Atrous 분리형 인코더-디코더 시맨틱 이미지 분할을 위한 컨볼루션

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff,
하트위그 아담

Google Inc.
{lcchen, yukun, gpapan, fschroff, hadam }@google.com

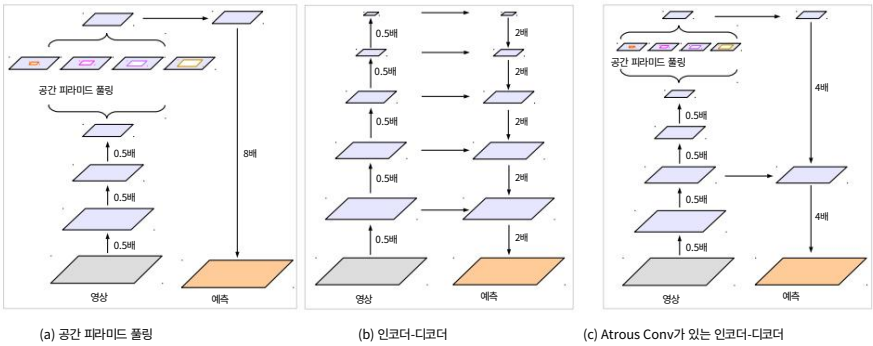
추상적인. 공간 피라미드 풀링 모듈 또는 인코딩-디코더 구조는 의미론적 분할 작업을 위해 심층 신경망에서 사용됩니다. 전자 네트워크는 다중 속도 및 다중 유효 시야에서 필터 또는 풀링 작업으로 들어오는 기능을 조사하여 다중 규모 컨텍스트 정보를 인코딩할 수 있는 반면 후자의 네트워크는 공간 정보를 점진적으로 복구하여 더 선명한 객체 경계를 캡처할 수 있습니다. . 이 작업에서 우리는 두 가지 방법의 장점을 결합할 것을 제안합니다. 특히 제안된 모델인 DeepLabv3+는 특히 객체 경계를 따라 분할 결과를 개선하기 위해 간단하지만 효과적인 디코더 모듈을 추가하여 DeepLabv3을 확장합니다. 우리는 Xception 모델을 더 탐색하고 Atrous Spatial Pyramid Pooling과 디코더 모듈 모두에 깊이별 분리 가능한 컨볼루션을 적용하여 더 빠르고 강력한 인코더-디코더 네트워크를 만듭니다. 우리는 PASCAL VOC 2012 및 Cityscapes 데이터 세트에서 제안된 모델의 효율성을 입증하여 후처리 없이 89.0% 및 82.1%의 테스트 세트 성능을 달성했습니다. 우리의 백서는 https://github.com/tensorflow/models/tree/master/research/deeplab에서 Tensorflow 에서 제안된 모델의 공개적으로 사용 가능한 참조 구현과 함께 제공됩니다 . .

키워드: 시맨틱 이미지 분할, 공간 피라미드 풀링, 인코더 디코더, 깊이별 분리 가능한 컨볼루션.

1. 소개

완전 컨볼루션 신경망[8,11]을 기반으로 하는 이미지 [1개의 진화적 신경망 [6,7,8,9,10]] 의 모든 픽셀에 의미론적 레이블을 할당하는 것을 목표로 하는 최첨단 의미론적 분할 방법에 대한 최근 작업은 이전 시스템에 비해 현저한 개선을 보여줍니다. 벤치마크 작업에서 손 으로 만든 기능 [12,13,14,15,16,17] . 이 작업에서 우리 는 의미론적 분할 을 위해 공간 피라미드 풀링 모듈 [18,19,20] 또는 인코더-디코더 구조 [21,22] 를 사용하는 두 가지 유형의 신경망을 고려합니다 . 후자는 선명한 물체 경계를 얻을 수 있지만 다른 해상도입니다.

여러 규모에서 상황 정보를 캡처하기 위해 DeepLabv3 [23] 는 서로 다른 비율로 여러 병렬 atrous convolution(Atrous라고 함)을 적용합니다 .



(a) 공간 피라미드 풀링 (b) 인코더-디코더 (c) Atrous Conv가 있는 인코더-디코더

그림 1. 공간 피라미드 풀링 모듈(a)을 사용하는 DeepLabv3를 인코더-디코더 구조(b)로 개선합니다. 제안된 모델인 DeepLabv3+에는 인코더 모듈의 풍부한 의미 정보가 포함되어 있으며 세부적인 객체 경계는 간단하지만 효과적인 디코더 모듈에 의해 복구됩니다. 인코더 모듈을 사용하면 atrous convolution을 적용하여 임의의 해상도에서 특징을 추출할 수 있습니다.

Spatial Pyramid Pooling 또는 ASPP), PSPNet [24] 은 서로 다른 그리드 스케일에서 풀링 작업을 수행합니다. 마지막 특징 맵에 풍부한 의미 정보가 인코딩되어 있음에도 불구하고 네트워크 백본 내에서 스트라이딩 작업이 포함된 풀링 또는 컨볼루션으로 인해 객체 경계와 관련된 세부 정보가 누락되었습니다. 이것은 더 조밀한 특징 맵을 추출하기 위해 atrous convolution을 적용함으로써 완화될 수 있습니다. 그러나 최첨단 신경망 [7,9,10,25,26] 의 설계 와 제한된 GPU 메모리를 고려할 때 입력 해상도. 예를 들어 ResNet-101 [25] 을 사용하면 입력 해상도보다 16배 작은 출력 특성을 추출하기 위해 atrous convolution을 적용할 때 마지막 3개의 잔여 블록(9개 레이어) 내의 특성을 확장해야 합니다. 설상가상으로, 입력보다 8배 작은 출력 기능이 필요한 경우 26개의 잔여 블록(78개의 레이어!)이 영향을 받습니다. 따라서 이러한 유형의 모델에 대해 더 조밀한 출력 기능을 추출하는 경우 계산 집약적입니다. 반면에 인코더-디코더 모델 [21,22] 은 인코더 경로에서 더 빠른 계산을 제공하고(확장된 기능이 없기 때문에) 디코더 경로에서 날카로운 객체 경계를 점진적으로 복구합니다. 두 방법의 장점을 결합하기 위해 다중 스케일 컨텍스트 정보를 통합하여 인코더-디코더 네트워크에서 인코더 모듈을 강화할 것을 제안합니다.

특히 우리가 제안한 DeepLabv3+ 모델 은 그림 1과 같이 객체 경계를 복구하기 위해 간단하지만 효과적인 디코더 모듈을 추가하여 DeepLabv3 [23] 을 확장합니다. 풍부한 의미 정보는 DeepLabv3의 출력에 인코딩됩니다. atrous convolution을 사용하면 계산 리소스의 예산에 따라 인코더 기능의 밀도를 제어할 수 있습니다.

또한 디코더 모듈은 상세한 객체 경계 복구를 허용합니다.

최근 deepwise separable convolution [27,28,26,29,30]의 성공에 동기를 부여하여, 우리는 또한 이 작업을 탐색하고 다음 작업 에 대해 [31] 과 유사한 Xception 모델 [26] 을 적용하여 속도와 정확도의 개선을 보여줍니다 .

의미론적 분할, ASPP 및 디코더 모듈 모두에 atrous 분리 가능한 컨볼루션 적용. 마지막으로 PASCAL VOC 2012 및 Cityscapes 데이터스트에 대해 제안된 모델의 효율성을 입증하고 후처리 없이 89.0% 및 82.1%의 테스트 세트 성능을 달성하여 새로운 최첨단을 설정합니다.

요약하면 다음과 같습니다.

– 우리는 DeepLabv3를 강력한 인코더 모듈과 간단하면서도 효과적인 디코더 모듈로 사용하는 새로운 인코더-디코더 구조를 제안합니다.
– 우리의 구조에서는 기존의 인코더-디코더 모델에서는 불가능했던 정밀도와 런타임을 트레이드 오프하기 위해 atrous convolution을 통해 추출된 인코더 특징의 해상도를 임의로 제어할 수 있습니다.

– 분할 작업을 위해 Xception 모델을 적용하고 ASPP 모듈과 디코더 모듈 모두에 깊이별 분리 가능한 컨볼루션을 적용하여 더 빠르고 강력한 인코더-디코더 네트워크를 생성합니다.

– 우리가 제안한 모델은 PASCAL VOC 2012 및 Cityscapes 데이터 세트에서 새로운 최첨단 성능을 얻습니다. 또한 설계 선택 및 모델 변형에 대한 자세한 분석을 제공합니다.

– 제안된 모델의 Tensorflow 기반 구현을 https://github.com/tensorflow/models/tree/master/research/deeplab 에서 공개적으로 사용할 수 있습니다.

2 관련 업무

FCN(Fully Convolutional Networks) [8,11] 에 기반한 모델 은 여러 분할 벤치마크 [1,2,3,4,5]에서 상당한 개선을 보여주었습니다. 다중 스케일 입력(즉, 이미지 피라미드)을 사용하는 모델을 포함하여 분할을 위해 컨텍스트 정보를 활용하기 위해 제안된 여러 모델 변형 [12,13,14,15,16,17,32,33] 이 있습니다. [34,35, 36,37,38,39] 또는 확률적 그래픽 모델을 채택한 모델(예: DenseCRF [40] 과 효율적인 추론 알고리즘 [41])

[42,43,44,37,45,46,47,48,49,50,51,39]. 이 연구에서는 주로 공간 피라미드 풀링과 인코더-디코더 구조를 사용하는 모델에 대해 논의합니다.

공간 피라미드 풀링: PSPNet [24] 또는 DeepLab [39,23] 과 같은 모델 은 여러 그리드 스케일(이미지 레벨 풀링 [52] 포함) 에서 공간 피라미드 풀링 [18,19] 을 수행 하거나 서로 다른 (Atrous Spatial Pyramid Pooling 또는 ASPP라고 함). 이러한 모델은 다중 스케일 정보를 활용하여 여러 세분화 벤치마크에서 유망한 결과를 보여주었습니다.

인코더-디코더: 인코더-디코더 네트워크는 인간의 자세 추정 [53], 물체 감지 [54,55,56], 의미론적 분할 [11,57,21,22, 58,59,60,61,62,63,64].

일반적으로 인코더-디코더 네트워크에는 (1) 기능 맵을 점진적으로 줄이고 더 높은 의미 정보를 캡처하는 인코더 모듈과 (2) 공간 정보를 점진적으로 복구하는 디코더 모듈이 포함됩니다. 이 아이디어를 기반으로 DeepLabv3 [23] 을 인코더 모듈로 사용하고 더 선명한 분할을 얻기 위해 간단하면서도 효과적인 디코더 모듈을 추가할 것을 제안합니다.

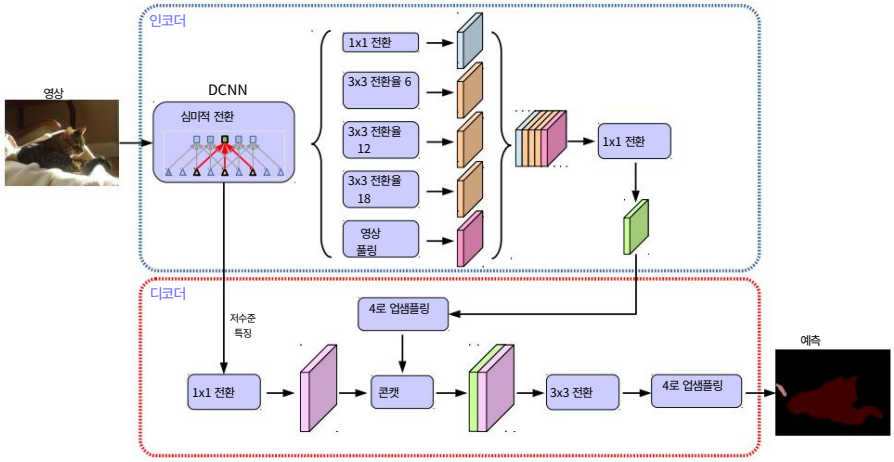


그림 2. 제안한 DeepLabv3+는 인코더 디코더 구조를 사용하여 DeepLabv3을 확장합니다. 인코더 모듈은 다중 스케일에서 atrous convolution을 적용하여 다중 스케일 컨텍스트 정보를 인코딩하는 반면, 간단하지만 효과적인 디코더 모듈은 객체 경계를 따라 분할 결과를 개선합니다.

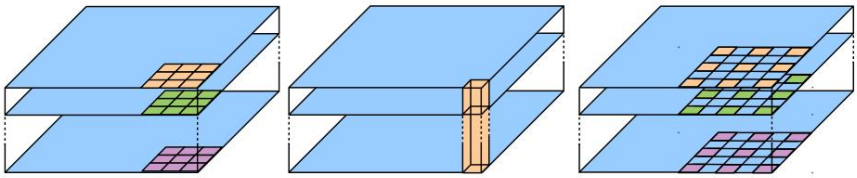
Depthwise separable convolution: Depthwise separable convolution [27,28] 또는 group convolution [7,65], 유사한(또는 약간 더 나은) 성능을 유지하면서 계산 비용과 매개변수 수를 줄이는 강력한 연산입니다. 이 연산은 최근 많은 신경망 설계에서 채택되었습니다 [66,67,26,29,30,31,68]. 특히 COCO 2017 탐지 도전 제출에 대한 [31] 과 유사한 Xception 모델 [26] 을 탐색 하고 의미론적 세분화 작업의 정확도와 속도 모두에서 개선을 보여줍니다.

3가지 방법

이 섹션에서는 atrous convolution [69,70,8,71,42] 과 depth wise separable convolution [27,28,67,26,29]을 간략하게 소개합니다. 그런 다음 인코더 출력에 추가되는 제안된 디코더 모듈을 논의하기 전에 인코더 모듈로 사용되는 DeepLabv3 [23] 을 검토합니다. 우리는 또한 더 빠른 계산으로 성능을 더욱 향상시키는 수정된 Xception 모델 [26,31] 을 제시합니다.

3.1 Atrous Convolution이 있는 인코더-디코더

Atrous convolution: Atrous convolution은 deep convolutional neural network에 의해 계산된 기능의 해상도를 명시적으로 제어하고 다중 스케일 정보를 캡처하기 위해 filter의 field of view를 조정할 수 있게 해주는 강력한 도구이며 표준 convolution 연산을 일반화합니다. 2차원 신호의 경우 출력 특성 맵 y의 각 위치 i와 컨볼루션 필터 w에 대해 다음과 같이 입력 특성 맵 x에 대해 atrous 컨볼루션이 적용됩니다.



(a) 깊이별 전환수 (b) 포인트별 전환 (c) Atrous 깊이별 conv.

그림 3. 3×3 Depthwise separable convolution은 표준 convolution을 다음과 같이 분해합니다.
(a) 깊이별 컨볼루션(각 입력 채널에 대해 단일 필터 적용) 및 (b) a
포인트별 컨볼루션(채널 전체에 걸쳐 깊이별 컨볼루션의 출력 결합). 이 작업에서 우리는 atrous convolution이 있는
atrous separable convolution을 탐구합니다.
속도 = 2인 (c)에서와 같이 깊이별 컨볼루션에서 채택됩니다.

$$y[i] = \sum_{k} x[i + r \cdot k]w[k] \tag{1}$$

여기서 atrous rate r은 입력을 샘플링하는 보폭을 결정합니다.
신호. 자세한 내용 은 관심 있는 독자를 [39] 참조하십시오. 참고로 표준
convolution은 rate r = 1인 특별한 경우입니다. 필터의 field-of-view는 다음과 같습니다.
비율 값을 변경하여 적응적으로 수정합니다.

Depthwise separable convolution: 표준 convolution을 depthwise convolution과 point wise convolution(즉, 1 × 1 convolution)으로 변환하는 Depthwise separable convolution은 계산 복잡성을 크게 줄입니다. 구체적으로, 깊이별 컨볼루션은 공간 컨볼루션을 수행합니다.
각 입력 채널에 대해 독립적으로, 점별 컨볼루션은 깊이별 컨볼루션의 출력을 결합하는 데 사용됩니다. 텐서플로우에서
[72] 깊이별 분리 가능한 컨볼루션의 구현, 아트러스 컨볼루션은
다음과 같이 깊이별 컨볼루션(즉, 공간 컨볼루션)에서 지원되었습니다.
그림 3에 나와 있습니다. 이 작업에서는 결과 컨볼루션을 atrous
separable convolution, 그리고 atrous separable convolution이
유사한(또는 더 나은) 성능을 유지하면서 제안된 모델의 계산 복잡성을 줄입니다.

인코더로서의 DeepLabv3: DeepLabv3 [23] 은 atrous convolution [69,70,8,71]을 사용합니다.
심층 컨볼루션 신경망에 의해 계산된 특징을 추출하기 위해
임의의 해상도. 여기에서 출력 스트라이드를 입력 이미지의 비율로 나타냅니다.
공간 해상도를 최종 출력 해상도로 변경합니다(글로벌 풀링 또는 완전 연결 계층 이전). 이미지 분류 작업의 경우 공간 해상도
최종 기능 맵은 일반적으로 입력 이미지 해상도보다 32배 작습니다.
따라서 출력 스트라이드 = 32. 의미론적 분할 작업을 위해 다음을 채택할 수 있습니다.
출력 stride = 16(또는 8), striding을 제거하여 더 조밀한 특징 추출
마지막 하나(또는 두 개) 블록에서 그리고 atrous convolution을 적용하여 상응하게 대응합니다(예: 마지막 두 블록에 각각 rate = 2 및 rate = 4를 적용합니다.
출력 스트라이드 = 8). 또한 DeepLabv3는 Atrous Spatial
여러 규모에서 컨벌루션 기능을 조사하는 피라미드 풀링 모듈
이미지 수준 기능을 사용하여 서로 다른 비율로 atrous convolution을 적용하여

트 [52]. 우리는 원본 DeepLabv3에서 logits 전에 마지막 기능 맵을 사용합니다.
제안된 인코더-디코더 구조에서 인코더 출력으로. 인코더 출력 기능 맵에는 256개의 채널과 풍부한 의미 정보가 포함되어 있습니다.
또한 다음을 적용하여 임의의 해상도에서 특징을 추출할 수 있습니다.
계산 예산에 따라 atrous convolution.

제안된 디코더: DeepLabv3의 인코더 기능은 일반적으로 출력 스트라이드 = 16으로 계산됩니다.
[23] 의 작업 에서 기능은 쌍선형입니다.
순진한 디코더 모듈로 간주될 수 있는 16배만큼 업샘플링됩니다.
그러나 이 순진한 디코더 모듈은 객체 분할 세부 정보를 성공적으로 복구하지 못할 수 있습니다. 따라서 우리는 다음과 같이 간단하면서도 효과적인 디코더 모듈을 제안합니다.
인코더 기능은 먼저 다음으로 이중 선형 업샘플링됩니다.
4의 인수를 적용한 다음 해당 하위 수준 기능과 연결 [73]
동일한 공간 해상도(예: Conv2
ResNet-101 [25]에서 스트라이딩하기 전에). 우리는 또 다른 1 × 1 convolution을 적용합니다.
해당 로우 레벨 기능에는 일반적으로 많은 수의 채널(예: 256 또는 512)이 포함되어 있으므로 채널 수를 줄이기 위한 로우 레벨 기능
풍부한 인코더 기능의 중요성을 능가할 수 있습니다.
우리 모델) 훈련을 더 어렵게 만듭니다. 연결 후 적용
특징을 다듬기 위한 몇 개의 3 × 3 컨볼루션 다음에 또 다른 단순한 쌍선형
4의 인수로 업샘플링합니다. 우리는 Sec에서 보여줍니다. 4 출력 스트라이드 사용 = 16
인코더 모듈의 경우 속도와 정확도 간에 최상의 균형을 유지합니다.
출력 스트라이드 = 8을 사용할 때 성능이 약간 향상됩니다.
추가적인 계산 복잡성을 대가로 인코더 모듈.

3.2 수정된 정렬 예외

Xception 모델 [26] 은 ImageNet [74] 에서 빠른 계산으로 유망한 이미지 분류 결과를 보여주었습니다. 보다 최근에 MSRA 팀 [31] 은 다음을 수정합니다 .
Xception 모델(Aligned Xception이라고 함)은 객체 감지 작업의 성능을 더욱 향상시킵니다. 이러한 발견에 동기를 부여하여 우리는
시맨틱 이미지 작업을 위해 Xception 모델을 적용하는 동일한 방향
분할. 특히 MSRA에 추가로 몇 가지 변경 사항을 적용합니다.
수정, 즉 (1) 우리가 하는 것을 제외하고 는 [31] 에서와 동일한 더 깊은 Xception
빠른 계산 및 메모리를 위해 진입 흐름 네트워크 구조를 수정하지 않음
효율성, (2) 모든 최대 풀링 작업은 깊이별 분리 가능으로 대체됩니다.
임의의 해상도에서 특징 맵을 추출하기 위해 atrous 분리 가능한 컨볼루션을 적용할 수 있는 striding이 있는 컨볼루션(또 다른 옵션은
atrous 알고리즘을 최대 풀링 작업으로 확장) 및 (3) 추가 배치
정규화 [75] 및 ReLU 활성화가 각각의 3 × 3 이후에 추가됩니다 .
Convolution, MobileNet 디자인과 유사합니다 [29]. 자세한 내용은 그림 4 를 참조하십시오.

4 실험적 평가

ImageNet-1k [74] 사전 훈련된 ResNet-101 [25] 또는 수정된 정렬 을 사용합니다.
Xception [26,31] 은 atrous convolution으로 조밀한 특징 맵을 추출합니다. 우리의 구현은 TensorFlow [72] 를 기반으로 하며 공개적으로 제공됩니다.

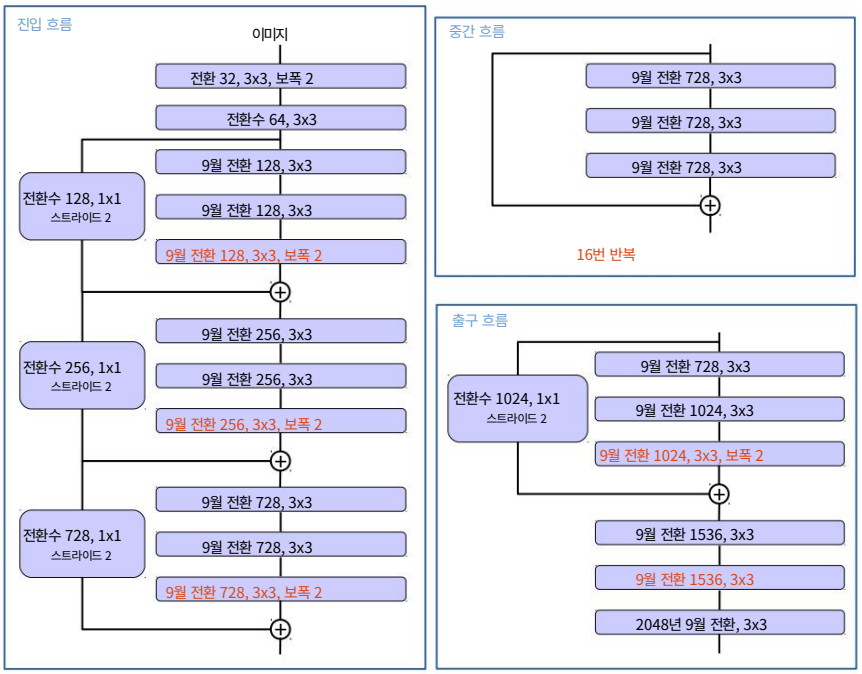


그림 4. Xception을 다음과 같이 수정합니다. (1) 더 많은 레이어(진입 흐름의 변경을 제외하고 MSRA 수정과 동일), (2) 모든 최대 풀링 작업이 스트라이딩이 있는 깊이별 분리 가능한 컨볼루션으로 대체됩니다.) MobileNet과 유사하게 각 3 × 3 깊이 방향 컨볼루션 후에 추가 배치 정규화 및 ReLU가 추가됩니다.

제안된 모델은 20개의 전경 객체 클래스와 하나의 배경 클래스를 포함 하는 PASCAL VOC 2012 시맨틱 분할 벤치마크 [1] 에서 평가됩니다. 원본 데이터 세트에는 1, 464(기차), 1, 449(발) 및 1, 456(테스트) 픽셀 수준 주석 이미지가 포함되어 있습니다. [76]에서 제공한 추가 주석으로 데이터 세트를 보강하여 10,582(trainaug) 훈련 이미지를 생성합니다.

성능은 21개 클래스(mIOU)에 걸쳐 평균화된 픽셀 교집합의 관점에서 측정됩니다.

우리는 [23] 에서와 동일한 교육 프로토콜을 따르고 자세한 내용은 관심 있는 독자를 [23] 으로 참조하십시오. 간단히 말해서, 우리는 동일한 학습률 일정(즉, "폴리" 정책 [52] 및 동일한 초기 학습률 0.007), 자르기 크기 513 × 513, 출력 스트라이드 = 16일 때 배치 정규화 매개변수 미세 조정 [75] 을 사용하고, 훈련 중 랜덤 스케일 데이터 증대. 제안된 디코더 모듈에는 배치 정규화 매개변수도 포함되어 있습니다. 제안된 모델은 각 구성 요소에 대한 부분적 사전 학습 없이 종단 간 학습됩니다.

4.1 디코더 설계 선택

“DeepLabv3 기능 맵”을 DeepLabv3(즉, ASPP 기능 및 이미지 수준 기능을 포함하는 기능)에서 계산한 마지막 기능 맵으로 정의하고 [k × k, f]를 커널 k × k 및 f를 사용한 컨볼루션 연산으로 정의합니다. 필터.

출력 보폭 = 16을 사용할 때 ResNet-101 기반 DeepLabv3 [23] bi는 훈련과 평가 모두에서 로짓을 16만큼 선형으로 업샘플링합니다. 이 간단한 쌍선형 업샘플링은 순진한 디코더 설계로 간주될 수 있으며 PASCAL VOC 2012 값 세트에서 77.21% [23] 의 성능을 달성 하고 훈련 중 이 순진한 디코더를 사용하지 않는 것보다 1.2% 더 좋습니다 (즉, 훈련 중 groundtruth 다운샘플링) . 이 순진한 기준선을 개선하기 위해 우리가 제안한 모델 "DeepLabv3+"는 그림 2와 같이 인코더 출력 위에 디코더 모듈을 추가합니다. 디코더 모듈에서 우리는 다른 디자인 선택을 위한 세 곳, 즉 (1)을 고려합니다. 인코더 모듈에서 저수준 기능 맵의 채널을 줄이는 데 사용되는 1x1 회선, (2) 더 선명한 분할 결과를 얻는 데 사용되는 3 × 3 회선, (3) 사용되어야 하는 인코더 저수준 기능 .

디코더 모듈에서 1 × 1 컨볼루션의 효과를 평가하기 위해 [3 × 3, 256] ResNet-101 네트워크 백본의 Conv2 기능, 즉 res2x 잔차 블록의 마지막 기능 맵(구체적으로, 우리는 스트라이딩 전에 기능 맵을 사용합니다). 탭에 표시된 대로. 1, 인코더 모듈에서 저수준 기능 맵의 채널을 48 또는 32로 줄이면 성능이 향상됩니다. 따라서 채널 축소를 위해 [1 × 1, 48]을 채택합니다.

그런 다음 디코더 모듈에 대한 3 × 3 회선 구조를 설계하고 결과를 Tab에 보고합니다. 2. 우리는 Conv2 기능 맵(스트라이딩 전)을 DeepLabv3 기능 맵과 연결한 후 단순히 하나 또는 세 개의 회선을 사용하는 것보다 256개 필터가 있는 두 개의 3x3 회선을 사용하는 것이 더 효과적이라는 것을 발견했습니다.

필터 수를 256에서 128로 변경하거나 커널 크기를 3 × 3에서 1 × 1로 변경하면 성능이 저하됩니다. 또한 디코더 모듈에서 Conv2 및 Conv3 기능 맵이 모두 활용되는 경우를 실험합니다. 이 경우 디코더 특성 맵은 2씩 점진적으로 업샘플링되고 Conv3과 Conv2가 차례로 연결되고 각각 [3 × 3, 256] 연산으로 정제됩니다. 전체 디코딩 절차는 U-Net/SegNet 설계와 유사합니다 [21,22].

그러나 우리는 유의미한 개선을 관찰하지 못했습니다. 따라서 결국 우리는 매우 간단하면서도 효과적인 디코더 모듈을 채택합니다. DeepLabv3 기능 맵과 채널 축소된 Conv2 기능 맵의 연결은 두 개의 [3 × 3, 256] 작업으로 정제됩니다. 제안된 DeepLabv3+ 모델은 출력 스트라이드 = 4입니다. 제한된 GPU 리소스를 감안할 때 더 조밀한 출력 기능 맵(즉, 출력 스트라이드 < 4)을 추구하지 않습니다.

4.2 네트워크 백본으로서의 ResNet-101

정확도와 속도 측면에서 모델 변형을 비교하기 위해 mIOU 및 Multiply-Adds in Tab을 보고합니다. 3 제안된 DeepLabv3+ 모델에서 네트워크 백본으로 ResNet-101 [25] 을 사용할 때. atrous convolution 덕분에 우리는

채널 8	16	32	48	64
IOU 77.61%	77.92%	78.16%	78.21%	77.94%

표 1. PASCAL VOC 2012 값 세트. 디코더 1 × 1 컨볼루션의 효과
인코더 모듈에서 저수준 기능 맵의 채널을 줄입니다. 우리는 고친다
디코더 구조의 다른 구성 요소는 [3 × 3, 256] 및 Conv2를 사용합니다.

특징 Conv2	3 × 3 전환 Conv3	구조	야옹
X		[3 × 3, 256]	78.21%
X		[3 × 3, 256] × 2	78.85%
X		[3 × 3, 256] × 3	78.02%
X		[3 × 3, 128]	77.25%
X		[1 × 1, 256]	78.07%
X	X	[3 × 3, 256]	78.61%

표 2. 인코더를 줄이기 위해 [1 × 1, 48] 고정 시 디코더 구조의 영향
기능 채널. Conv2(스트라이드 전)를 사용하는 것이 가장 효과적이라는 것을 알았습니다.
기능 맵 및 두 개의 추가 [3 × 3, 256] 작업. VOC 2012 val set에서의 성능.

인코더	디코더	MS	Flip	mIOU	Multiply-Adds	train OS	eval OS
16	16			77.21%	81.02B		
16	8			78.51%	276.18B		
16	8		X	79.45%	2435.37B		
16	8		XX	79.77%	4870.59B		
16	16	엑스		78.85%	101.28B		
16	16	X	X	80.09%	898.69B		
16	16	X	XX	80.22%	1797.23B		
16	8	X		79.35%	297.92B		
16	8	X	X	80.43%	2623.61B		
16	8	X	XX	80.57%	5247.07B		
32	32			75.43%	52.43B		
32	32	엑스		77.37%	742억		
32	16	엑스		77.80%	101.28B		
32	8	엑스		77.92%	297.92B		

표 3. ResNet-101을 사용하여 설정한 PASCAL VOC 2012 값에 대한 추론 전략.
train OS: 훈련 중에 사용된 출력 스트라이드입니다. 평가 OS: 사용된 출력 스트라이드
평가 중. 디코더: 제안된 디코더 구조를 사용합니다. MS: 평가 중 다중 스케일 입력. Flip: 왼쪽에서 오른쪽으로 뒤집힌 입력을 추가합니다.

교육 및 평가 중에 다양한 해상도에서 기능을 얻을 수 있습니다.
단일 모델을 사용합니다.

모델	상위 1개 오류	상위 5개 오류
ResNet-101 재현	22.40%	20.19%
수정된 예외		6.02% 5.17%

표 4. ImageNet-1K 검증 세트의 단일 모델 오류율.

기준선: 탭의 첫 번째 행 블록입니다. 3 은 평가(즉, 평가 출력 스트라이드 = 8) 동안 더 조밀한 특징 맵을 추출하고 다중 스케일 입력을 채택하면 성능이 향상된다는 것을 보여주는 [23] 의 결과를 포함합니다 . 게다가 왼쪽 오른쪽으로 뒤집힌 입력을 추가하면 미미한 성능 향상만으로 계산 복잡성이 두 배가 됩니다.

디코더 추가: Tab의 두 번째 행 블록. 3 은 제안된 디코더 구조를 적용했을 때의 결과이다. eval output stride = 16 또는 8을 사용할 때 성능은 각각 77.21%에서 78.85%로 또는 78.51%에서 79.35%로 향상되지만 약 20B의 추가 계산 오버헤드가 발생합니다. 멀티 스케일 및 좌우 반전 입력을 사용할 때 성능이 더욱 향상됩니다.

더 거친 기능 맵: 빠른 계산을 위해 train output stride = 32(즉, 훈련 중에 atrous convolution 이 전혀 없음)를 사용하는 경우도 실험합니다. Tab의 세 번째 행 블록에 표시된 대로. 3, 디코더를 추가하면 74.20B Multiply-Adds만 필요하지만 약 2% 개선됩니다.

그러나 성능은 우리가 train output stride = 16과 다른 eval output stride 값을 사용하는 경우보다 항상 약 1%에서 1.5% 낮습니다. 따라서 복잡도 예산에 따라 교육 또는 평가 중에 output stride = 16 또는 8을 사용하는 것을 선호합니다.

4.3 네트워크 백본으로서의 예외

우리는 더 강력한 Xception [26] 을 네트워크 백본으로 사용합니다. [31] 에 이어 Sec .에 설명된 대로 몇 가지 더 변경합니다. 3.2.

ImageNet 사전 훈련: 제안된 Xception 네트워크는 [26] 의 유사한 훈련 프로토콜을 사용 하여 ImageNet-1k 데이터 세트 [74] 에서 사전 훈련됩니다. 구체적으로, 우리는 모멘텀 = 0.9, 초기 학습률 = 0.05, 비율 감소 = 0.94 매 2 epoch, 가중치 감소 4e − 5를 갖는 Nesterov 모멘텀 옵티마이저를 채택합니다. 우리는 50개의 GPU로 비동기식 훈련을 사용하고 각 GPU는 이미지 크기가 있는 배치 크기 32를 가집니다. 299×299. 목표는 의미론적 분할을 위해 ImageNet에서 모델을 사전 훈련하는 것이기 때문에 하이퍼 매개변수를 매우 세게 조정하지 않았습니다. 탭의 검증 세트에 대한 단일 모델 오류율을 보고합니다. 4 는 베이스라인과 함께 동일한 훈련 프로토콜에서 ResNet-101 [25] 을 재현했습니다. 수정된 Xception에서 각 3 × 3 깊이 방향 컨볼루션 후에 추가 배치 정규화 및 ReLU를 추가하지 않을 때 Top1 및 Top5 정확도에 대해 0.75% 및 0.29% 성능 저하가 관찰되었습니다.

제안된 Xception을 의미론적 네트워크 백본으로 사용한 결과 세분화는 탭에 보고됩니다. 5.

기준선: 먼저 Tab의 첫 번째 행 블록에서 제안된 디코더를 사용하지 않고 결과를 보고합니다. 5는 Xception을 네트워크로 사용하는 것을 보여줍니다.

백본은 ResNet-101을 사용하는 경우보다 train output stride = eval output stride = 16일 때 성능을 약 2% 향상시킵니다. eval output stride = 8, 추론 중 다중 스케일 입력을 사용하고 왼쪽에서 오른쪽으로 뒤집힌 입력을 추가하여 추가 개선을 얻을 수도 있습니다. 성능이 향상되지 않는 다중 그리드 방법 [77,78,23]을 사용하지 않습니다.

디코더 추가: Tab의 두 번째 행 블록에 표시된 대로. 5에서, 디코더를 추가하면 모든 다른 추론 전략에 대해 eval output stride = 16을 사용할 때 약 0.8% 개선을 가져옵니다. eval output stride = 8을 사용할 때 개선 사항이 줄어듭니다.

깊이별 분리 가능한 컨볼루션 사용: 깊이별 분리 가능한 컨볼루션의 효율적인 계산에 동기를 부여하여 ASPP 및 디코더 모듈에서 이를 추가로 채택합니다. Tab의 세 번째 행 블록에 표시된 대로. 도 5에 도시된 바와 같이, 곱셈-덧셈의 측면에서 계산 복잡성은 33%에서 41%로 크게 감소되는 반면 유사한 mIOU 성능이 얻어집니다.

COCO에 대한 사전 훈련: 다른 최신 모델과의 비교를 위해 MS-COCO 데이터 세트 [79]에 대해 제안된 DeepLabv3+ 모델을 추가로 사전 훈련 하여 모든 다양한 추론 전략에 대해 약 2%의 추가 개선을 산출합니다.

JFT에 대한 사전 학습: [23]과 유사하게 ImageNet-1k [74] 및 JFT-300M 데이터 세트 [80,26,81] 모두에 대해 사전 학습된 제안된 Xception 모델을 사용하여 추가로 0.8~1%를 가져옵니다 . 개선.

테스트 세트 결과: 벤치마크 평가에서 계산 복잡성이 고려되지 않기 때문에 최상의 성능 모델을 선택하고 출력 스트라이드 = 8 및 고정 배치 정규화 매개변수를 사용하여 훈련합니다. 결국 우리의 'DeepLabv3+'는 JFT 데이터 세트 사전 훈련 없이 87.8% 및 89.0%의 성능을 달성합니다.

정성적 결과: 우리는 그림 6에서 최상의 모델의 시각적 결과를 제공합니다.
그림에서 볼 수 있듯이 우리 모델은 후처리 없이 객체를 아주 잘 분할할 수 있습니다.

실패 모드: 그림 6의 마지막 행에서 볼 수 있듯이 우리 모델은 (a) 소파 대 의자, (b) 심하게 가려진 물체, (c) 보기 드문 물체를 분할하는 데 어려움이 있습니다.

4.4 객체 경계에 따른 개선

이 하위 섹션에서는 객체 경계 근처에서 제안된 디코더 모듈의 정확도를 정량화하기 위해 트리맵 실험 [14,40,39] 으로 분할 정확도를 평가합니다 . 특히, 우리는 일반적으로 객체 경계 주변에서 발생하는 val 세트의 'void' 레이블 주석에 형태학적 팽창을 적용합니다. 그런 다음 'void' 레이블의 확장된 대역(trimap이라고 함) 내에 있는 픽셀에 대한 평균 IOU를 계산합니다. 그림 5 (a) 에서 보는 바와 같이 ResNet-101 [25] 과 Xception [26] 네트워크 백본 모두에 대해 제안된 디코더를 사용하면 순진한 쌍선형 업샘플링에 비해 성능이 향상됩니다. 확장된 밴드가 좁을 때 개선이 더 중요합니다. ResNet-101 및 Xception에 대해 각각 4.8% 및 5.4% mIOU 개선을 관찰했습니다.

인코더	디코더	MS	Flip	SC	COCO	JFT	mIOU	Multiply-Adds	train OS	eval OS
16	16							79.17%	680억	
16	16		엑스					80.57%	601.74B	
16	16		더블 엑스					80.79%	1203.34B	
16	8							79.64%	240.85B	
16	8		엑스					81.15%	2149.91B	
16	8		더블 엑스					81.34%	4299.68B	
16	16	엑스						79.93%	89.76B	
16	16	더블 엑스						81.38%	790.12B	
16	16	트리플 엑스						81.44%	1580.10B	
16	8	엑스						80.22%	262.59B	
16	8	더블 엑스						81.60%	2338.15B	
16	8	트리플 엑스						81.63%	4676.16B	
16	16	엑스		엑스				79.79%	54.17B	
16	16	XXX						81.21%	928.81B	
16	8	엑스		엑스				80.02%	177.10B	
16	8	XXX						81.39%	3055.35B	
16	16	엑스			더블 엑스			82.20%	54.17B	
16	16	XXXXX						83.34%	928.81B	
16	8	엑스			더블 엑스			82.45%	177.10B	
16	8	XXXXX						83.58%	3055.35B	
16	16				XXX		83.03%	X	54.17B	
16	16	XXXXXX					84.22%	928.81B		
16	8				XXX		83.39%	177.10B		
16	8	XXXXXX					84.56%	3055.35B		

표 5. 수정된 Xception을 사용할 때 설정한 PASCAL VOC 2012 값에 대한 추론 전략. train OS: 훈련 중에 사용된 출력 스트라이드입니다. 평가 OS:
평가 중에 사용된 출력 스트라이드입니다. 디코더: 제안된 디코더 구조를 사용합니다. MS: 평가 중 다중 스케일 입력. Flip: 왼쪽에서 오른쪽으로 뒤집힌 입력을 추가합니다.
SC: ASPP 및 디코더 모듈 모두에 대해 깊이별 분리 가능한 컨볼루션 채택.
COCO: MS-COCO에서 사전 훈련된 모델입니다. JFT: JFT에서 사전 훈련된 모델입니다.

그림과 같이 가장 작은 트리맵 너비. 우리는 또한 의 효과를 시각화합니다.
그림 5 (b) 에서 제안한 디코더를 사용한다 .

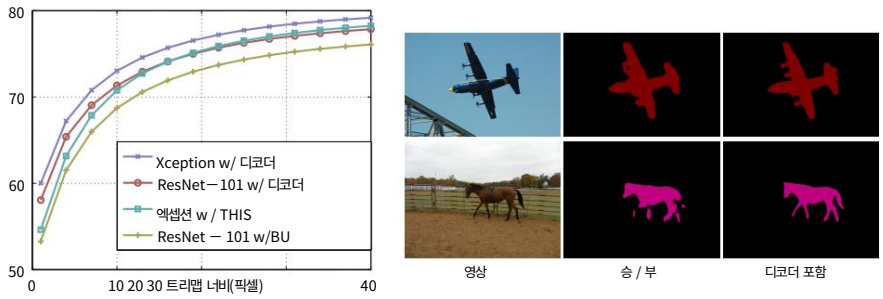
4.5 도시경관에 대한 실험결과

이 섹션에서는 Cityscapes 데이터 세트 [3]에서 DeepLabv3+를 실험합니다.
5000개 이미지의 고품질 픽셀 수준 주석이 포함된 대규모 데이터 세트
(훈련, 검증 및 테스트 세트에 대해 각각 2975, 500 및 1525) 및
대략 20000개의 거친 주석이 달린 이미지.

탭에 표시된 대로. 7 (a), 제안된 Xception 모델을 네트워크로 사용
ASPP 를 포함하는 DeepLabv3 [23] 상단의 백본(X-65로 표시)

방법	야옹
심층 캐스케이드(LC) [82]	82.7
투심플 [77]	83.1
큰 커널 문제 [60] –	83.6
다중경로-RefineNet [58]	84.2
ResNet-38 MS 코코-[83]	84.9
PSP넷 [24]	85.4
IDW-CNN [84]	86.3
CASIA IVA-SDN [63]	86.6
DIS [85]	86.8
DeepLabv3 [23]	85.7
DeepLabv3-JFT [23]	86.9
DeepLabv3+(익셉션)	87.8
DeepLabv3+ (Xception-JFT)	89.0

표 6. 최고 성능 모델의 PASCAL VOC 2012 테스트 세트 결과.



(a) mIOU 대 Trimap 너비 (b) 디코더 효과

그림 5. (a) 객체 경계 주변의 트라이맵 대역폭에 따른 mIOU
기차 출력 스트라이드를 사용할 때 = 평가 출력 스트라이드 = 16. BU: 쌍선형 업샘플링. (b) 제안된 디코더 모듈을 적용했을 때와 비교했을 때의 질적 효과
순진한 쌍선형 업샘플링(BU로 표시). 예제에서는 Xception을 채택합니다.
특징 추출기 및 훈련 출력 스트라이드 = 평가 출력 스트라이드 = 16.

모듈 및 이미지 수준 기능 [52] 은 77.33%의 성능을 달성합니다.
검증 세트. 제안하는 디코더 모듈을 추가하면
78.79%(1.46% 개선)로 성능이 향상되었습니다. 강화된 이미지 수준 기능을 제거하면 성능이 79.14%로 향상되어
DeepLab 모델에서 이미지 수준 기능은 PASCAL에서 더 효과적입니다.
VOC 2012 데이터 세트. 우리는 또한 Cityscapes 데이터셋에서 Xception [26] 의 진입 흐름에서 더 많은 레이어를 늘리는 것이 효과적이라는 것을 발견 했습니다 .
[31] 은 물체 감지 작업을 위해 무엇 을 했습니까? 결과 모델 빌딩
더 깊은 네트워크 백본(표에서 X-71로 표시)의
검증 세트에서 79.55%의 성능을 보였습니다.

val 집합에서 최상의 모델 변형을 찾은 후 추가로 미세 조정합니다.
다른 최신 기술과 경쟁하기 위해 거친 주석에 대한 모델

그림 6. val set에 대한 시각화 결과. 마지막 행은 실패 모드를 보여줍니다.

백본	디코더	ASPP	이미지 수준	mIOU
X-65		엑스	엑스	77.33
X-65	엑스	엑스	엑스	78.79
X-65	엑스	엑스		79.14
X-71	엑스	엑스		79.55

(a) 값 세트 결과

방법	거친	MIOU
ResNet-38 [83]	X	80.6
PSPnet [24]	엑스	81.2
메이필러리 [86]	엑스	82.0
DeepLabv3	엑스	81.3
DeepLabv3+	엑스	82.1

(b) 테스트 세트 결과

표 7. (a) Train fine 세트로 훈련할 때 Cityscapes val 세트에 대한 DeepLabv3+.
(b) Cityscapes 테스트 세트의 DeepLabv3+. Coarse: 기차 추가 세트(거친 주석)도 사용합니다. 이 표에는 몇 가지 상위 모델만 나열되어 있습니다.

모델. 탭에 표시된 대로. 7 (b)에서 제안한 DeepLabv3+는 성능을 달성합니다.
테스트 세트에서 82.1%의 비율로 Cityscapes에서 새로운 최첨단 성능을 설정했습니다.

5. 결론

우리가 제안한 모델 "DeepLabv3+"는 인코더-디코더 구조를 사용합니다.
DeepLabv3는 풍부한 컨텍스트 정보를 인코딩하는 데 사용되며 단순하지만
효과적인 디코더 모듈은 객체 경계를 복구하기 위해 채택됩니다. 하나는 할 수 있었다
또한 임의의 위치에서 인코더 기능을 추출하기 위해 atrous convolution을 적용합니다.
사용 가능한 계산 리소스에 따라 해상도. 우리는 또한 탐구
Xception 모델과 atrous separable convolution을 사용하여 제안된
더 빠르고 강력하게 모델링합니다. 마지막으로, 우리의 실험 결과는 제안된 모델이 PASCAL VOC 2012 및

도시 풍경 데이터 세트.

감사의 말 귀중한 토론에 감사드립니다.
Aligned Xception에 대해 Haozhi Qi 및 Jifeng Dai와 함께, Chen의 피드백
Sun과 Google Mobile Vision 팀의 지원

참고문헌

1. Everingham, M., Eslami, SMA, Gool, LV, Williams, CKI, Winn, J., Zisser man, A.: 파스칼 시각적 개체 클래스는 회고에 도전합니다. IJCV (2014)
2. Mottaghi, R., Chen, X., Liu, X., Cho, NG, Lee, SW, Fidler, S., Urtasun, R., Yuille, A.: 객체 감지 및 의미론적 분할을 위한 컨텍스트의 역할 야생에서. 에서: CVPR. (2014)
3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: 도시 경관 데이터 세트 의미론적 도시 장면 이해를 위한 에서: CVPR. (2016)
4. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: 장면 구문 분석 ade20k 데이터셋을 통해 에서: CVPR. (2017)
5. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: 컨텍스트의 사물 및 사물 클래스. 에서: CVPR. (2018)
6. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning 적용 문서 인식. In: Proc. IEEE. (1998)
7. Krizhevsky, A., Sutskever, I., Hinton, GE: 심도 있는 Imagenet 분류 컨볼루션 신경망. 에서: NIPS. (2012)
8. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: 합성곱 네트워크를 사용한 통합 인식, 지역화 및 탐지. 에서: ICLR. (2014)
9. Simonyan, K., Zisserman, A.: 대규모 이미지 인식을 위한 매우 깊은 컨볼루션 네트워크. 에서: ICLR. (2015)
10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: 더 깊이 회선. 에서: CVPR. (2015)
11. Long, J., Shelhamer, E., Darrell, T.: 의미론적 분할을 위한 완전 컨볼루션 네트워크. 에서: CVPR. (2015)
12. He, X., Zemel, RS, Carreira-Perpindn, M.: 이미지 라벨링을 위한 다중 스케일 조건부 무작위 필드. 에서: CVPR. (2004)
13. Shotton, J., Winn, J., Rother, C., Criminisi, A.: 이미지 이해를 위한 Textonboost: 텍스처, 레이아웃 및 컨텍스트를 공동으로 모델링하여 다중 클래스 개체 인식 및 분할. IJCV (2009)
14. Kohli, P., Torr, PH, et al.: 라벨 시행을 위한 강력한 고차 잠재력 일관성. IJCV 82(3) (2009) 302–324 15. Ladicky, L., Russell, C., Kohli, P., Torr, PH: 객체 클래스 이미지 분할을 위한 연관 계층적 crfs. 에서: ICCV. (2009)
16. Gould, S., Fulton, R., Koller, D.: 장면을 기하학적 의미와 의미로 분해적으로 일관된 영역. 에서: ICCV. (2009)
17. Yao, J., Fidler, S., Urtasun, R.: 전체 장면 설명: 공동 개체 감지, 장면 분류 및 의미론적 분할. 에서: CVPR. (2012)
18. Grauman, K., Darrell, T.: 피라미드 매치 커널: 판별 분류 이미지 기능 세트와 함께. 에서: ICCV. (2005)
19. Lazebnik, S., Schmid, C., Ponce, J.: 기능의 가방 너머: 자연 장면 범주를 인식하기 위한 공간 피라미드 매칭. 에서: CVPR. (2006)
20. He, K., Zhang, X., Ren, S., Sun, J.: 시각적 인식을 위한 심층 컨볼루션 네트워크의 공간 피라미드 풀링. 에서: ECCV. (2014)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: 생물의학을 위한 컨볼루션 네트워크 ical 이미지 분할. 에서: 미카이. (2015)
22. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: 이미지 분할을 위한 깊은 컨볼루션 인코더-디코더 아키텍처. 파미 (2017)

23. Chen, LC, Papandreou, G., Schroff, F., Adam, H.: atrous convolution에 대한 재고
시맨틱 이미지 분할을 위해 arXiv:1706.05587 (2017)
24. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: 피라미드 장면 구문 분석 네트워크. 안에:
CVPR.(2017)
25. He, K., Zhang, X., Ren, S., Sun, J.: 이미지 인식을 위한 딥 레지듀얼 학습.
에서: CVPR. (2016)
26. Chollet, F.: Xception: 깊이별 분리 가능한 컨볼루션을 사용한 딥 러닝. 안에:
CVPR.(2017)
27. Sifre, L.: 이미지 분류를 위한 강성 모션 산란. 박사 논문(2014)
28. Vanhoucke, V.: 시각적 표현을 대규모로 학습합니다. ICLR 초청 강연(2014)
29. Howard, AG, Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., An dreetto, M., Adam,
H.: Mobilenets: 모바일을 위한 효율적인 컨볼루션 신경망 비전 응용 프로그램. arXiv:1704.04861 (2017)

30. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: 모바일 장치를 위한 매우 효율적인 컨볼루션 신경망. 에서:
CVPR. (2018)
31. Qi, H., Zhang, Z., Xiao, B., Hu, H., Cheng, B., Wei, Y., Dai, J.: 변형 가능한 컨볼루션 네트워크 - coco
detection and segmentation Challenge 2017 항목.
ICCV COCO 챌린지 워크숍 (2017)
32. Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.: 피드포워드 시맨틱 세그먼트
축소 기능이 있는 언급. 에서: CVPR. (2015)
33. Dai, J., He, K., Sun, J.: 조인트 개체 및 물건 분할을 위한 컨볼루션 기능 마스킹. 에서: CVPR. (2015)

34. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: 장면 라벨링을 위한 계층적 기능 학습. 파미 (2013)

35. Eigen, D., Fergus, R.: 공통 다중 스케일 컨볼루션 아키텍처를 사용하여 깊이, 표면 법선 및 의미 레이블 예측. 에서:
ICCV. (2015)
36. Pinheiro, P., Collobert, R.: 장면에 대한 반복적 컨볼루션 신경망
라벨링. 에서: ICML. (2014)
37. Lin, G., Shen, C., van den Hengel, A., Reid, I.: 깊은 깊이의 효율적인 조각별 훈련
의미론적 세분화를 위한 구조화된 모델. 에서: CVPR. (2016)
38. Chen, LC, Yang, Y., Wang, J., Xu, W., Yuille, AL: 규모에 대한 주의: 규모
시맨틱 이미지 분할을 인식합니다. 에서: CVPR. (2016)
39. Chen, LC, Papandreou, G., Kokkinos, I., Murphy, K., Yuille, AL: Deeplab: 깊은 컨볼루션 그물, atrous
컨볼루션 및 완전히 연결된 crfs를 사용한 의미론적 이미지 분할. 티파미 (2017)

40. Kr¨ahhenb¨uhl, P., Koltun, V.: 가우스 에지 전위가 있는 완전히 연결된 crfs의 효율적인 추론. 에서: NIPS.
(2011)
41. Adams, A., Beck, J., Davis, MA: per를 사용한 고속 고차원 필터링
다면체 격자. 에서: 유로그래픽스. (2010)
42. Chen, LC, Papandreou, G., Kokkinos, I., Murphy, K., Yuille, AL: 깊은 컨볼루션 네트와 완전히 연결된 crfs
를 사용한 시맨틱 이미지 분할. 에서: ICLR. (2015)

43. Bell, S., Upchurch, P., Snavely, N., Bala, K.: 컨텍스트 데이터베이스의 자료를 사용한 야생의 자료 인식. 에서:
CVPR. (2015)
44. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: 순환
신경으로서의 조건부 랜덤 필드 네트워크. 에서: ICCV. (2015)

45. Liu, Z., Li, X., Luo, P., Loy, CC, Tang, X.: 심층 분석 네트워크를 통한 의미론적 이미지 분할. 에서: ICCV. (2015)

46. Papandreou, G., Chen, LC, Murphy, K., Yuille, AL: 의미론적 이미지 분할을 위한 dcnn의 약한 및 반 지도 학
습. 에서: ICCV. (2015)

47. Schwing, AG, Urtasun, R.: 완전히 연결된 심층 구조 네트워크. arXiv:1503.02351 (2015)
48. Jampani, V., Kiefel, M., Gehler, PV: 희소 고차원 필터 학습: 이미지 필터링, 조밀한 crfs 및 양방향 신경망. 에서: CVPR. (2016)
49. Vemulapalli, R., Tuzel, O., Liu, MY, Chellappa, R.: 의미론적 분할을 위한 가우스 조건부 랜덤 필드 네트워크. 에서: CVPR. (2016)
50. Chandra, S., Kokkinos, I.: 깊은 Gaussian CRF를 사용한 의미론적 이미지 분할을 위한 빠르고 정확한 다중 스케일 추론. 에서: ECCV. (2016)
51. Chandra, S., Usunier, N., Kokkinos, I.: 깊은 임베딩을 사용하는 조밀하고 낮은 순위의 가우스 crfs. 에서: ICCV. (2017)
52. Liu, W., Rabinovich, A., Berg, AC: Parsenet: 더 잘 보기 위해 더 넓게 봅니다. arXiv:1506.04579 (2015)

53. Newell, A., Yang, K., Deng, J.: 인간 포즈 추정을 위한 누적 모래시계 네트워크 짝. 에서: ECCV. (2016)
54. Lin, TY, Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: 기능 물체 감지를 위한 피라미드 네트워크. 에서: CVPR. (2017)
55. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: 건너뛰기 연결 너머: 물체 감지를 위한 하향식 변조. arXiv:1612.06851 (2016)
56. Fu, CY, Liu, W., Ranga, A., Tyagi, A., Berg, AC: Dssd: Deconvolutional single shot detector. arXiv:1701.06659 (2017)
57. Noh, H., Hong, S., Han, B.: 의미론적 분할을 위한 학습 디콘볼루션 네트워크. 에서: ICCV. (2015)

58. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: 고해상도 의미론적 분할을 위한 ID 매핑이 있는 다중 경로 정제 네트워크. 에서: CVPR. (2017)

59. Pohlen, T., Hermans, A., Mathias, M., Leibe, B.: 거리 장면에서 의미론적 분할을 위한 전체 해상도 잔여 네트워크. 에서: CVPR. (2017)
60. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: 큰 커널 문제 - 글로벌 컨볼루션 네트워크에 의한 의미론적 분할 개선. 에서: CVPR. (2017)
61. 이슬람, MA, Rochan, M., Bruce, ND, Wang, Y.: 고밀도 이미지 라벨링을 위한 게이트 피드백 개선 네트워크. 에서: CVPR. (2017)
62. Wojna, Z., Ferrari, V., Guadarrama, S., Silberman, N., Chen, LC, Fathi, A., Uijlings, J.: 악마는 디코더에 있습니다. 에서: BMVC. (2017)
63. Fu, J., Liu, J., Wang, Y., Lu, H.: 의미론적 분할을 위한 누적 디콘볼루션 네트워크. arXiv:1708.04943 (2017)

64. Zhang, Z., Zhang, X., Peng, C., Cheng, D., Sun, J.: Exfuse: 의미론적 분할을 위한 기능 융합 향상. 에서: ECCV. (2018)
65. Xie, S., Girshick, R., Dollr, P., Tu, Z., He, K.: 집계된 잔차 변환 심층 신경망용. 에서: CVPR. (2017)
66. Jin, J., Dundar, A., Culurciello, E.: Flattened convolutional neural network for 피드포워드 가속. arXiv:1412.5474 (2014)
67. Wang, M., Liu, B., Foroosh, H.: 단일 채널 내 컨볼루션, 토폴로지 세분화 및 공간 "병목" 구조를 사용한 효율적인 컨볼루션 레이어 설계. arXiv:1608.04337 (2016)

68. Zoph, B., Vasudevan, V., Shlens, J., Le, QV: 양도 가능한 아키텍처 학습 확장 가능한 이미지 인식을 위해 에서: CVPR. (2018)
69. Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P.: 웨이블릿 변환의 도움으로 신호 분석을 위한 실시간 알고리즘. In: 웨이블릿: 시간-주파수 방법 및 위상 공간. (1989) 289–297 70. Giusti, A., Ciresan, D., Masci, J., Gambardella, L., Schmidhuber, J.: 깊은 최대 풀링 컨볼루션 신경망을 사용한 빠른 이미지 스캐닝. 에서: ICIP. (2013)

71. Papandreou, G., Kokkinos, I., Savalle, PA: 딥 러닝의 로컬 및 글로벌 변형 모델링: 에피토믹 컨볼루션, 다중 인스턴스 학습 및 슬라이딩 윈도우 감지. 에서: CVPR. (2015)

72. Abadi, M., Agarwal, A., et al.: Tensorflow: 열에 대한 대규모 기계 학습 성감대 분산 시스템. arXiv:1603.04467 (2016)
73. Hariharan, B., Arbel'aez, P., Girshick, R., Malik, J.: 개체 세분화 및 세분화된 지역화를 위한 하이퍼컬럼. 에서: CVPR. (2015)
74. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M ., Berg, AC, Fei-Fei, L.: ImageNet 대규모 시각적 인식 과제. IJCV (2015)

75. Ioffe, S., Szegedy, C.: 배치 정규화: 내부 공변량 이동을 줄여 심층 네트워크 훈련을 가속화합니다. 에서: ICML. (2015)
76. Hariharan, B., Arbel'aez, P., Bourdev, L., Maji, S., Malik, J.: 의미 윤곽 역 검출기에서. 에서: ICCV. (2011)
77. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: 의미론적 분할을 위한 이해 컨볼루션. arXiv:1702.08502 (2017)
78. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: 변형 가능한 회선 네트워크. In: ICCV(2017)

79. Lin, TY, et al.: Microsoft COCO: 컨텍스트의 공통 개체. 에서: ECCV. (2014)
80. Hinton, G., Vinyals, O., Dean, J.: 신경망에서 지식 증류. 에서: NIPS. (2014)
81. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: 불합리한 효과에 대한 재검토 딥러닝 시대의 데이터. 에서: ICCV. (2017)
82. Li, X., Liu, Z., Luo, P., Loy, CC, Tang, X.: 모든 픽셀이 동일한 것은 아닙니다. 깊은 레이어 캐스케이드를 통한 의미론적 세분화의 어려움. 에서: CVPR. (2017)
83. Wu, Z., Shen, C., van den Hengel, A.: 더 넓거나 더 깊숙이: resnet 모델 재방문 시각적 인식을 위해. arXiv:1611.10080 (2016)
84. Wang, G., Luo, P., Lin, L., Wang, X.: 의미 이미지 분할을 위한 학습 객체 상호 작용 및 설명. 에서: CVPR. (2017)
85. Luo, P., Wang, G., Lin, L., Wang, X.: 의미론적 이미지 분할을 위한 심층 이중 학습. 에서: ICCV. (2017)

86. Bu`o, SR, Porzi, L., Kontschieder, P.: 메모리에 대한 제자리 활성화 배치 표준 DNS 최적화 교육. 에서: CVPR. (2018)