 충실히 보존합니다. 에도 불구하고

1일반적인 용어에 따라 객체 감지를 사용하여
마스크가 아닌 경계 상자를 통한 감지 및 의미론적 분할
인스턴스를 구별하지 않고 픽셀별 분류를 나타냅니다. 그래도 우리는
인스턴스 분할은 의미론적이며 감지의 한 형태라는 점에 유의하십시오.

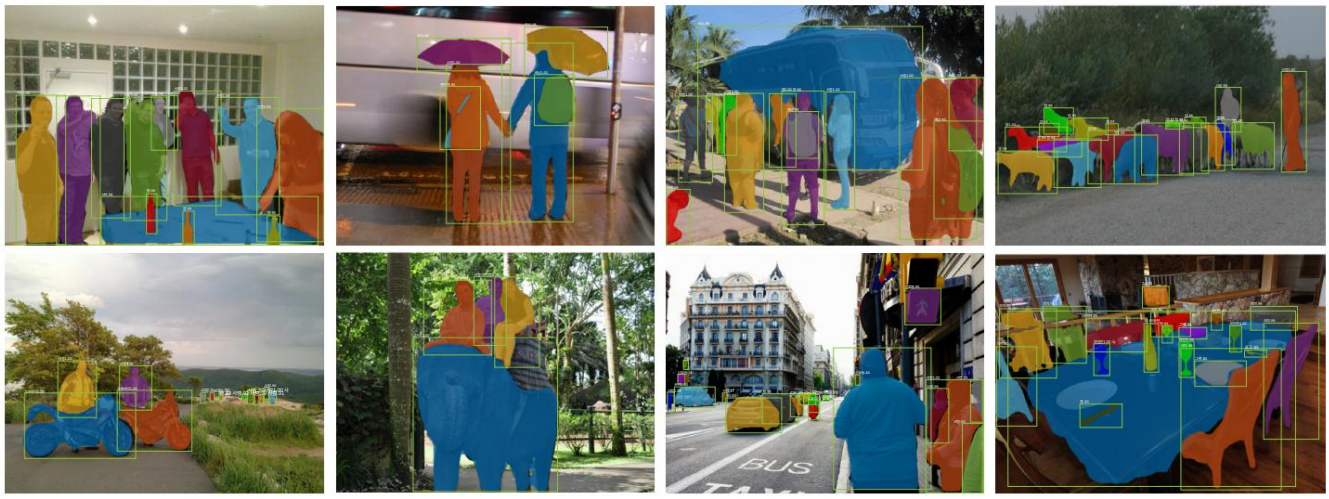

그림 2. COCO 테스트 세트에서 마스크 R-CNN 결과. 이 결과는 ResNet-101 [19] 을 기반으로 하며 35.7의 마스크 AP를 달성하고 5fps에서 실행됩니다. 마스크는 색상으로 표시되며 경계 상자, 범주 및 신뢰도도 표시됩니다.

겉보기에 사소한 변경이지만 RoIAlign은 큰 영향을 미칩니다. 마스크 정확도를 상대적으로 10%에서 50% 향상시켜 더 엄격한 현지화 측정 기준에서 더 큰 이득을 보여줍니다. 둘째, 마스크와 클래스 예측을 분리하는 것이 필수적이라는 것을 발견했습니다. 클래스 간의 경쟁 없이 독립적으로 각 클래스에 대한 이진 마스크를 예측하고 네트워크의 RoI 분류 분기에 의존하여 범주를 예측합니다. 대조적으로, FCN은 일반적으로 분할과 분류를 결합하는 픽셀당 다중 클래스 분류를 수행하며 우리의 실험에 따르면 인스턴스 분할에서는 제대로 작동하지 않습니다.

종소리와 휘파람이 없으면 Mask R-CNN 은 2016년 대회 우승자의 고도로 엔지니어링된 항목을 포함하여 COCO 인스턴스 분할 작업 [28]에서 이전의 모든 최첨단 단일 모델 결과를 능가합니다 . 부산물로서 우리의 방법은 COCO 물체 감지 작업에서도 탁월합니다. 절제 실험에서 우리는 여러 기본 인스턴스화를 평가하여 견고성을 입증하고 핵심 요소의 효과를 분석할 수 있습니다.

우리 모델은 GPU에서 프레임당 약 200ms로 실행할 수 있으며 COCO에 대한 교육은 단일 8-GPU 시스템에서 1~2일이 소요됩니다. 우리는 프레임워크의 유연성 및 정확성과 함께 빠른 훈련 및 테스트 속도가 인스턴스 분할에 대한 향후 연구에 도움이 되고 용이할 것이라고 믿습니다.

마지막으로 COCO 키 포인트 데이터 세트 [28] 에서 인간 포즈 추정 작업을 통해 프레임워크의 일반성을 보여줍니다 . 각 키포인트를 원-핫 바이너리 마스크로 보고 최소한의 수정으로 Mask R-CNN을 적용하여 인스턴스별 포즈를 감지할 수 있습니다. Mask R-CNN은 2016 COCO keypoint 경쟁의 우승자를 능가함과 동시에 5fps로 실행됩니다. 따라서 Mask R-CNN은 인스턴스 수준 인식을 위한 유연한 프레임워크로 더 광범위하게 볼 수 있으며 더 복잡한 작업으로 쉽게 확장될 수 있습니다.

향후 연구를 용이하게 하기 위해 코드를 릴리스했습니다.

2. 관련 업무

R-CNN: 경계 상자 객체 감지에 대한 지역 기반 CNN(R-CNN) 접근 [13] 은 관리 가능한 수의 후보 객체 영역에 주의를 기울이고 [42, 20] 컨볼루션 네트워크 [25, 24]를 평가하는 것입니다.] 각 ROI에 대해 독립적입니다. R-CNN이 확장되어 [18, 12] RoIPool을 사용하여 기능 맵에서 RoI에 주의를 기울일 수 있으므로 빠른 속도와 정확도가 향상됩니다. Faster R-CNN [36] 은 RPN(Region Proposal Network)으로 주의 메커니즘을 학습하여 이 스트림을 발전시켰습니다. Faster R-CNN은 많은 후속 개선 사항(예: [38, 27, 21])에 대해 유연하고 견고하며 여러 벤치마크에서 현재 최고의 프레임워크입니다.

인스턴스 분할: R CNN의 효율성에 힘입어 인스턴스 분할에 대한 많은 접근 방식은 세그먼트 제안을 기반으로 합니다. 이전 방법 [13, 15, 16, 9]은 상향식 세그먼트 [42, 2]로 분류되었습니다. DeepMask [33] 및 후속 작업 [34, 8]은 세그먼트 후보 날짜를 제안한 다음 Fast R-CNN에 의해 분류되는 방법을 학습합니다. 이러한 방법에서는 분할이 인식보다 우선하므로 느리고 정확도가 떨어집니다. 마찬가지로, Dai et al. [10] 은 경계 상자 제안에서 세그먼트 제안을 예측한 후 분류하는 복잡한 다단계 캐스케이드를 제안했습니다.

대신 우리의 방법은 마스크와 클래스 레이블의 병렬 예측을 기반으로 하며 이는 더 간단하고 유연합니다.

가장 최근에는 Li et al. [26] 은 "완전 컨볼루션 인스턴스 분할"(FCIS)을 위해 [8] 의 세그먼트 제안 시스템과 [11] 의 객체 감지 시스템을 결합했습니다. [8, 11, 26] 의 일반적인 아이디어 는 위치 감지 출력 채널 세트를 완전히 컨볼루션으로 예측하는 것입니다. 이러한 채널은 객체 클래스, 상자 및 마스크를 동시에 처리하여 시스템을 빠르게 만듭니다. 그러나 FCIS는 중복되는 인스턴스에 대해 시스템적 오류를 나타내고 가상 에지를 생성하여(그림 6), 인스턴스를 분할하는 근본적인 어려움에 직면해 있음을 보여줍니다.

인스턴스 분할에 대한 또 다른 솔루션 제품군 [23, 4, 3, 29] 은 의미론적 분할의 성공에 의해 주도됩니다. 픽셀별 분류 결과에서 시작(예:
FCN 출력), 이러한 방법은 픽셀을 자르려고 시도합니다.
같은 카테고리를 다른 인스턴스로. 대조적으로
이러한 방법의 segmentation-first 전략, Mask R-CNN
인스턴스 우선 전략을 기반으로 합니다. 우리는 두 전략의 기업이 앞으로 더 깊이 연구될 것으로 기대합니다.

3. 마스크 R-CNN

Mask R-CNN은 개념적으로 간단합니다. Faster R-CNN은
각 후보 객체에 대한 두 개의 출력, 클래스 레이블 및
경계 상자 오프셋; 여기에 개체 마스크를 넣는 세 번째 분기를 추가합니다. 따라서 Mask R-CNN은 자연스럽고 직관적인 아이디어입니다. 그러나 추가 마스크 출력은
훨씬 더 미세한 추출이 필요한 클래스 및 상자 출력
객체의 공간 레이아웃. 다음으로 픽셀 대 픽셀 정렬을 포함한 Mask R-CNN의 핵심 요소를 소개합니다.
Fast/Faster R-CNN의 주요 누락 부분입니다.

Faster R-CNN: Faster를 간략하게 검토하는 것으로 시작합니다 .
R-CNN 검출기 [36]. Faster R-CNN은 두 단계로 구성됩니다.
RPN(Region Proposal Network)이라고 하는 첫 번째 단계는
후보 객체 경계 상자를 제안합니다. 두번째
본질적으로 Fast R-CNN [12]인 단계는 각 후보 상자에서 RoIPool을 사용하여 특성을 추출하고 수행합니다.
분류 및 경계 상자 회귀. 특징
더 빠른 추론을 위해 두 단계에서 사용되는 데이터를 공유할 수 있습니다. 우리
최신의 포괄적인 비교를 위해 독자에게 [21] 참조
Faster R-CNN과 다른 프레임워크 사이.

Mask R-CNN: Mask R-CNN은 동일한 2단계를 채택
동일한 첫 번째 단계(RPN)가 있는 절차입니다. 에
두 번째 단계, 클래스 및 상자 예측과 병행
offset, Mask R-CNN은 각각의 바이너리 마스크를 출력합니다.
투자 수익 이것은 분류가 마스크 예측에 의존하는 가장 최근의 시스템과 대조적입니다(예: [33, 10, 26]).
우리의 접근 방식은 Fast R-CNN [12] 의 정신을 따릅니다.
경계 상자 분류 및 회귀를 병렬로 적용합니다(다단계를 크게 단순화하는 것으로 판명됨).
원래 R-CNN의 파이프라인 [13]).

공식적으로 교육 중에 우리는 다중 작업 손실을 다음과 같이 정의합니다.
각각의 샘플링된 RoI는 L = Lcls + Lbox + Lmask입니다. 분류 손실 Lcls 및
경계 상자 손실 Lbox 는 [12]에 정의된 것과 동일합니다. 마스크 분기에는 Km2 가 있습니다 .
K 바이너리를 인코딩하는 각 RoI에 대한 차원 출력
m × m 해상도의 마스크, K 클래스 각각에 대해 하나씩.
이를 위해 픽셀당 시그모이드를 적용하고 Lmask 를 다음과 같이 정의합니다.
평균 이진 교차 엔트로피 손실. 관련된 ROI의 경우
실측 클래스 k에서 Lmask 는 k번째에만 정의됩니다.
마스크(다른 마스크 출력은 손실에 기여하지 않음).

Lmask 의 정의 는 네트워크가 다음을 생성하도록 허용합니다.
클래스 간 경쟁 없이 클래스별 마스크
우리는 예측하기 위해 전용 분류 분기에 의존합니다.

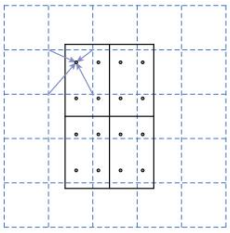

그림 3. RoIAlign: 점선 그리드는 기능 맵을 나타내고
실선은 RoI를 나타냅니다.
(이 예에서는 2×2 저장소 사용), 점
각 빈의 4개 샘플링 지점. RoIAlign
각 샘플링 포인트의 값을 계산합니다.
가까운 그리드에서 쌍선형 보간법으로
기능 맵의 포인트. 양자화 없음
관련된 모든 좌표에서 수행
RoI, 해당 빈 또는 샘플링 지점.

출력 마스크를 선택하는 데 사용되는 클래스 레이블입니다. 이것은 분리
마스크 및 클래스 예측. 이것은 일반과 다릅니다
FCN [30] 을 의미론적 분할에 적용할 때의 연습 은 일반적으로 픽셀당 softmax와 다항 교차 엔트로피 손실을 사용합니다. 이 경우 클래스 간 마스크

경쟁하다; 우리의 경우 픽셀당 시그모이드 및 바이너리
손실, 그들은하지 않습니다. 우리는 이 공식이 좋은 인스턴스 분할 결과의 핵심이라는 것을 실험을 통해 보여줍니다.

마스크 표현: 마스크는 입력 객체의
공간 레이아웃. 따라서 클래스 레이블이나 상자 오프셋과 달리
필연적으로 짧은 출력 벡터로 축소됩니다.
완전 연결(fc) 레이어, 공간 구조 추출
픽셀 대 픽셀로 마스크를 자연스럽게 처리할 수 있습니다.
회선에 의해 제공되는 통신.

구체적으로, 각 RoI에서 m × m 마스크를 예측합니다.
FCN 사용 [30]. 이렇게 하면 마스크의 각 레이어가
벡터 표현으로 축소하지 않고 명시적 m × m 객체 공간 레이아웃을 유지하기 위한 분기
공간적 차원이 부족하다. 마스크 예측을 위해 fc 레이어로 다시 정렬하는 이전 방법 [33, 34, 10]과 달리,
컨볼루션 표현은 더 적은 수의 매개변수를 필요로 하며,
실험에 의해 입증된 바와 같이 더 정확합니다.

이 픽셀 대 픽셀 동작에는 RoI 기능이 필요합니다.
자체적으로 잘 정렬된 작은 기능 맵입니다.
명시적인 픽셀당 공간 대응성을 충실하게 보존합니다. 이것은 우리가 다음을 개발하도록 동기를 부여했습니다.
마스크 예측에서 핵심적인 역할을 하는 RoIAlign 레이어.

RoIAlign: RoIPool [12] 은 각 RoI에서 작은 특징 맵(예: 7x7)을 추출하기 위한 표준 작업입니다. 로이풀
먼저 부동 숫자 RoI를 피쳐 맵의 이산 입도로 양자화하고, 이 양자화된 RoI는 자체적으로 양자화되는 공간 빈으로 세분됩니다.

마지막으로 각 빈에 포함된 기능 값이 집계됩니다.
(일반적으로 최대 풀링에 의해). 양자화가 수행된다. 예를 들어,
[x/16]을 계산하여 연속 좌표 x에서
16은 기능 맵 보폭이고 [·]는 반올림입니다. 마찬가지로 bin(예: 7×7)으로 나눌 때 양자화를 수행한다.
이러한 양자화는
ROI 및 추출된 특징. 영향을 미치지 않을 수 있지만
소규모 번역에 강력한 분류
픽셀 정확도 마스크 예측에 큰 부정적인 영향.

이를 해결하기 위해 RoIPool의 거친 양자화를 제거하고 적절하게 정렬하는 RoIAlign 레이어를 제안합니다.
입력으로 추출된 특징. 제안된 변경 사항
간단합니다. RoI 경계의 양자화를 피합니다.

또는 bins(즉, [x/16] 대신 x/16을 사용함). 우리는 양방향 선형 보간법 [22] 을 사용하여 각 RoI 빈의 정기적으로 샘플링된 4개 위치에서 입력 기능의 정확한 값을 계산하고 결과를 집계합니다(최대 또는 평균 사용). 자세한 내용은 그림 3 을 참조하세요. 양자화가 수행되지 않는 한 결과는 정확한 샘플링 위치 또는 샘플링된 포인트 수에 민감하지 않습니다.

RoIAlign은 §4.2에서 볼 수 있듯이 크게 개선되었습니다 . 또한 [10] 에서 제안한 RoIWarp 연산과 비교합니다 . RoIAlign과 달리 RoIWarp 는 정렬 문제를 간과하고 RoIPool과 마찬가지로 RoI를 양자화 하는 것으로 [10] 에서 구현되었습니다. 따라서 RoIWarp도 [22]에 의해 동기 부여된 이중 선형 리샘플링을 채택하지만 실험(표 2c에서 더 자세한 내용)에서 볼 수 있듯이 RoIPool과 동등하게 수행하여 정렬의 중요한 역할을 보여줍니다.

네트워크 아키텍처: 접근 방식의 일반성을 보여주기 위해 다중 아키텍처로 Mask R-CNN을 인스턴스화합니다. 명확성을 위해 (i) 전체 이미지에 대한 특징 추출에 사용되는 컨볼루션 백본 아키텍처와 (ii) 경계 상자 인식(분류 및 회귀) 및 각각에 별도로 적용되는 마스크 예측을 위한 네트워크 헤드를 구분합니다. 투자 수익

우리는 명명법 네트워크 깊이 기능을 사용하여 백본 아키텍처를 나타냅니다. 깊이 50 또는 101 레이어의 ResNet [19] 및 ResNeXt [45] 네트워크를 평가합니다. ResNet을 사용한 Faster R-CNN의 원래 구현 [19] 은 C4라고 하는 4단계의 최종 컨볼루션 계층에서 특징을 추출했습니다. 예를 들어 ResNet-50이 있는 이 백본은 ResNet-50-C4로 표시됩니다. 이것은 [19, 10, 21, 39]에서 사용되는 일반적인 선택입니다.

우리는 또한 최근에 Lin et al.에 의해 제안된 또 다른 보다 효과적인 백본을 탐구합니다. [27], FPN(Feature Pyra mid Network)이라고 합니다. FPN은 측면 연결이 있는 하향식 아키텍처를 사용하여 단일 규모 입력에서 네트워크 내 기능 피라미드를 구축합니다. FPN 백본을 사용하는 더 빠른 R-CNN은 규모에 따라 피쳐 피라미드의 다양한 수준에서 RoI 피쳐를 추출하지만 나머지 접근 방식은 바닐라 ResNet과 유사합니다. Mask R CNN과 함께 특징 추출을 위해 ResNet-FPN 백본을 사용하면 정확도와 속도 모두에서 탁월한 이점을 얻을 수 있습니다. FPN에 대한 자세한 내용은 독자에게 [27]을 참조하십시오.

네트워크 헤드의 경우 우리는 완전 컨볼루션 마스크 예측 분기를 추가하는 이전 작업에서 제시된 아키텍처를 밀접하게 따릅니다. 특히, 우리는 ResNet [19] 및 FPN [27] 논문에서 Faster R-CNN 상자 헤드를 확장합니다. 자세한 내용은 그림 4에 나와 있습니다. ResNet-C4 백본의 헤드에는 계산 집약적인 ResNet의 5번째 단계(즉, 9계층 'res5' [19])가 포함됩니다. FPN의 경우 백본에는 이미 res5가 포함되어 있으므로 더 적은 수의 필터를 사용하는 더 효율적인 헤드가 가능합니다.

우리의 마스크 브랜치는 간단한 구조를 가지고 있습니다. 더 복잡한 디자인은 성능을 향상시킬 가능성이 있지만 이 작업의 초점은 아닙니다.

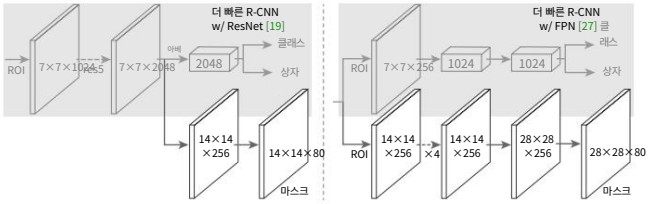

그림 4. 헤드 아키텍처: 기존 Faster R CNN 헤드 2개를 확장합니다 [19, 27]. 왼쪽/오른쪽 패널 은 마스크 분기가 추가된 각각 [19] 및 [27] 의 ResNet C4 및 FPN 백본에 대한 헤드를 보여줍니다 . 숫자는 공간 해상도와 채널을 나타냅니다. 화살표는 컨텍스트에서 추론할 수 있는 conv, deconv 또는 fc 레이어를 나타냅니다(conv는 공간 차원을 유지하지만 deconv는 공간 차원을 증가시킴). 출력 conv는 1x1이고 deconv는 2x2이고 stride 2이며 숨겨진 레이어에서 ReLU [31] 를 사용하는 것을 제외하고 모든 변환은 3x3입니다. 왼쪽: 'res5'는 ResNet의 다섯 번째 단계를 나타냅니다. 단순화를 위해 첫 번째 변환이 보폭 1([19]에서와 같이 14×14 / 보폭 2 대신)이 있는 7×7 RoI에서 작동하도록 변경했습니다. 오른쪽: '×4'는 4개의 연속 전환 스택을 나타냅니다.

3.1. 구현 세부 정보

기존 Fast/Faster R-CNN 작업 [12, 36, 27] 에 따라 하이퍼파라미터를 설정했습니다 . 이러한 결정은 원본 논문 [12, 36, 27]에서 객체 감지에 대해 이루어졌지만 인스턴스 분할 시스템이 이에 대해 강력하다는 것을 알았습니다.

훈련: Fast R-CNN에서와 같이, RoI는 최소 0.5의 ground-truth box와 함께 IoU가 있으면 양수로 간주되고 그렇지 않으면 음수로 간주됩니다. 마스크 손실 Lmask 는 양의 RoI에서만 정의됩니다. 마스크 타겟은 RoI와 관련 실측 마스크 사이의 교차점입니다.

우리는 이미지 중심 교육을 채택합니다 [12]. 이미지는 크기(짧은 가장자리)가 800픽셀이 되도록 크기가 조정됩니다 [27]. 각 미니 배치에는 GPU당 2개의 이미지가 있고 각 이미지에는 N 샘플링된 RoI가 있으며 포지티브 대 네거티브의 비율이 1:3입니다 [12]. N은 C4 백본의 경우 64([12, 36]에서와 같이)이고 FPN의 경우([27]에서와 같이) 512입니다. 우리는 160,000번의 반복을 위해 8개의 GPU(유효한 미니 배치 크기는 16)에서 훈련하고 120,000번의 반복에서 10만큼 감소하는 0.02의 학습률을 사용합니다. 0.0001의 가중치 감쇠와 0.9의 운동량을 사용합니다. ResNeXt [45]를 사용 하여 GPU당 1개의 이미지와 0.01의 시작 학습률로 동일한 반복 횟수로 훈련합니다.

RPN 앵커는 [27]에 따라 5개의 스케일과 3개의 종횡비에 걸쳐 있습니다. 편리한 절제를 위해 RPN은 별도로 훈련되며 덜 지정되지 않는 한 Mask R-CNN과 기능을 공유하지 않습니다. 이 문서의 모든 항목에 대해 RPN과 Mask R-CNN은 동일한 백본을 가지고 있으므로 공유 가능합니다.

추론: 테스트 시간에 제안 번호는 C4 백본의 경우 300([36]) , FPN의 경우 1000([27])입니다. 우리는 이러한 제안에 대해 상자 예측 분기를 실행한 다음 최대가 아닌 억제를 실행합니다 [14]. 그런 다음 마스크 분기가 가장 높은 점수를 받은 100개의 감지 상자에 적용됩니다. 이것은 훈련에 사용되는 병렬 계산과 다르지만 추론 속도를 높이고 정확도를 향상시킵니다(더 적고 더 정확한 RoI를 사용하기 때문에). 마스크 브랜치

그림 5. ResNet-101-FPN을 사용하고 35.7 마스크 AP로 5fps로 실행한 COCO 테스트 이미지에 대한 Mask R-CNN 의 추가 결과(표 1).

	등뼈	AP	AP50	AP75	APS	APM	APL
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5 확장	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5 확장	33.6	54.5	-	-	-	-
마스크 R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
마스크 R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
마스크 R-CNN	ResNeXt-101-FPN	37.1	60.0	39.4	16.9	39.9	53.5

표 1. COCO test-dev의 인스턴스 분할 마스크 AP MNC [10] 및 FCIS [26] 는 COCO 2015 및 2016의 우승자입니다.
세분화 문제. 종소리와 휘파람이 없으면 Mask R-CNN은 다음을 포함하는 더 복잡한 FCIS+++보다 성능이 뛰어납니다.
다중 스케일 트레인/테스트, 수평 플립 테스트 및 OHEM [38]. 모든 항목은 단일 모델 결과입니다.

RoI당 K개의 마스크를 예측할 수 있지만 k번째 마스크만 사용합니다.
여기서 k는 분류 분기에 의해 예측된 클래스입니다.
그런 다음 m×m 부동 숫자 마스크 출력의 크기가 다음과 같이 조정됩니다.
RoI 크기 및 0.5의 임계값에서 이진화됩니다.

상위 100위의 마스크만 계산하므로
Mask R-CNN은 탐지 상자에 약간의 오버헤드를 추가합니다.
더 빠른 R-CNN 대응물(예: 일반 모델에서 ~20%).

4. 실험: 인스턴스 분할

우리는 Mask R-CNN과의 철저한 비교를 수행합니다.
포괄적인 절제와 함께 최신 기술
COCO 데이터 세트 [28]. AP(IoU 임계값에 대한 평균), AP50,

AP75 및 APS, APM, APL (다른 척도의 AP). AP는 IoU 마스크를 사용하여 평가하고 있습니다. 이전과 같이
work [5, 27], 우리는 80k 기차 이미지의 합집합을 사용하여 훈련합니다
및 35k val 이미지의 하위 집합(trainval35k) 및 나머지 5k val 이미지
(minival)에 대한 절제를 보고합니다.
test-dev [28]에서도 결과를 보고합니다.

4.1. 주요 결과

우리는 Mask R-CNN을 표 1의 인스턴스 분할에서 최신 방법과 비교합니다. 우리 모델의 모든 인스턴스화는 이전 최신 모델의 기준 변형보다 성능이 뛰어납니다. 여기에는 MNC가 포함됩니다 [10]

및 FCIS [26], COCO 2015 및 2016 수상자
세분화 문제. 종소리와
휘파람, ResNet-101-FPN 백본으로 R-CNN 마스크
다중 스케일을 포함하는 FCIS+++ [26] 보다 우수한 성능
훈련/테스트, 수평 플립 테스트 및 온라인 하드 예제 마이닝(OHEM) [38].
이 작업의 범위를 벗어나는 동안 우리는
이러한 많은 개선 사항이 당사에 적용될 것으로 기대합니다.

마스크 R-CNN 출력은 그림 2 와 5 에 시각화되어 있습니다.
Mask R-CNN은 어려운 조건에서도 좋은 결과를 얻습니다. 그림 6 에서는
Mask R-CNN을 비교합니다 .
기준선 및 FCIS+++ [26]. FCIS++는 체계적인 전시
중복되는 인스턴스에 대한 인공물을 발견하여 인스턴스 분할의 근본적인 어려움으로 인해 어려움을 겪고 있음을 시사합니다. Mask R-CNN은 그러한 아티팩트를 보여주지 않습니다.

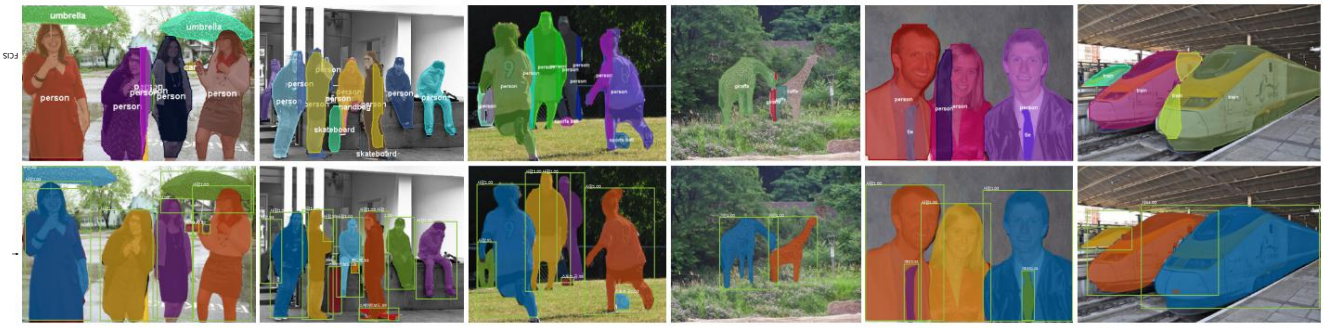

그림 6. FCIS+++ [26] (위)와 Mask R-CNN(아래, ResNet-101-FPN). FCIS는 겹치는 물체에 체계적인 인공물을 전시합니다.

네트워크 깊이 기능	AP	AP50	AP75
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	36.7	59.5	38.9

(a) 백본 아키텍처: 더 나은 백본은 예상되는 이득을 가져옵니다: 더 깊은 네트워크 더 잘하고, FPN이 C4 기능을 능가하며, ResNeXt는 ResNet에서 개선되었습니다.

	AP	AP50	AP75
소프트맥스	24.8	44.1	25.1
시그모이드	30.3	51.2	31.5
	+5.5	+7.1	+6.4

(b) 다항식 대 독립 마스크 (ResNet-50-C4): 클래스별 바이너리 마스크(sigmoid)를 통한 디커플링은 큰 다항 마스크(softmax)보다 이득.

	맞추다?	쌍선형?	애그	AP	AP50	AP75
로이풀 [12]			최대	26.9	48.8	26.4
로이워프 [10]			X 최대	27.2	49.2	27.1
			X 아베	27.1	48.9	27.1
RoIAlign	XX	최대		30.2	51.0	31.8
	XX	평균		30.3	51.2	31.5

(c) RoIAlign (ResNet-50-C4): 다양한 RoI로 결과를 마스킹 레이어. RoIAlign 레이어는 AP를 ~3포인트 향상시키고 AP75 ~5포인트 . 적절한 정렬을 사용하는 것이 RoI 레이어 간의 큰 간격에 기여하는 유일한 요소입니다.

	AP	AP50	AP75	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	30.9 +7.3	51.8	32.1	34.0	55.3	36.4
		+5.3	+10.5	+5.8	+2.6	+9.5

(d) RoIAlign (ResNet-50-C5, stride 32): 마스크 수준 및 상자 수준 대규모 기능을 사용하는 AP. 부정교합이 더 심하다. stride-16 기능(표 2c)을 사용하면 정확도 차이가 커집니다.

	마스크 분기 fc:	AP	AP50	AP75
MLP	1024→1024→80·282	31.5	53.7	32.8
MLP FC:	1024 → 1024 → 1024 → 80 · 282	31.5	54.0	32.6
FCN	전환수: 256→256→256→256→256→80	33.6	55.2	35.3

(e) 마스크 분기 (ResNet-50-FPN): 완전 컨볼루션 네트워크(FCN) 대 마스크 예측을 위한 다층 퍼셉트론(MLP, 완전 연결). FCN은 공간 레이아웃을 명시적으로 인코딩하므로 결과를 개선합니다.

표 2. 절제. 우리는 trainval35k에서 훈련하고, minival에서 테스트하고, 달리 명시되지 않는 한 마스크 AP를 보고합니다.

4.2. 절제 실험

우리는 Mask R-CNN을 분석하기 위해 많은 절제를 실행합니다. 결과는 표 2 에 나와 있으며 다음에 자세히 설명합니다.

아키텍처: 표 2a 는 다양한 마스크 R-CNN을 보여줍니다 .

등뼈. 더 깊은 네트워크의 이점을 얻습니다(50 대 101). FPN 및 ResNeXt를 포함한 고급 설계. 우리 모든 프레임워크가 자동으로 혜택을 받는 것은 아닙니다. 심층 또는 고급 네트워크([21]의 벤치마킹 참조).

Multinomial vs. Independent Masks: Mask R-CNN de 커플 마스크와 클래스 예측: 기존 상자로 분기가 클래스 레이블을 예측하고 각각에 대한 마스크를 생성합니다. 클래스 간의 경쟁이 없는 클래스(픽셀당 시그모이드 및 이진 손실). 표 2b에서 우리는 이것을 다음과 비교합니다. 픽셀당 softmax 및 다항 손실을 사용합니다(FCN [30]에서 일반적으로 사용됨). 이 대안은 작업을 결합합니다. 마스크 및 클래스 예측의 결과로 심각한 손실 마스크 AP에서 (5.5 포인트). 이것은 입장이 전체로 분류되면(상자 분기에 의해),

에 대한 걱정 없이 이진 마스크를 예측하는 것으로 충분합니다. 모델을 더 쉽게 학습시킬 수 있는 카테고리입니다.

Class-Specific vs. Class-Agnostic Masks: 인스턴스화의 기본값은 클래스별 마스크, 즉 1m×m을 예측합니다. 클래스별 마스크. 흥미롭게도, 클래스 불가지론 마스크가 있는 Mask R-CNN(즉, 클래스에 관계없이 단일 m×m 출력 예측)은 거의 효과적입니다. 29.7 마스크 AP가 있습니다. vs. ResNet-50-C4의 클래스별 대응물에 대한 30.3. 이것은 우리의 접근 방식에서 노동 분업을 더욱 강조합니다. 분류와 세분화를 크게 분리합니다.

RoIAlign: 제안된 RoIAlign 레이어에 대한 평가는 다음과 같습니다. 표 2c에 나와 있습니다. 이 실험에서는 보폭이 16인 ResNet 50-C4 백본을 사용합니다. AP가 RoIPool에 비해 약 3점 높으며 많은 이점이 있습니다. 높은 IoU (AP75)로 제공됩니다. RoIAlign은 최대/평균 풀; 나머지 논문에서는 평균을 사용합니다.

또한 에서 제안한 RoIWarp와 비교합니다. 쌍선형 샘플링도 채택한 MNC [10] . 논의 § 3에서 RoIWarp 는 여전히 RoI를 양자화하여 정렬을 잃습니다. 입력으로. 표 2c 에서 볼 수 있듯이 RoIWarp per form은 RoIPool과 동등하고 RoIAlign보다 훨씬 나쁩니다. 이것은 적절한 정렬이 핵심임을 강조합니다.

우리는 또한 32픽셀의 훨씬 더 큰 보폭을 가진 ResNet-50-C5 백본으로 RoIAlign을 평가합니다. 우리는 사용 res5 헤드가 아니므로 그림 4 (오른쪽) 와 동일한 헤드 해당되는. 표 2d 는 RoIAlign이 마스크를 개선함을 보여줍니다. AP 7.3포인트, AP75 마스크 10.5포인트

	등뼈	APbb	APbb$_{50}$	APbb$_{75}$	APbb$_S$	APbb$_M$	APbb$_L$
더 빠른 R-CNN++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
더 빠른 R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
G-RMI에 의한 더 빠른 R-CNN [21]	Inception-ResNet-v2 [41]	34.7	55.5	36.7	13.5	38.1	52.0
더 빠른 R-CNN w TDM [39]	시작-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
더 빠른 R-CNN, RoIAlign	ResNet-101- FPN	37.3	59.6	40.3	19.8	40.2	48.8
마스크 R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
마스크 R-CNN	ResNeXt-101-FPN	39.8	62.3	43.4	22.1	43.2	51.2

표 3. 객체 감지 단일 모델 결과(경계 상자 AP)와 test-dev의 최신 기술 비교 ResNet-101-을 사용하여 R-CNN 마스크
FPN은 이전의 모든 최첨단 모델의 기본 변형을 능가합니다(이 실험에서는 마스크 출력이 무시됨). 의 이득
Mask R-CNN over [27] 은 RoIAlign(+1.1 APbb), 멀티태스킹 훈련(+0.9 APbb) 및 ResNeXt-101(+1.6 APbb)을 사용하여 생성됩니다.

(상대적 개선 50%). 또한, 우리는
stride-32 C5 기능(30.9 AP)을 사용하는 RoIAlign은 stride-16 C4 기
능(30.3 AP, 표 2c)을 사용하는 것보다 더 정확합니다.
RoIAlign의 오랜 과제를 크게 해결합니다.
탐지 및 분할을 위해 큰 보폭 기능을 사용합니다.
마지막으로 RoIAlign은 1.5 마스크 AP와 0.5의 이득을 보여줍니다.
FPN과 함께 사용할 때 더 미세한 다중 레벨이 있는 상자 AP
보폭. 더 미세한 정렬이 필요한 키포인트 감지의 경우,
RoIAlign은 FPN에서도 큰 이득을 보입니다(표 6).

마스크 분기: 분할은 픽셀 대 픽셀 작업이며
우리는 FCN을 사용하여 마스크의 공간 레이아웃을 활용합니다.
표 2e에서 다층 퍼셉트론(MLP)을 비교합니다 .
ResNet-50-FPN 백본을 사용하는 FCN. FCN 사용
MLP에 비해 2.1 마스크 AP 이득을 제공합니다. 우리는 우리가
이 백본을 선택하여 FCN의 변환 레이어가
헤드는 MLP와 공정한 비교를 위해 사전 훈련되지 않았습니다.

4.3. 경계 상자 감지 결과

Mask R-CNN을 최첨단 COCO와 비교합니다.
표 3의 경계 상자 객체 감지. 이 결과의 경우,
전체 Mask R-CNN 모델을 학습하더라도
분류 및 상자 출력은 추론에 사용됩니다(
마스크 출력은 무시됨). ResNet-101-을 사용하여 R-CNN 마스크
FPN 은 COCO 2016 탐지 챌린지에서 우승한 G RMI [21] 의 단일 모델
변형을 포함하여 이전의 모든 최첨단 모델의 기본 변형을 능가합니다 .
ResNeXt-101-FPN을 사용하여 Mask R-CNN은 3.0포인트 상자 AP 이
상의 여백으로 결과를 더욱 향상시킵니다.

[39] 의 최고의 이전 단일 모델 항목 (사용
시작-ResNet-v2-TDM).
추가 비교를 위해 우리는 Mask 버전을 훈련했습니다.
R-CNN이지만 마스크 분기 없이 "더 빠르게
R-CNN, RoIAlign'이 표 3에 나와 있습니다. 이 모델이 더 나은 성능 을 보입니다.
RoIAlign으로 인해 [27] 에 제시된 모델보다 에
반면 Mask R-CNN보다 0.9포인트 박스 AP가 낮다.
따라서 상자 감지에서 마스크 R-CNN의 이러한 간격은 다음과 같습니다.
멀티태스킹 교육의 이점만을 누릴 수 있습니다.
마지막으로 Mask R-CNN은 작은 간격을 얻습니다.
마스크와 상자 AP 사이: 예: 37.1 사이의 2.7 포인트
(마스크, 표 1) 및 39.8 (상자, 표 3). 이것은 다음을 나타냅니다.
우리의 접근 방식은 객체 감지와 더 까다로운 인스턴스 분할 작업 사이의
간격을 크게 좁힙니다.

4.4. 타이밍

추론: 우리는 공유하는 ResNet-101-FPN 모델을 훈련합니다 .
Faster R-CNN [36] 의 4단계 훈련에 따라 RPN과 Mask R-CNN 단계
사이의 기능 . 이 모델
Nvidia Tesla M40 GPU에서 이미지당 195ms로 실행(플러스
15ms CPU 시간은 출력을 원래 해상도로 크기 조정하고 통계적으로 동일
한 마스크 AP를 달성합니다.
공유되지 않은 것. 우리는 또한 ResNet-101-C4 변형이 더 무거운 박스
헤드를 가지고 있기 때문에 ~400ms가 걸린다고 보고합니다(그림 4).
실제로 C4 변형을 사용하지 않는 것이 좋습니다.
Mask R-CNN은 빠르지만 우리의 디자인은
속도에 최적화되어 있지 않으며 , 예를 들어 다양한 이미지 크기 및

제안 번호는 이 백서의 범위를 벗어납니다.

훈련: Mask R-CNN은 훈련 속도도 빠릅니다. 교육
COCO trainval35k의 ResNet-50-FPN은 32시간이 걸립니다.
동기화된 8-GPU 구현에서(16-당 0.72초
이미지 미니 배치), ResNet-101-FPN 사용 시 44시간. 에
사실, 빠른 프로토타이핑은 하루 이내에 완료될 수 있습니다.
기차 세트에서 훈련할 때. 우리는 이러한 신속한 훈련이 이 분야의 주요 장
애물을 제거하고 격려하기를 바랍니다.
더 많은 사람들이 이 도전적인 주제에 대한 연구를 수행할 수 있습니다.

5. 인체 포즈 추정을 위한 마스크 R-CNN

우리의 프레임워크는 인간 포즈로 쉽게 확장될 수 있습니다.
견적. 우리는 keypoint의 위치를 one-hot으로 모델링합니다.
마스크하고 K개의 마스크를 예측하기 위해 Mask R-CNN을 채택합니다.
각각의 K 키포인트 유형(예: 왼쪽 어깨, 오른쪽 팔꿈치).
이 작업은 Mask R-CNN의 유연성을 입증하는 데 도움이 됩니다.
우리는 인간 포즈에 대한 최소한의 도메인 지식이
실험은 주로
Mask R-CNN 프레임워크의 일반성을 보여줍니다.
도메인 지식(예: 모델링 구조 [6])이 간단한 접근 방식을 보완할 것으로 기
대합니다.

구현 세부 정보: 다음 을 약간 수정합니다.
키포인트에 맞게 조정할 때 세분화 시스템.
인스턴스의 각 K 키포인트에 대해 훈련
target은 one-hot m × m 바이너리 마스크입니다.
픽셀은 전경으로 레이블이 지정됩니다. 훈련하는 동안 각 가시적 진실 키포
인트에 대해 교차 엔트로피를 최소화합니다.
m2 -way softmax 출력 에 대한 손실 (이는

그림 7. Mask R-CNN(ResNet-50-FPN)을 사용한 COCO 테스트의 Keypoint 검출 결과, Person Segmentation Mask 예측
같은 모델에서. 이 모델은 63.1의 키포인트 AP를 가지며 5fps로 실행됩니다.

	APK _ 50	APK 75	APK 중	APK 웰
CMU-포즈+++ [6]	61.8 84.9 67.5 57.1 68.2			
G-RMI [32] † 마	62.4 84.0 68.5 59.1 68.1			
스크 R-CNN, 키포인트 전용 마스크 R-	62.7 87.0 68.4 57.4 71.1			
CNN, 키포인트 및 마스크 63.1 87.3 68.7 57.8 71.4				

표 4. COCO test-dev의 키포인트 감지 AP 우리는
5fps에서 실행되는 단일 모델(ResNet-50-FPN). CMU-포즈++
[6] 은 다중 규모 테스트를 사용하는 2016년 대회 우승자이며,
CPM [44] 으로 후처리 하고 물체 감지기로 필터링하여 누적 ~5점 추가 (개인 커뮤에 명
시됨: G-RMI는 COCO + MPII [1] (25k imnation). 연령), 두 가지 모델 사용(경계 상
†자용 Inception-ResNet-v2

감지 및 키포인트용 ResNet-101).

감지할 단일 지점). 인스턴스 분할에서와 같이 K 키포인트는 여전히 독립적
으로 처리됩니다.

ResNet-FPN 변형을 채택하고 키포인트 헤드
아키텍처는 그림 4 (오른쪽)와 유사합니다. 키 포인트 헤드는 8개의 3×3
512-d 변환 레이어 스택과 deconv 레이어와 2x 쌍선형 업스케일링으로
구성됩니다.
56×56의 출력 해상도를 생성합니다. 우리는 그것을 발견했다
마스크에 비해 상대적으로 고해상도 출력
키포인트 수준의 현지화 정확도에 필요합니다.

모델은 주석이 달린 키포인트가 포함된 모든 COCO trainval35k 이
미지에서 훈련됩니다. 과적합을 줄이기 위해 이 훈련 세트가 더 작기 때문
에 이미지를 사용하여 훈련합니다.
[640, 800] 픽셀에서 무작위로 샘플링된 스케일; 추론
800픽셀의 단일 스케일에 있습니다. 우리는 90,000번의 반복을 위해 훈련합니다.
0.02의 학습률에서 시작하여 10으로 줄입니다.
60k 및 80k 반복. 우리는 경계 상자 NMS를 사용합니다.
임계값 0.5. 기타 사항은 3.1항과 동일합니다 .

주요 결과 및 절제: 사람 키 포인트 AP (APkp) 를 평가하고 ResNet-50-
FPN으로 실험합니다.
등뼈; 부록에서 더 많은 백본을 연구할 것입니다.
표 4 는 우리의 결과(62.7 APkp) 가 0.9포인트 더 높음 을 보여줍니다.
COCO 2016 키포인트 탐지 우승자 [6] 보다
다단계 처리 파이프라인을 사용합니다(표 4의 캡션 참조). 우리의 방법은
훨씬 더 간단하고 빠릅니다.

더 중요한 것은, 우리는 si

	APbb 사람	AP마스크 사람	APK
더 빠른 R-CNN	52.5	-	-
마스크 R-CNN, 마스크 전용	53.6	45.8	-
마스크 R-CNN, 키포인트 전용	50.7	-	64.2
마스크 R-CNN, 키포인트 및 마스크	52.0	45.1	64.7

표 5. Box, Mask, Keypoint의 Multi-task 학습
사람 범주, 최소값으로 평가됨. 모든 항목이 교육되었습니다.
공정한 비교를 위해 동일한 데이터에 대해 백본은 ResNet 50-FPN입니다. minival에
64.2 및 64.7 AP가 있는 항목은

test-dev AP는 각각 62.7 및 63.1입니다(표 4 참조).

	APK _ 50	APK 75	APK 중	APK 웰
로이풀	59.8 86.2 66.7 55.1 67.4			
RoIAlign	64.2 86.6 69.7 58.7 73.0			

표 6. 키포인트 감지를 위한 RoIAlign 대 RoIPool
미니어처. 백본은 ResNet-50-FPN입니다.

동시에 상자, 세그먼트 및 키포인트를 예측합니다.
5fps로 실행됩니다. 세그먼트 분기(개인 범주용)를 추가하면 APkp 가
63.1(표 4) 로 향상됩니다.
테스트 개발 다중 작업 학습의 더 많은 절제
minival은 표 5 에 나와 있습니다.
box-only(즉, Faster R-CNN) 또는 keypoint-only 버전
지속적으로 이러한 작업을 개선합니다. 그러나 추가
키포인트 분기는 상자/마스크 AP를 약간 줄여 키포인트 감지가 멀티태스킹
의 이점을 제공함을 시사합니다.
다른 작업에는 도움이 되지 않습니다. 그럼에도 불구하고 세 가지 작업을
모두 학습하면 통합 시스템이 가능합니다.
모든 출력을 동시에 효율적으로 예측합니다(그림 7).

우리는 또한 키포인트에 대한 RoIAlign의 영향을 조사합니다.
검출(표 6). 이 ResNet-50-FPN 백본
더 미세한 보폭(예: 가장 정밀한 수준에서 4픽셀), RoIAlign
여전히 RoIPool에 비해 상당한 개선을 보여주고 APkp 를 4.4포인트 증
가시킵니다. 이는 키포인트 감지가 현지화 정확도에 더 민감하기 때문입니
다. 이 또
마스크와 키포인트를 포함한 픽셀 수준 현지화에 정렬이 필수적임을 나타
냅니다.

추출을 위한 Mask R-CNN의 효과를 감안할 때
개체 경계 상자, 마스크 및 키포인트
다른 인스턴스 수준 작업을 위한 효과적인 프레임워크가 됩니다.

	트레이닝 데이터	AP [val]	AP50	인	라이더			차	트럭	버스	기차	mcycle	자전거	
InstanceCut [23]	미세 + 거친	15.8	27.9	10.0	8.0	23.7	14.0	19.5	15.2	9.3	4.7			
DWT [4]	19.8	13.0												
SAIS [17]	좋아		15.6	30.0	15.1	11.7	32.9	17.1	20.4	15.0	7.9	4.9		
	좋아	-	17.4	36.7	14.6	12.9	35.7	16.0	23.2	19.0	10.3	7.8		
DIN [3]	미세 + 거친	-	20.0	38.8	16.5	16.7	25.7	20.6	30.0	23.4	17.1	10.1		
SGN [29]	미세 + 거친	29.2	25.0	44.9	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4		
마스크 R-CNN	파인	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0		
마스크 R-CNN	파인 + 코코	36.4	32.0	58.1	34.8	27.0	49.1	30.1	40.9	30.9	24.1	18.7		

표 7. Cityscapes val('AP [val]' 열) 및 테스트(나머지 열) 집합에 대한 결과. 우리의 방법은 ResNet-50-FPN을 사용합니다.

부록 A: 도시경관에 대한 실험

우리는 추가로 인스턴스 세분화 결과를 보고합니다.
도시경관 [7] 데이터세트. 이 데이터 세트에는 2975개의 기차, 500개 발 및 1525
개의 테스트 이미지에 대한 정밀한 주석이 있습니다. 그것은 가지고있다
인스턴스 주석이 없는 20k 거친 훈련 이미지,
우리는 사용하지 않습니다. 모든 이미지는 2048×1024 픽셀입니다.
인스턴스 분할 작업에는 8개의 객체 범주가 포함됩니다.
미세 훈련 세트의 인스턴스 수는 다음과 같습니다.

사람	라이더	차	트럭	버스	기차	mcycle	자전거
17.9k							
1.8k	26.9k	0.5k		0.4k	0.2k	0.7k	3.7k

이 작업에 대한 인스턴스 분할 성능이 측정됩니다.
COCO 스타일 마스크 AP(IoU 임계값에 대한 평균); AP50 (즉, 0.5의 IoU
에서 마스크 AP)도 보고됩니다.

구현: 우리는 Mask R-CNN 모델을 다음과 같이 적용합니다 .
ResNet-FPN-50 백본; 101층을 찾았습니다
대응하는 데이터 세트 크기가 작기 때문에 유사하게 수행됩니다.
무작위로 샘플링된 이미지 스케일(짧은 쪽)으로 훈련합니다.
과적합을 줄이는 [800, 1024]; 추론이 켜져 있습니다
1024픽셀의 단일 스케일. 우리는 미니 배치 크기를 사용합니다.
GPU당 이미지 1개(GPU 8개에서 8개) 및 모델 학습
0.01의 학습률에서 시작하여 24,000회 반복에 대해
18k 반복에서 0.001로 줄입니다. ~4시간 소요
이 설정에서 단일 8-GPU 머신에서 훈련합니다.

결과: 표 7 은 우리의 결과를
val 및 테스트 세트에 대한 예술. 거친 것을 사용하지 않고
훈련 세트, 우리의 방법은 테스트에서 26.2 AP를 달성합니다.
이전 최고 항목(DIN [3])에 비해 30% 이상 상대적으로 개선되었으며 동시
작업보다 우수합니다.
SGN의 25.0 [29]. DIN 및 SGN 모두 미세 + 거친 사용
데이터. 미세 데이터만을 사용한 베스트 엔트리와 비교
(17.4 AP), 우리는 ~50% 개선을 달성했습니다.
사람 및 자동차 카테고리의 경우 Cityscapes 데이터 세트
많은 수의 범주 내 중첩 자세가 나타납니다(이미지당 평균 6명과 9대의 자동
차). 우리
범주 내 중복이 입각세 세분화의 핵심 어려움이라고 주장합니다. 우리의 방법
은 다른 최고의 항목에 비해 이 두 범주에서 엄청난 개선을 보여줍니다(21.8
에서 30.5로 사람에 대한 상대적 약 40% 개선 및

39.4에서 46.9로 자동차에서 ~20% 개선), 그럼에도 불구하고
우리의 방법은 거친 데이터를 이용하지 않습니다.
Cityscapes 데이터 세트의 주요 과제는 훈련입니다.
특히 범주에 대한 낮은 데이터 영역의 모델
약 200-500 기차가있는 트럭, 버스 및 기차

그림 8. Cityscapes 테스트의 Mask R-CNN 결과(32.0 AP).
오른쪽 하단 이미지는 실패 예측을 보여줍니다.

ing 샘플 각각. 이 문제를 부분적으로 해결하기 위해 더 나아가
COCO 사전 교육을 사용하여 결과를 보고합니다. 이를 위해 우리는
Cityscapes에서 해당 7개 카테고리를 초기화합니다.
사전 훈련된 COCO Mask R-CNN 모델(라이더가 domly 초기화됨). 4k 반
복을 위해 이 모델을 미세 조정합니다.
여기서 학습률은 3k 반복에서 감소합니다.
COCO 모델이 주어지면 훈련에 ~1시간 이 걸립니다 .
COCO 사전 훈련된 Mask R-CNN 모델은
테스트에서 32.0 AP, 거의 6포인트 향상
벌금 전용 상대. 이것은 중요한 역할을 나타냅니다
훈련 데이터 재생량. 또한 다음과 같이 제안합니다.
Cityscapes의 방법은 낮은 샷 학습 성능의 영향을 받을 수 있습니다.
COCO 사전 교육을 사용하는 것이 이 데이터 세트에 대한 효과적인 전략임
을 보여줍니다.
마지막으로 val과 test 사이의 편향을 관찰했습니다.
AP, [23, 4, 29] 의 결과에서도 관찰됩니다 . 우리
이러한 편견이 주로 트럭, 버스,
그리고 fine-only 모델이 있는 기차 카테고리
val/test AP는 각각 28.8/22.8, 53.5/32.2, 33.0/18.6입니다. 이는 도메
인 이동이 있음을 나타냅니다.
훈련 데이터가 거의 없는 이러한 범주. 머리
사전 훈련은 이러한 범주에 대한 결과를 가장 잘 개선하는 데 도움이 됩니다.
그러나 도메인 이동은 38.0/30.1에서 지속되며,
각각 57.5/40.9 및 41.2/30.9 val/test AP. 메모
사람 및 자동차 카테고리에 대해서는
이러한 바이어스(val/test AP는 ±1 포인트 이내).
Cityscapes에 대한 예시 결과는 그림 8에 나와 있습니다.

설명 원래 기준	등뼈	AP AP50 AP75 APbb APbb	50 75 APbb
선 X-101-FPN + 업데이트된 기준선 X-101-FPN + e2e 교육 X-101-FPN + ImageNet-5k X-101-FPN + 증대 시간 X-101-FPN + 더 깊은 X-152-FPN + 논로컬 [43] X-152-FPN-NL 40.3 64.4 42.8 45.0 67.8 48.9	X-101-FPN	36.7 59.5 38.9 37.0 59.7 39.0 40.5 63.0 43.7 37.6 60.4 39.9 41.7 64.1 45.2 38.6 61.7 40.9 42.7 65.1 46.6 39.2 62.5 41.6 43.5 65.9 47.2 39.7 63.2 42.2 44.1 66.4 48.4	39.6 61.5 43.2
+ 테스트 시간 augm. X-152-FPN-NL 41.8 66.0 44.8 47.3 69.3 51.5			

표 8. COCO에서 Mask R-CNN의 향상된 검출 결과
미니어처. 각 행은 위의 행에 추가 구성요소를 추가합니다.
표기의 간결함을 위해 ResNeXt 모델을 'X'로 표시합니다.

설명 원래 기준	백본 APkp _	50	APK 75	APK 중	APK 그
선 R-50-FPN 64.2 + 업데이트된 기준선 R-50-FPN 65.1	R-50-FPN	86.6	69.7	58.7	73.0
+ 더 깊은 R-101-FPN 66.1 + ResNeXt-101-FPN 67.3	R-101-FPN	86.6	70.9	59.9	73.6
FPN 69.1 + 테스트 시간 8월	X-101-FPN	87.7	71.7	60.5	75.0
		88.0	73.3	62.2	75.6
		88.9	75.3	64.1	77.1
	X-101-FPN 70.4 89.3		76.8	65.8	78.1

표 9. COCO에서 Mask R-CNN의 향상된 키포인트 결과
미니어처. 각 행은 위의 행에 추가 구성요소를 추가합니다.
여기서는 키포인트 주석만 사용하고 마스크 주석은 사용하지 않습니다.
간결함을 위해 ResNet을 'R'로 표시하고 ResNeXt를 'X'로 표시합니다.

부록 B: COCO에 대한 향상된 결과

일반적인 프레임워크로서 Mask R-CNN은
빠르고/빠른 R-CNN 및 FCN. 이 부록에서는 원래 결과보다 개선된 몇 가지 기술에 대해 설명합니다. 일반성과 유연성 덕분에 Mask R-CNN
에서 우승한 세 팀의 프레임워크로 사용되었습니다.
COCO 2017 인스턴스 분할 대회,
모두 이전의 최신 기술을 훨씬 능가했습니다.

인스턴스 분할 및 객체 감지

우리는 표 8에서 Mask R-CNN의 일부 개선된 결과를 보고합니다. 전반적으로 개선은 마스크 AP 5.1을 증가시킵니다.
포인트(36.7에서 41.8로) 및 박스 AP 7.7 포인트(39.6에서
47.3). 각 모델 개선은 마스크 AP를 모두 증가
및 상자 AP를 일관되게 보여주며,
마스크 R-CNN 프레임워크 우리는 개선 사항을 자세히 설명합니다
다음. 이 결과는 향후 업데이트와 함께 https://github.com/ 에서 릴리스된 코드로 재현할 수 있습니다.
페이스북리서치/디텍트론, 역할을 할 수 있습니다
미래 연구를 위한 더 높은 기준선.

업데이트된 기준선: 업데이트된 기준선으로 시작합니다.
다른 하이퍼파라미터 세트로 우리는 연장
학습률이 120k 및 160k 반복에서 10만큼 감소하는 180k 반복으로 훈련합니다. 우리도 변한다
NMS 임계값을 0.5로 설정합니다(기본값 0.3에서). 그만큼
업데이트된 베이스라인에는 37.0 마스크 AP와 40.5 박스 AP가 있습니다.

End-to-end training: 이전의 모든 결과는 stage wise training,
즉 RPN을 첫 번째 단계로 훈련하고 Mask를 사용했습니다.
두 번째로 R-CNN. [37]에 이어 RPN과 Mask R CNN을 공동으로 훈련하는 end-to-end('e2e') 훈련을 평가합니다. 우리는 [37] 에서 '대략적인' 버전을 채택합니다 .
기울기 wrt RoI 좌표를 무시하여 RoIAlign 레이어의 부분 기울기를 계산합니다.
표 8 은 다음을 보여줍니다.
e2e 훈련은 마스크 AP를 0.6, 박스 AP를 1.2로 향상시킵니다.

ImageNet-5k 사전 훈련: [45]에 따라 ImageNet의 5k 클래스 하위 집합에서 사전 훈련된 모델을 실험합니다(표준 1k 클래스 하위 집합과 대조). 이 5×
사전 훈련 데이터의 증가는 마스크와 상자 1 모두를 향상시킵니다.
AP. 참고로 [40] 은 ~250배 더 많은 이미지(300M)를 사용했습니다.
그리고 베이스라인에서 2-3박스 AP 개선을 보고했습니다.

열차 시간 증강: 열차에서의 스케일 증강
시간은 결과를 더욱 향상시킵니다. 훈련 중에 무작위로
[640, 800] 픽셀에서 스케일을 샘플링하고
260k까지의 반복 횟수(학습률 감소
200k 및 240k 반복에서 10만큼). Train-time Augmentation은 마스크 AP를 0.6, 박스 AP를 0.8 향상시킵니다.

모델 아키텍처: 101-layer 업그레이드
ResNeXt는 152-layer 대응물 [19]에 대해, 우리는 관찰합니다.
0.5 마스크 AP와 0.6 박스 AP가 증가합니다. 이것은 보여줍니다
더 깊은 모델은 여전히 COCO에 대한 결과를 개선할 수 있습니다.

최근에 제안된 NL(non-local) 모델 [43]을 사용하여,
우리는 40.3 마스크 AP와 45.0 박스 AP를 달성합니다. 이 결과는
테스트 시간 증가 없이 메서드는 3fps에서 실행됩니다.
테스트 시간에 Nvidia Tesla P100 GPU에서.

테스트 시간 증대: 모델 결과를 결합합니다.
다음 단계로 [400, 1200] 픽셀의 스케일을 사용하여 평가
100과 그들의 수평 플립. 이것은 41.8 마스크 AP와 47.3 박스 AP의 단일 모델 결과를 제공합니다.

위의 결과는
COCO 2017 대회(앙상블도 사용했으며,
여기서는 논의하지 않음). 처음으로 우승한 세 팀은
인스턴스 세분화 작업은 모두 보고된 바에 따르면
Mask R-CNN 프레임워크의 확장.

키포인트 감지

우리는 표 9에 키포인트 감지의 향상된 결과를 보고 합니다. 업데이트된 기준으로 학습률이 감소하는 130,000번의 반복으로 학습 일정을 확장합니다.

100k 및 120k 반복에서 10만큼. 이렇게 하면 APkp 가 향상됩니다.
약 1점. ResNet-50을 ResNet-101로 교체하고
ResNeXt-101은 APkp 를 각각 66.1 및 67.3으로 증가시킵니다.

데이터 증류 [35] 라는 최근 방법 으로 우리는
COCO에서 제공하는 추가 120,000개의 레이블이 지정되지 않은 이미지를 활용할 수 있습니다. 간단히 말해서 데이터 증류는 자가 훈련입니다.
레이블이 지정된 데이터에 대해 학습된 모델을 사용하여 레이블이 지정되지 않은 이미지의 주석을 예측하고 차례로 업데이트하는 전략
이러한 새 주석이 있는 모델입니다. Mask R-CNN은 이러한 자기 훈련 전략을 위한 효과적인 프레임워크를 제공합니다. 데이터 증류를 통해 Mask R-CNN APkp 는 다음과 같이 향상됩니다.
1.8은 69.1을 나타냅니다. Mask R-CNN은 데이터에 레이블이 지정되지 않은 경우에도 추가 데이터에서 이점을 얻을 수 있음을 관찰합니다.

에 사용된 것과 동일한 테스트 시간 증대를 사용하여
인스턴스 세분화를 통해 APkp 를 70.4로 추가로 높였습니다.

감사의 말: Ilija를 인정하고 싶습니다 .
코드 릴리스 및 향상된 기능에 대한 라도사보비치
결과 및 엔지니어링 지원을 위한 Caffe2 팀.

참고문헌

[1] M. Andriluka, L. Pishchulin, P. Gehler 및 B. Schiele. 2D
인간 포즈 추정: 새로운 벤치마크 및 최신 기술
분석. CVPR에서, 2014. 8

[2] P. Arbelaez, J. Pont-Tuset, JT Barron, F. Marques 및 J. Malik. 다중 스케일 조합
그룹화. CVPR에서는
2014. 2

[3] A. Arnab 및 PH Torr. 픽셀 단위 인스턴스 분할
동적으로 인스턴스화된 네트워크를 사용합니다. CVPR에서는 2017. 3. 3,
9

[4] M. Bai 및 R. Urtasun. 자세 분할을 위한 깊은 유역 변환. CVPR에서 2017. 3, 9

[5] S. Bell, CL Zitnick, K. Bala 및 R. Girshick. 내부 외부 네트워크: 스킵 풀링으로 컨텍스
트에서 객체 감지
및 순환 신경망. CVPR에서 2016. 5

[6] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh. 부품 친화성 필드를 사용한 실시간 다중인형
2D 포즈 추정. CVPR에서는
2017. 7, 8

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler,
R. Benenson, U. Franke, S. Roth 및 B. Schiele. 그만큼
의미론적 도시 장면 이해를 위한 도시 풍경 데이터 세트.
CVPR에서 2016. 9

[8] J. Dai, K. He, Y. Li, S. Ren 및 J. Sun. 인스턴스 구분
완전 컨볼루션 네트워크. ECCV에서 2016. 2

[9] J. Dai, K. He, J. Sun. 컨볼루션 피쳐 마스킹
공동 개체 및 물건 분할. CVPR에서 2015. 2

[10] J. Dai, K. He, J. Sun. 다중 작업 네트워크 캐스케이드를 통한 인스턴스 인식 시맨틱 분할.
CVPR에서는 2016. 2, 3,
4, 5, 6

[11] J. Dai, Y. Li, K. He, J. Sun. R-FCN: 물체 감지
지역 기반 완전 컨볼루션 네트워크. NIPS에서 2016. 2

[12] R. 거식. 빠른 R-CNN. ICCV에서, 2015. 1, 2, 3, 4, 6

[13] R. Girshick, J. Donahue, T. Darrell 및 J. Malik. 정확한 객체 감지 및 의미 체계를 위
한 풍부한 기능 계층
분할. CVPR에서 2014. 2, 3

[14] R. Girshick, F. Iandola, T. Darrell 및 J. Malik. 변형 가능
부품 모델은 합성곱 신경망입니다. CVPR에서는
2015. 4

[15] B. Hariharan, P. Arbelaez, R. Girshick 및 J. Malik. 동시 감지 및 분할. ECCV에서.
2014. 2

[16] B. Hariharan, P. Arbelaez, R. Girshick 및 J. Malik. 개체 세분화 및 세분화된 현지화를
위한 하이퍼 컬럼. CVPR에서 2015. 2

[17] Z. Hayder, X. He 및 M. Salzmann. 모양 인식 인스턴스
분할. CVPR에서 2017. 9

[18] K. He, X. Zhang, S. Ren, J. Sun. 공간 피라미드 풀링
시각적 인식을 위한 딥 컨볼루션 네트워크에서 에
ECCV. 2014. 1, 2

[19] K. He, X. Zhang, S. Ren, J. Sun. 딥 레지듀얼 러닝
이미지 인식을 위해. CVPR에서 2016. 2, 4, 7, 10

[20] J. Hosang, R. Benenson, P. Dollar 및 B. Schiele. 효과적인 탐지 제안의 근거는 무엇입
니까? 파미, 2015. 2

[21] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara,
A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al.
최신 컨볼루션 객체의 속도/정확도 트레이드오프
탐지기. CVPR에서 2017. 2, 3, 4, 6, 7

[22] M. Jaderberg, K. Simonyan, A. Zisserman 및
K. Kavukcuoglu. 공간 변압기 네트워크.
NIPS, 2015. 4

[23] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy 및
C. 로더. Instancecut: 가장자리에서 다중 절단이 있는 인스턴스까지. CVPR에서
2017. 3, 9

[24] A. Krizhevsky, I. Sutskever 및 G. Hinton. 심층 컨볼루션 신경망을 사용한 ImageNet
분류. NIPS에서는
2012. 2

[25] Y. LeCun, B. Boser, JS Denker, D. Henderson, RE
Howard, W. Hubbard 및 LD Jackel. 역전파
필기 우편번호 인식에 적용됩니다. 신경 계산, 1989. 2

[26] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei. 완전 컨볼루션
인스턴스 인식 의미론적 세분화. CVPR에서 2017. 2,
3, 5, 6

[27] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan 및 S. Belongie. 물체 감지를 위
한 기능 피라미드 네트워크.
CVPR에서 2017. 2, 4, 5, 7

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra manan, P. Dollar 및
CL Zitnick. Microsoft COCO: 컨텍스트의 공통 개체. ECCV에서 2014. 2, 5

[29] S. Liu, J. Jia, S. Fidler 및 R. Urtasun. SGN: 인스턴스 분할을 위한 순차 그룹화 네트워
크. ICCV에서는
2017. 3, 9

[30] J. Long, E. Shelhamer 및 T. Darrell. 완전 컨볼루션
의미론적 세분화를 위한 네트워크. CVPR에서 2015. 1, 3, 6

[31] V. Nair 및 GE Hinton. 수정된 선형 장치는 제한된 볼츠만 기계를 개선합니다. ICML에서
2010. 4

[32] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tomp 아들, C. Bregler 및
K. Murphy. 야생에서 정확한 다중인물 포즈 추정을 향하여. CVPR에서 2017. 8

[33] PO Pinheiro, R. Collobert 및 P. Dollar. 대상 후보를 분할하는 방법을 배웁니다. NIPS
에서 2015. 2, 3

[34] PO Pinheiro, T.-Y. 린, R. 콜로버트, P. 달러. 배우다-
개체 세그먼트를 세분화합니다. ECCV에서는 2016. 2, 3

[35] I. Radosavovic, P. Dollar, R. Girshick, G. Gkioxari 및 K. He. 데이터 증류: omni-
supervised learning을 향하여.
arXiv:1712.04440, 2017. 10

[36] S. Ren, K. He, R. Girshick 및 J. Sun. 더 빠른 R-CNN: 영역 제안 네트워크로 실시간 물
체 감지를 지원합니다. NIPS에서 2015. 1, 2, 3, 4, 7

[37] S. Ren, K. He, R. Girshick 및 J. Sun. 더 빠른 R-CNN: 영역 제안 네트워크로 실시간 물
체 감지를 지원합니다. TPAMI에서는 2017. 10

[38] A. Shrivastava, A. Gupta, R. Girshick. 온라인 하드 예제 마이닝으로 영역 기반 객체 감
지기를 훈련합니다. 에
CVPR, 2016. 2, 5

[39] A. Shrivastava, R. Sukthankar, J. Malik 및 A. Gupta. 건너뛰기 연결을 넘어서: 물체
감지를 위한 하향식 변조. arXiv:1612.06851, 2016. 4, 7

[40] C. Sun, A. Shrivastava, S. Singh 및 A. Gupta. 재방문
딥러닝 시대에 데이터의 부당한 효율성. 에
ICCV, 2017. 10

[41] C. Szegedy, S. Ioffe 및 V. Vanhoucke. Inception-v4, inception-resnet 및 학습에 대한 잔여 연결의 영향. ICLR Workshop, 2016. 7 [42] JR Uijlings, KE van de Sande, T. Gevers 및 AW

스멜더. 객체 인식을 위한 선택적 검색. IJCV, 2013. 2

[43] X. Wang, R. Girshick, A. Gupta, K. He. 비-로컬 신경망. arXiv:1711.07971, 2017. 10

[44] S.-E. Wei, V. Ramakrishna, T. Kanade 및 Y. Sheikh. 컨볼루션 포즈 머신. CVPR, 2016. 8 [45] S. Xie, R. Girshick, P. Dǫllar, Z. Tu 및 K. He. 심층 신경망에 대한 집계된 잔차 변환입니다. CVPR에서 2017. 4. 10 .