DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Liang-Chieh Chen, George Papandreou, Senior Member, IEEE, Iasonas Kokkinos, Member, IEEE,
Kevin Murphy, and Alan L. Yuille, Fellow, IEEE

Abstract

우리는 딥러닝을 통한 semantic segmentation으로 3가지의 실용적인 이점을 보여줍니다. 먼저 atrous convolution는 딥러닝에서 해상도를 조절할 수 있게 해줍니다. 또한, 별도의 추가적인 연산 없이 필터의 크기를 키울 수 있습니다. 두번째로 다양한 크기의 객체에 대해 강건한 atrous spatial pyramid pooling(ASPP)를 제시합니다. ASPP는 convolution layer에서 다양한 크기를 사용하여 다양한 크기의 이미지 정보를 잡아낼 수 있습니다. 세번째로 DCNN과 확률적 그래프 모델을 결합하여 물체의 경계 위치를 더욱 개선시킵니다. 대부분의 DCNN에서의 max pooling은 분류에는 높은 정확도를 가져다 주지만 위치의 정확성은 떨어집니다. 우리는 fully connected Conditional Random Field(CRF)를 사용하여 이를 극복하고자 했습니다. 우리가 제안하는 DeepLab은 PASCAL VOC-2012에서 79.7%mIOU를 달성했습니다.

Introduction

DCNNs는 다양한 기본적인 문제에서 높은 성과를 나타내고 있다. 이러한 높은 성능은 불변성에 있다. 불변성은 다른 input이 들어갔을 때 같은 output을 나타내는 것이며 이는 데이터의 표현을 요약하기 때문에 나타나진다. 이러한 것은 미세한 테스크인 semantic segmentation과 같은 경우 성능을 떨어뜨리는 요소가 된다.

그래서 우리는 DCNNs의 semantic segmentation에서 다음 3가지 문제를 다루고자 한다.

(1) **Reduced feature resolution**

(2) **Existence of objects at multiple scales**

(3) **Reduced localization accuracy due to DCNN invariance**

(1) **Reduced feature resolution**

첫 번째 문제의 경우 max-pooling과 downsampling에 의해 발생된다. 자세히 말하자면 max-pooling이나 conv layer에 의해 피쳐맵은 점점 축약되어지며 작아지게 되는데 이러한 경우 원래 이미지에서의 정확한 정보값을 추론하는데 어려움이 있을 수 있다. 그래서 DeepLab에서는

기존의 마지막 몇몇 부분에 max-pooling기법을 제거하였으며, conv layer를 다른 upsample filters로 교체한다. upsample filter는 값 필터들 사이에 hole을 삽입하여 진행한다. 이러한 기법은 신호처리에서 많이 사용되어졌으며 효율적인 계산 방식에서 발전되었다.

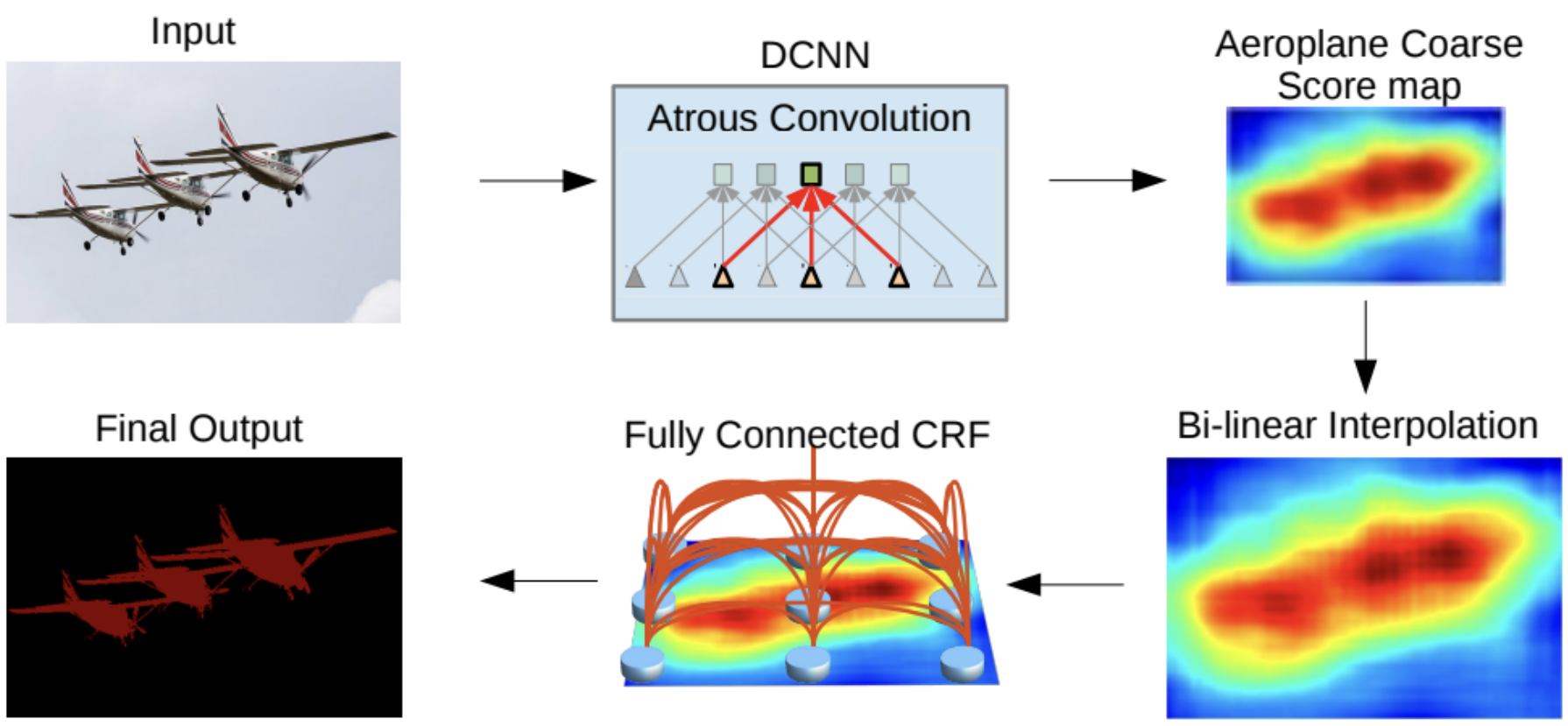
우리는 이러한 upsample filters를 atrous convolution이라 부른다. atrous convolution은 피쳐맵을 더 정밀하게 계산하며 원본 이미지의 크기로 변형시켜주는 간단한 보간이 따른다. 이러한 간단하며 효과적인 방식으로 deconvolutional layer를 대체할 수 있다. 일반적인 보통의 conv와 비교하여 atrous는 추가적인 연산량 없이 더욱 큰 필터의 관점을 가지게 해준다.

(2) **Existence of objects at multiple scales**

두 번째 문제의 경우 다양한 크기의 객체의 존재이다. 일반적으로는 같은 이미지에 대한 DCNN의 rescaled version을 통해 이를 해결한다. 이러한 방식은 좋은 성능으로 유도되지만, 그만큼 많은 컴퓨팅 자원이 필요하다. 대신에 SPP-net에서의 spatial pyramid pooling을 사용하면 작은 연산으로 convolution이전 다양한 규모에서의 피쳐를 얻을 수 있다. 다양한 크기의 필터를 통해 이미지를 확인하는 것은 다양한 관점 뿐만 아니라 다양한 크기에서 효과적이다. 실제로 피쳐를 리샘플링하는것이 아닌 다른 rate의 병렬적인 atrous convolutional layer는 효과적이었습니다. 이를 ASPP라 합니다.

(3) **Reduced localization accuracy due to DCNN invariance**

세 번째 문제는 DCNN의 불변성에 따른 class에 대한 좋은 성능과 위치 예측의 부족과 관련있습니다. 이러한 문제를 해결하기 위해 사용하는 방식으로 skip connection이 존재합니다. 우리는 자세한 부분에 대한 성능을 높여주기 위한 fully-connected Conditional Random Field (CRF)를 사용합니다. CRF는 semantic segmentation에서 multi-way classifier를 사용하여 class score를 결합하기 위해 광범위하게 사용되어집니다. CRF는 미세한 부분을 처리하기 위해 사용되어지며 효율적인 연산과 모서리 부분을 잘 포착해냅니다. 또한, 넓은 범위의 종속성을 포착합니다. 이러한 CRF는 성능 향상에 큰 도움이 되었습니다.



- A high-level illustration of the proposed DeepLab v2 model

먼저 우리는 pretrained 된 VGG-16 or ResNet-101을 사용합니다. 그리고 fc-layer부분을 cv-layer로 전환합니다. atrous cv-layer를 통해서 해상도를 증가시킵니다. 이는 원래 이미지의 8픽셀마다가 아닌 32픽셀마다 반응이 이루어지도록 합니다. 그리고 보간 기법을 업샘플링을 위해 진행합니다. 그 후, CRF에 인풋으로 집어넣습니다. 이러한 우리의 DeepLab의 장점은 크게 세가지입니다.

1. **Speed**

   atrous-conv에 의해 우리의 DCNN은 8FPS를 나타냅니다.

2. **Accuracy**

   우리는 여러 대회에서 Sota를 달성합니다.

3. **Simplicity**

   우리의 모델은 두개의 모듈 DCNN과 CRF의 계단식으로 구성되어있습니다.

RELATED WORK

semantic segmentation은 이전의 직접 피쳐를 만드는 방식으로부터 발전해왔습니다. 하지만 이러한 방식은 피쳐의 표현 부족으로 인해 한계에 부딪혔고, 지난 몇년간 딥러닝은 classification과 segmentation의 결합인 semantic segmentation을 발전시켜왔습니다.

첫 번째 발전 방식은 바텀업 방식입니다. 이 방식은 객체 내 여러개의 segmentation을 둔 후, 합쳐나가는 방식이다. 이러한 방식은 정밀한 바운더리를 예측하지만, 오류가 생길 경우 많은 오차가 발생한다.

두 번째 방식은 라벨링을 위한 conv를 통해 나온 피쳐와 독립적으로 얻은 segmentation 정보를 결합하는 방식입니다.

세 번째 방식은 DCNN을 직접적으로 category-level pixel label에 이용하는 것입니다. 이러한 방식의 접근은 DCNN의 마지막 fc-layer를 conv-layer로 변경합니다. cnn을 이용하여 공간의 정보가 변경되는 것은 업샘플링과 skip-connection방식으로 방지합니다. 이 논문에서 제시하는 방식은 여기에 속하며 우리는 여기다 해상도 조절, multi scale pooling, CRF등을 추가하여 진행합니다. 우리는 이러한 방식이 더 좋은 결과를 이끌거라 판단합니다. 또한 CRF는 이전에도 사용되어졌는데 이때는 계산 에러와 장기 종속성을 무시하여 한계가 존재했습니다. 우리의 방식은 모든 픽셀을 CRF node로 생각하여 진행합니다.

그리고 Gaussian CRF는 장기 의존성을 포착하며 동시에 필드에 대한 평균을 빠르게 잡을 수 있습니다. 필드에 대한 평균은 이전에도 많이 발전되었지만 단기 연결의 제한이 존재했습니다.

우리가 제시한 첫 번째 버전은 공개되어 semantic segmentation에서의 많은 발전을 거두었습니다. 대부분의 좋은 성능을 내는 모델들은 Atrous convolution을 채택하거나 fully connected CRF를 채택하였습니다.

METHODS

Atrous Convolution for Dense Feature Extraction and Field-of-View Enlargement

semantic segmentation에서 DCNN이 많이 사용되어지고 있습니다. 그러나 이는 해상도를 떨어뜨리게 되며 해결책으로 deconvolution을 사용하지만 이는 많은 메모리와 시간을 사용합니다. 우리는 대신에 atrous convolution을 사용할 것을 권고합니다. 이 알고리즘은 해상도를 원하는대로 설정하게 해줍니다.

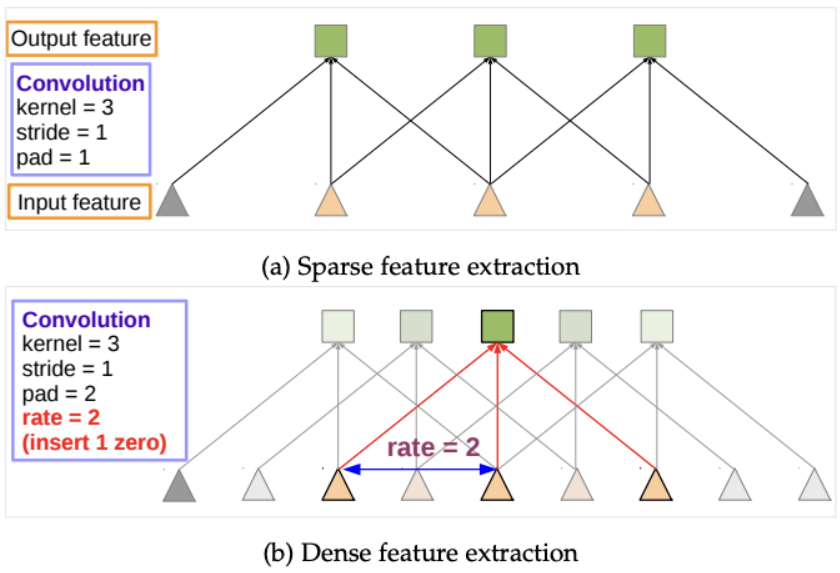


Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

Atrous convolution의 수식 : $y[i] = \Sigma_k x[i + r \cdot k] w[k]$

r은 stride(samping rate), k는 kernel, x는 input, y는 output을 의미합니다.

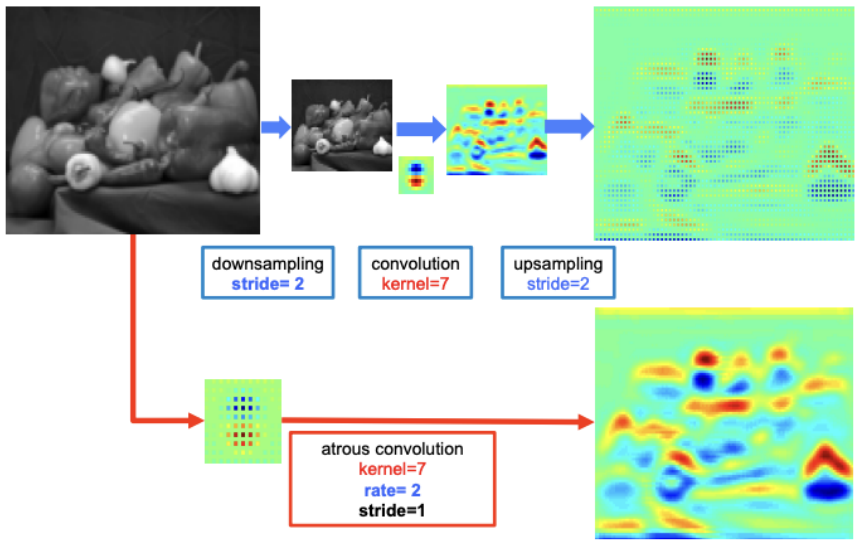


Fig. 3: Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

위에 사진을 설명하자면 위에 파란색 방향은 먼저 factor of 2로 2분의 1로 줄이는 다운샘플링을 우선 진행합니다. 그리고 convolution을 진행합니다. 그 후 업샘플링을 진행하고 피쳐맵을 원본 이미지에 이식하면 이미지 위치의 4분의 1정도에 해당하는 정보를 획득할 수 있습니다. 대신 우리의 atrous conv를 사용할 경우 모든 포지션에 대한 정보를 얻을 수 있습니다. 대신 필터 사이즈의 크기가 증가할지라도, hole의 값이 존재하기 때문에 연산량은 유지됩니다. 이처럼 작은 연산량을 통해 넓은 receptive filed를 커버하며, pooling이 존재하지 않아 최대한 원본 이미지의 해상도를 유지할 수 있습니다.

DCNN의 네트워크를 우리의 방식으로 변환하기 위해서 해상도를 감소시키는 끝 부분의 pooling과 conv layer를 변경시킵니다. pooling은 stride1, conv layer는 atrous conv로 변경합니다. 모든 layer에 대해 변경하고 싶지만, 해상도가 아예 유지되는 것은 비용이 너무 커지기 때문에 적절한 hybrid방식을 선택합니다. 그래서 우리는 기존 DCNN이 해상도가 32분의 1로 줄어드는 것에 비해 atrous conv를 통해 기존보다 해상도를 4배 증가시킵니다. 그리고 이후 blinear interpolation을 통해 8배를 증가시켜 원본 이미지와 동일하게 만듭니다. 여기서 기존의 DCNN에서 연산량이 많이 드는 deconv대신 blinear를 쓰지 않는 이유는 blinear는 너무 작아진 정보로부터 복구하는데 한계가 있기 때문! 이러한 방식을 통해 성능 증가 및 연산량 감소를 얻을 수 있다.

또한 atrous convolution은 넓은 범위의 필터를 통해 큰 물체를 판단해낼 수 있다. 일반적인 DCNN에서 conv의 필터는 3인데 3을 통해 큰 물체를 판단하기 위해서는 레이어가 깊어져야한다. 하지만 atrous convolution은 rate가 2만 되어도 필터가 3일 때 7*7의 물체를 확인할 수 있다.

max pooling X & atrous convolution

max pooling O & convolution

이라고 생각이 든다.

Atrous convolution layer를 적용하는 것은 두 가지 방법이 있는데,

1. filter에다가 구멍을 뚫어 놓기
2. feature map을 구역별로 sampling을 진행한 다음 각 부분에 대해 conv를 수행한다.

Multiscale Image Representations using Atrous Spatial Pyramid Pooling

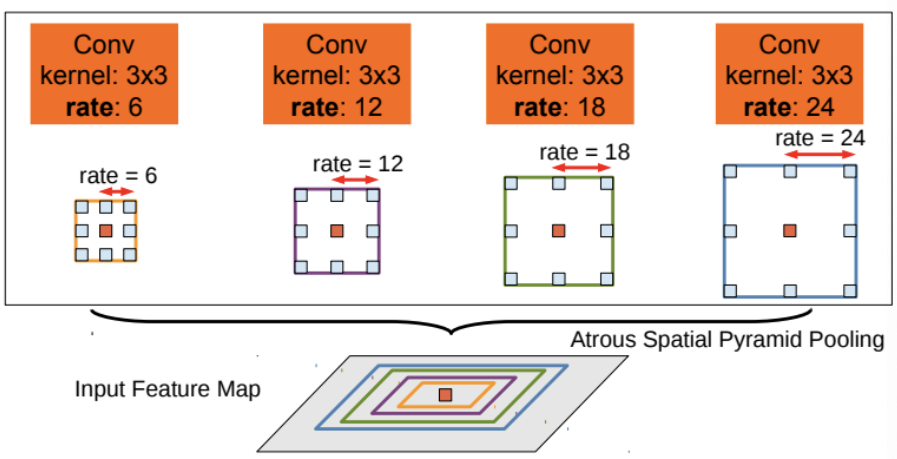
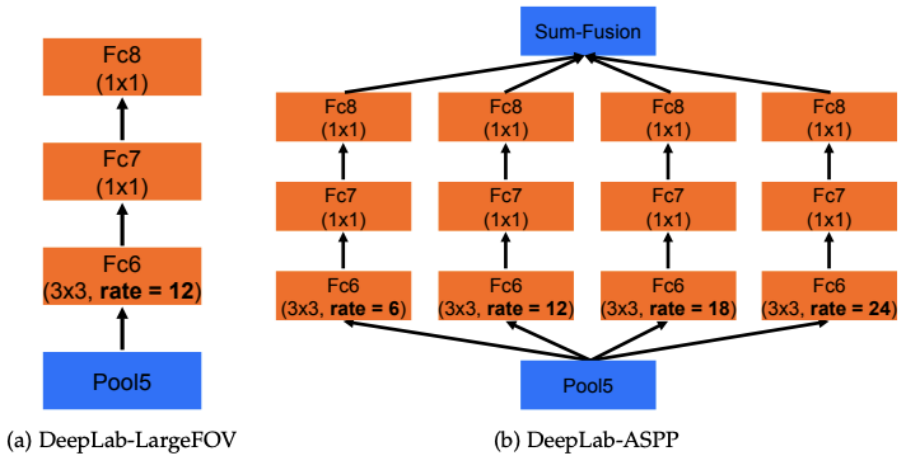


Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

이미지에는 큰 물체도 있고 작은 물체도 존재한다. 이를 위해 우리는 두가지 방식을 사용하였다. 첫 번째로 multiscale processing을 진행한다. 그 다음 각기 다른 크기의 featuremap을 보간을 통해 크기를 맞춘 후 이를 다 더 혼합하여 최종 결과물을 낸다. 이때 성능이 좋지만 많은 연산량을 요구한다.

두 번째는 우리가 제안하는 ASPP방식이며, V1과 다른 점이다. ASPP는 rate를 다양하게 사용하여 여러 규모의 특징을 표현하며 이를 더해 결과물을 만들어낸다.



(a) DeepLab-LargeFOV (b) DeepLab-ASPP

Structured Prediction with Fully-Connected Conditional Random Fields for Accurate Boundary Recovery

DCNN에서 depth와 pooling은 많은 성공을 거두었다. 특히 분류 문제에서는 객체의 유무 즉 객체의 큰 특징을 파악하는게 중요하였기 때문이다. 하지만 이에 따라 디테일한 예측은 힘들어지게 되고 invariance한 결과를 보인다.

우리는 이에따라 semantic segmentation에서 boundary부분을 디테일하게 잡아내기 위하여 CRF방식을 이용한다.

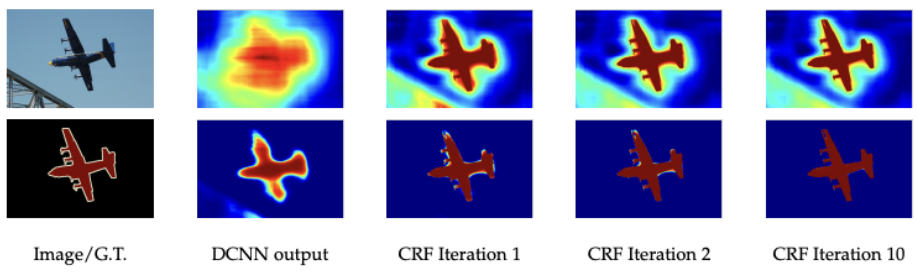


Fig. 5: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane. We show the score (1st row) and belief (2nd row) maps after each mean field iteration. The output of last DCNN layer is used as input to the mean field inference.

CRF는 noise를 스무스하게 만들어주며 비슷한 라벨값끼리 값을 유사하게 만들어주며 경계선을 만들어주는 방식이다. 수식은 다음과 같다.

$$E(x) = \Sigma_i \theta_i(x_i) + \Sigma ij \theta ij(x_i, x_j)$$

θi(xi) = − log P(xi)에서 P(xi)는 라벨 할당 확률을 나타내며

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2} \right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2} \right) \right] \quad (3)$$

오른쪽의 항은 위와 같다. $\mu(x_i, x_j) = 1$ if $x_i = / = x_j$ 이며 이는 같은 부분의 픽셀이 올 경우 건너 뛰는 것이다. θij (x_i, x_j)에서 왼쪽항의 경우 픽셀의 위치와 RGB에 해당하는 차이값에 대한 패널티이며 유사한 색깔과 위치를 가진 픽셀이 비슷한 라벨을 같도록 한다. 오른쪽 항은 위치 정보 차이에 따른 패널티 값이며 유사한 위치를 가진 픽셀이 비슷한 라벨을 갖도록 한다. 이를 통해 학습을 할수록 피쳐맵의 노이즈가 제거 되며 detatied한 정보를 표현할 수 있게 된다.

EXPERIMENTIAL RESULTS

우리의 모델은 이미지 넷으로 학습한 VGG-16과 RESNET-101을 사용하였다. 그리고 기본적인 로스는 픽셀마다의 cross-entropy를 사용하며 모든 포지션과 라벨은 동등한 가중치값을 가진다. (배경을 제외한) 또한 학습은 DCNN과 CRF를 따로 구분하여 진행한다.

우리는 PASCAL VOC 2012, PASCAL-Context, PASCALPerson-Part, and Cityscapes 데이터셋에 대해 실험을 진행하였다.

Kernel	Rate	FOV	Params	Speed	bef/aft CRF
7×7	4	224	134.3M	1.44	64.38 / 67.64
4×4	4	128	65.1M	2.90	59.80 / 63.74
4×4	8	224	65.1M	2.90	63.41 / 67.14
3×3	12	224	20.5M	4.84	62.25 / 67.64

TABLE 1: Effect of Field-Of-View by adjusting the kernel size and atrous sampling rate r at 'fc6' layer. We show number of model parameters, training speed (img/sec), and *val* set mean IOU before and after CRF. DeepLab-LargeFOV (kernel size 3×3, $r = 12$) strikes the best balance.

fc6 layer에서 atrous conv의 rate와 kernel을 바꿔가며 실행한 결과이다. kernel은 3이고 r=12일 때 가장 좋은 성능을 보인다.

Learning policy	Batch size	Iteration	mean IOU
step	30	6K	62.25
poly	30	6K	63.42
poly	30	10K	64.90
poly	10	10K	64.71
poly	10	20K	65.88

TABLE 2: PASCAL VOC 2012 *val* set results (%) (before CRF) as different learning hyper parameters vary. Employing "poly" learning policy is more effective than "step" when training DeepLab-LargeFOV.

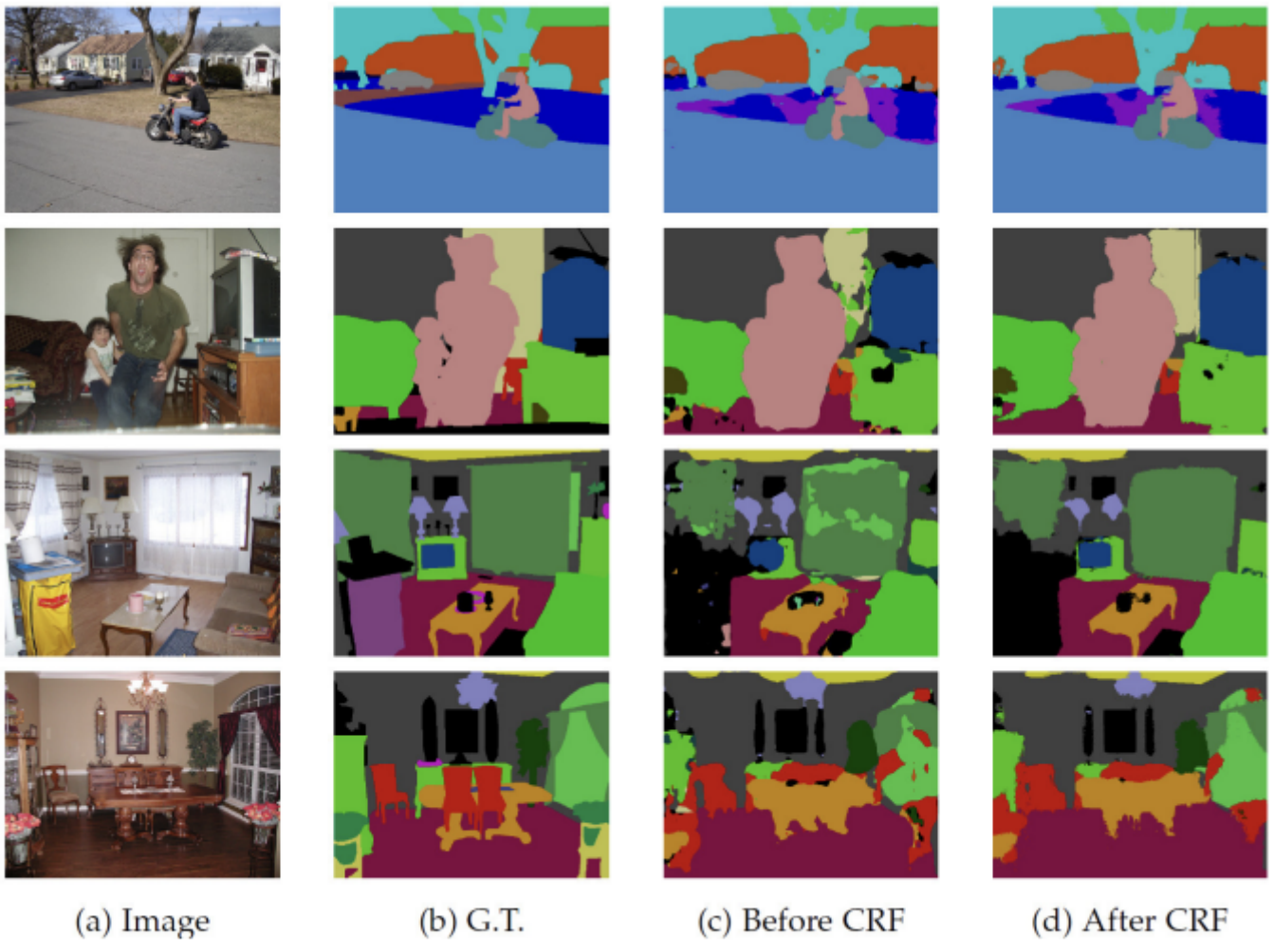
배치와 학습량을 바꿔가며 실험한 결과 poly 방식의 step은 아래와 같다.

$$\left(1 - \frac{iter}{max\_iter}\right)^{power}$$

Method	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
VGG-16							
DeepLab [38]				✓			37.6
DeepLab [38]				✓		✓	39.6
ResNet-101							
DeepLab							39.6
DeepLab	✓		✓				41.4
DeepLab	✓	✓	✓				42.9
DeepLab	✓	✓	✓	✓			43.5
DeepLab	✓	✓	✓		✓		44.7
DeepLab	✓	✓	✓		✓	✓	45.7
O_2P [45]							18.1
CFM [51]							34.4
FCN-8s [14]							37.8
CRF-RNN [59]							39.3
ParseNet [86]							40.4
BoxSup [60]							40.5
HO_CRF [91]							41.3
Context [40]							43.3
VeryDeep [93]							44.5

TABLE 6: Comparison with other state-of-art methods on PASCAL-Context dataset.

Sota를 달성!



한계점

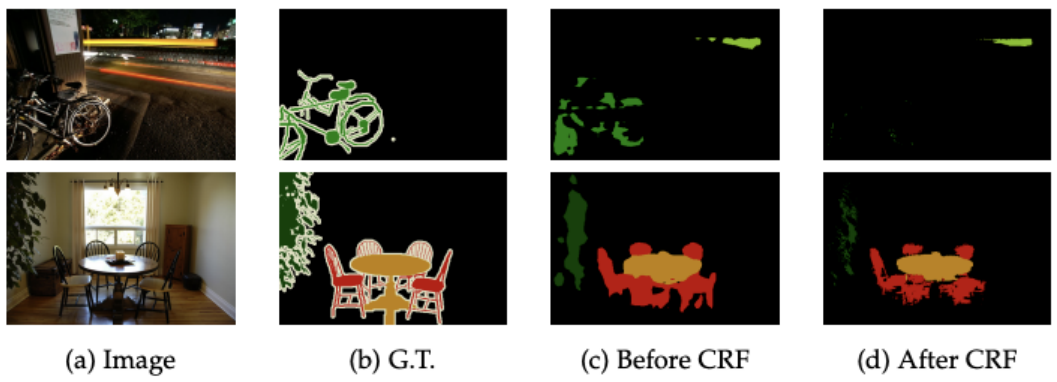


Fig. 14: Failure modes. Input image, ground-truth, and our DeepLab results before/after CRF.

PASCAL VOC 2-12 val set에서 예측을 잘 못하는 경우가 있었다. CRF 이후 디테일한 부분을 잘 살려내지 못하였다. 우리는 encoder-decoer구조를 사용한다면 이러한 문제를 완화시킬 수 있을거라 생각합니다.

CONCLUSION

우리가 제시한 “DeepLab”시스템은 Atrous convolution을 이용하여 네트워크를 재구축해냈습니다. 또한 CRF를 적용하여 boundary에 대해 더욱 자세히 예측한 결과를 내보입니다.