

# Deconvolution Network (DeconvNet)

Learning Deconvolution Network for Semantic Segmentation

# 0. Abstract

## • (1) Abstract

### ✓ Deconvolution Network[1]의 Contribution

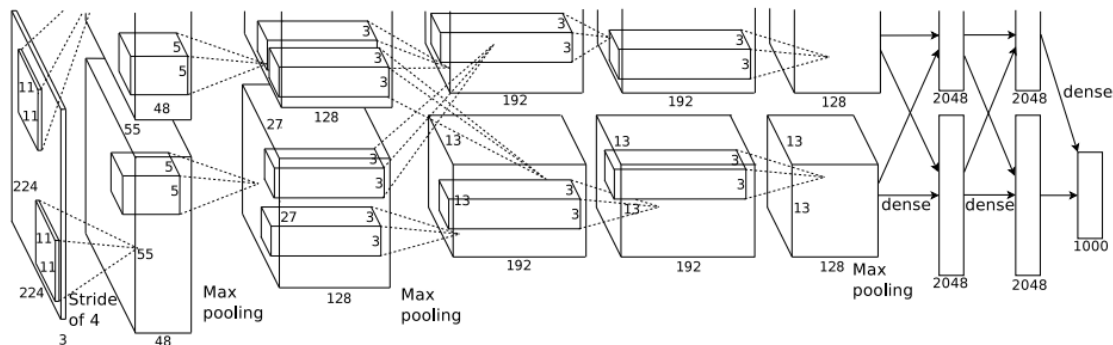
1. Deconvolution + Unpooling을 통해 Deconvolution Network를 구성
2. Pretrained된 네트워크를 이용 (VGG-16)
3. Proposals를 추출해서 예측한 후에 결과들을 결합
4. FCN의 한계를 극복 (large, small object에 대한 성능)
  - 새로운 Deconvolution Network를 제안
  - Proposal-wise prediction
5. multi-scale에 강한 모습을 보임
6. PASCAL VOC 2012에서 72.5%의 정확도를 보임
  - FCN과의 Ensemble 없이도 높은 성능을 보임

# 1 Introduction

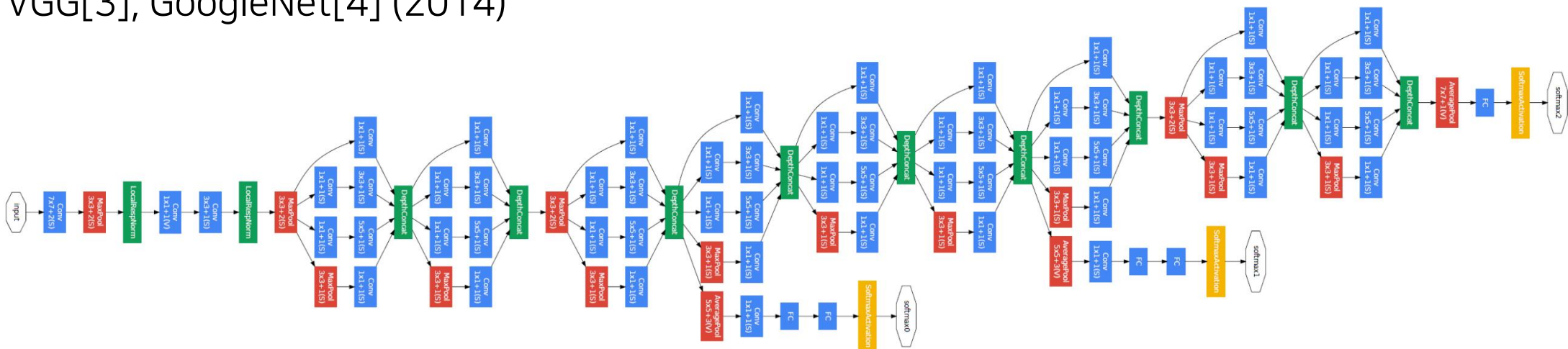
- (1) Convolutional networks

✓ Recognition에서 Deep Learning 네트워크의 개발에 따른 다른 분야로의 확장

AlexNet (2012) [2]



VGG[3], GoogleNet[4] (2014)

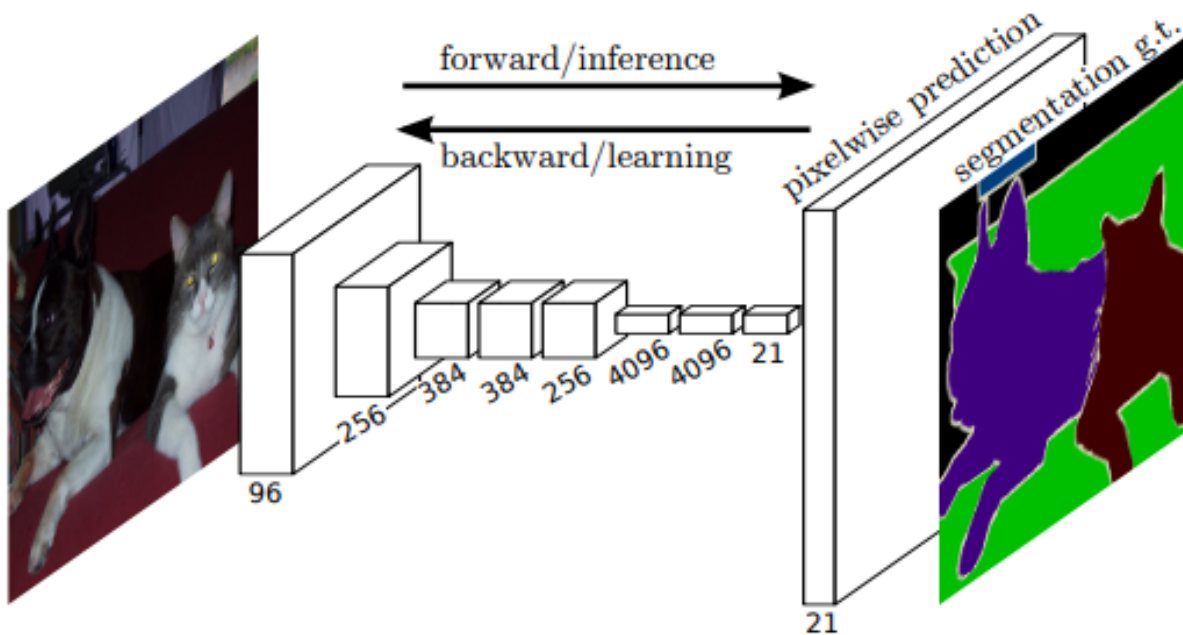


# 1 Introduction

- (1) Convolutional networks

✓ Recently Approaches convnets for semantic segmentation

Fully convolutional networks for semantic segmentation

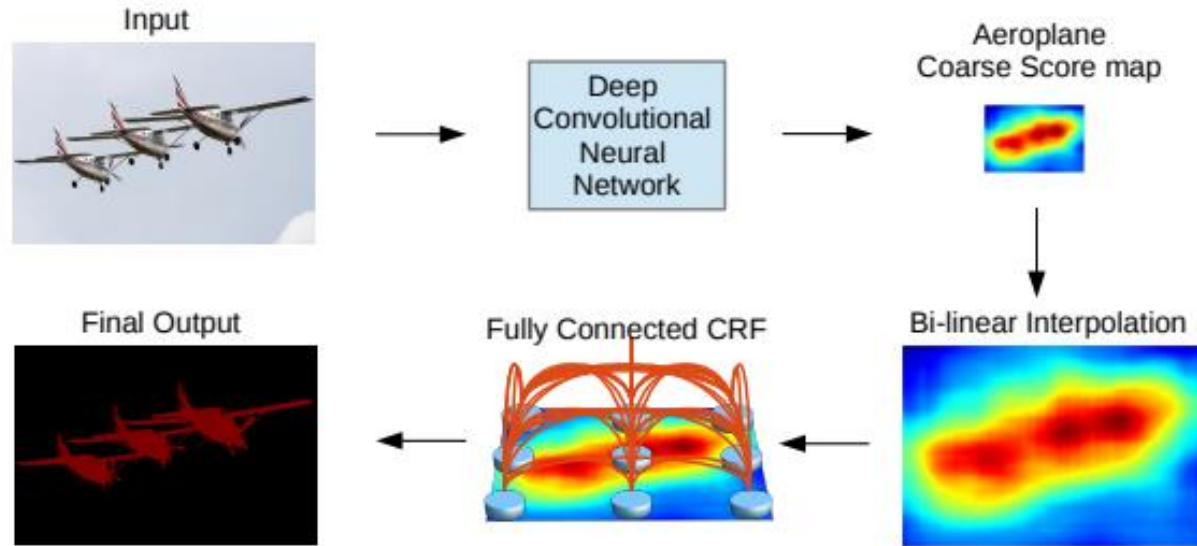


# 1. Introduction

- (1) Convolutional networks

✓ Recently Approaches convnets for semantic segmentation

SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS

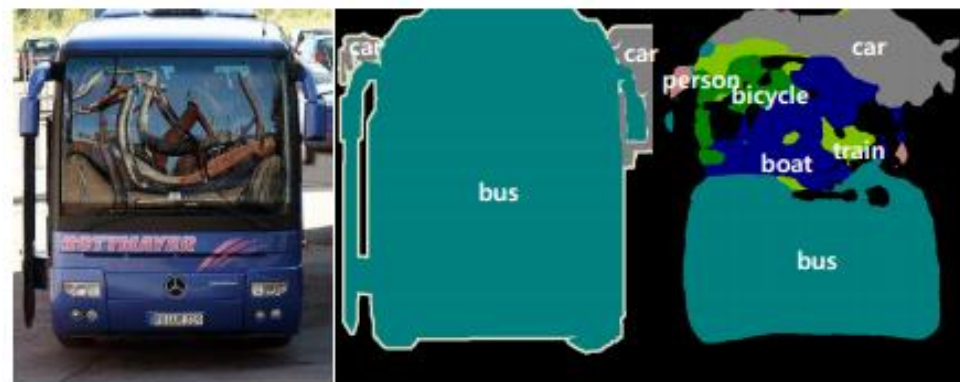


# 1. Introduction

## • (2) FCN의 한계

### ✓ FCN 기반의 Semantic Segmentation이 가지는 한계

1. Predefined fixed size receptive field를 가짐
  - Object가 Receptive field보다 크거나 작은 경우 성능이 낮음
  - 큰 Object의 지역적인 정보로만 예측
  - 같은 Object여도 다르게 라벨링을 함
  - 작은 Object가 무시되는 문제가 있음
  - Skip architecture를 이용해도 한계가 있음



(a) Inconsistent labels due to large object size



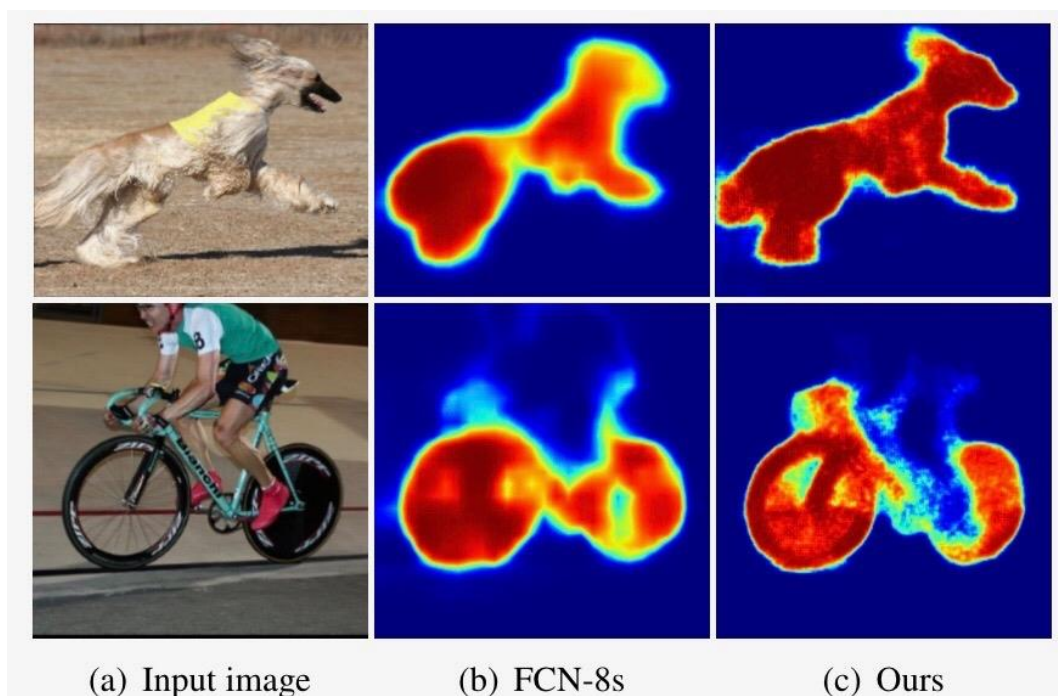
(b) Missing labels due to small object size

# 1. Introduction

## • (2) FCN의 한계

### ✓ FCN 기반의 Semantic Segmentation이 가지는 한계

2. Object의 디테일한 모습이 사라지는 문제가 발생
  - 데이터 자체의 문제 (coarse)
  - Deconvolution 절차가 너무 간단함 (1 layer)
  - 단순한 절차가 nonlinear한 경계를 학습하기 어려움
    - CRF를 이용해서 어느정도 해결





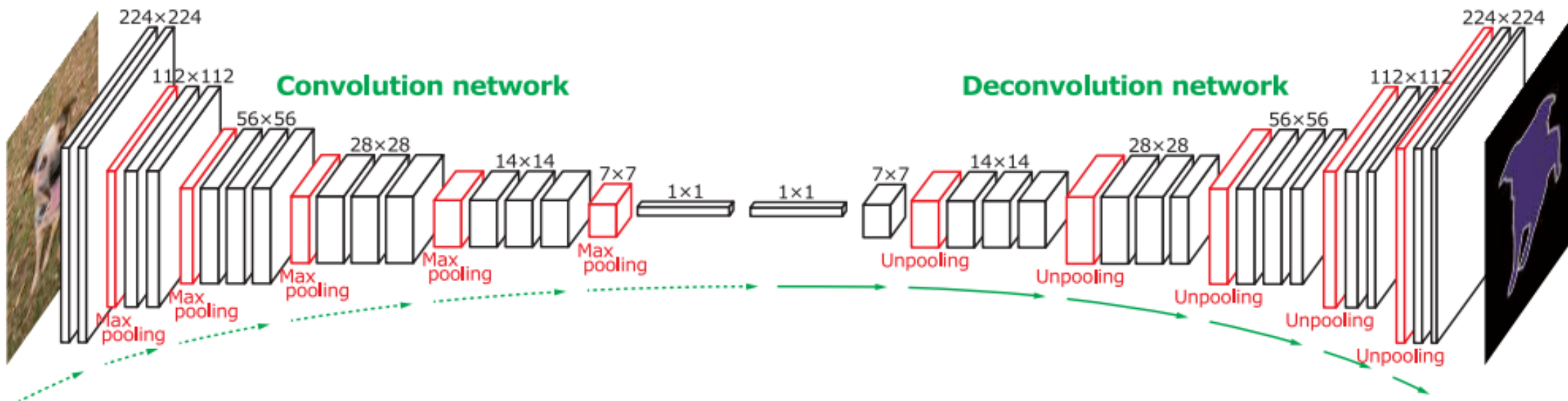
# 1 Introduction

- (3) FCN의 한계 극복

- ✓ FCN의 한계를 극복하는 2가지 방식

1. Deep Deconvolution Network

- Deconvolution + unpooling + ReLU





# 1 Introduction

- (3) FCN의 한계 극복

- ✓ FCN의 한계를 극복하는 2가지 방식

- 2. Instance-wise Segmentation을 적용

- individual object proposals 후에 segmentation
- FCN의 Scale 이슈를 해결
- Object의 Detail을 확인



# 1. Introduction

- (3) FCN의 한계 극복

## ✓ FCN의 한계를 극복하는 2가지 방식

### 3. 위의 2가지 방식을 이용해서 PASCAL VOC 2012에서 좋은 성능을 달성

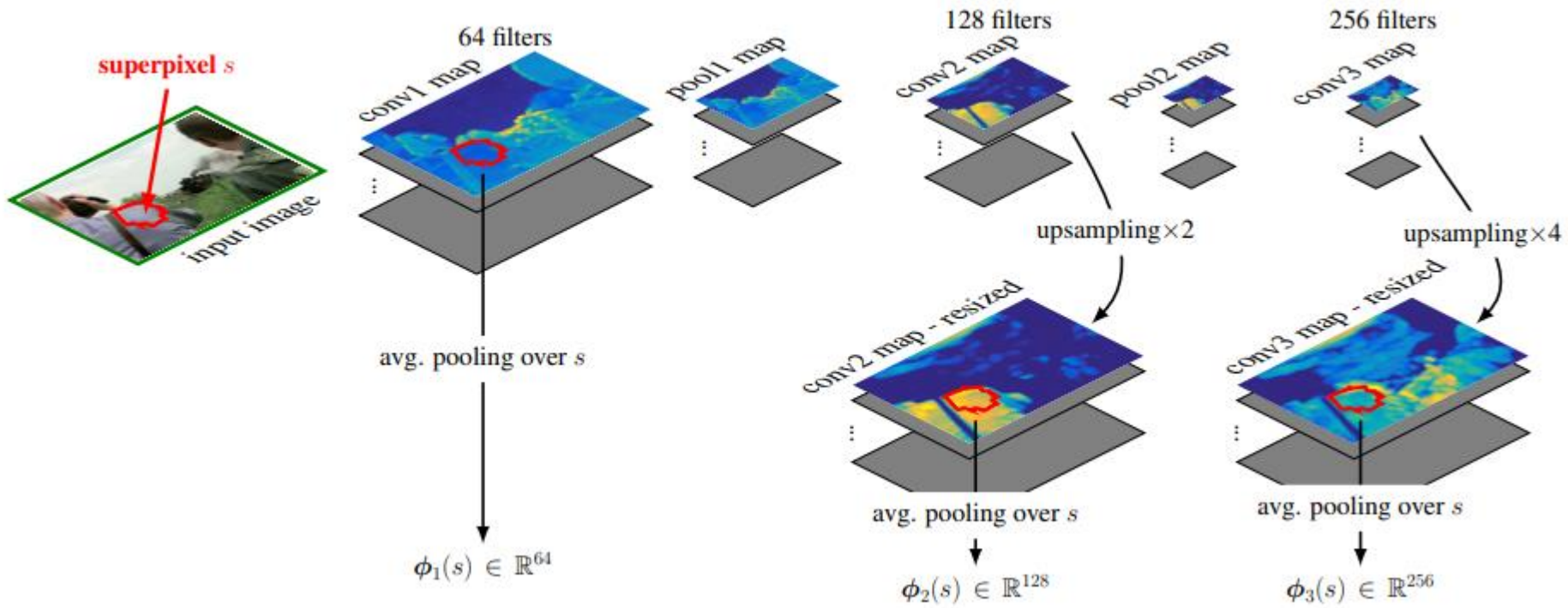
- Augmented Dataset 사용
- FCN과 Ensemble
- Ensemble한 결과에 CRF 적용

Table 1. Evaluation results on PASCAL VOC 2012 test set. (Asterisk (\*) denotes the algorithms that also use Microsoft COCO for training.)

Method	bkg	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
Hypercolumn [11]	88.9	68.4	27.2	68.2	47.6	61.7	76.9	72.1	71.1	24.3	59.3	44.8	62.7	59.4	73.5	70.6	52.0	63.0	38.1	60.0	54.1	59.2
MSRA-CFM [3]	87.7	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN8s [19]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
TTI-Zoomout-16 [20]	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [1]	93.1	84.4	<b>54.5</b>	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	<b>59.7</b>	82.2	50.4	73.1	63.7	71.6
DeconvNet	92.7	85.9	42.6	78.9	62.5	66.6	87.4	77.8	79.5	26.3	73.4	60.2	70.8	76.5	79.6	77.7	58.2	77.4	52.9	75.2	59.8	69.6
DeconvNet+CRF	92.9	87.8	41.9	80.6	63.9	67.3	<b>88.1</b>	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	<b>55.0</b>	75.2	61.5	70.5
EDeconvNet	92.9	88.4	39.7	79.0	63.0	67.7	87.1	<b>81.5</b>	84.4	27.8	76.1	61.2	78.0	79.3	83.1	79.3	58.0	82.5	52.3	80.1	64.0	71.7
EDeconvNet+CRF	93.1	<b>89.9</b>	39.3	79.7	63.9	<b>68.2</b>	87.4	81.2	<b>86.1</b>	28.5	<b>77.0</b>	62.0	79.0	<b>80.3</b>	<b>83.6</b>	80.2	58.8	<b>83.4</b>	54.3	<b>80.7</b>	65.0	<b>72.5</b>
* WSSL [21]	93.2	85.3	36.2	<b>84.8</b>	61.2	67.5	84.7	81.4	81.0	<b>30.8</b>	73.8	53.8	77.5	76.5	82.3	<b>81.6</b>	56.3	78.9	52.3	76.6	63.3	70.4
* BoxSup [2]	<b>93.6</b>	86.4	35.5	79.7	<b>65.2</b>	65.2	84.3	78.5	83.7	30.5	76.2	<b>62.6</b>	<b>79.3</b>	76.1	82.1	81.3	57.0	78.2	<b>55.0</b>	72.5	<b>68.1</b>	71.0

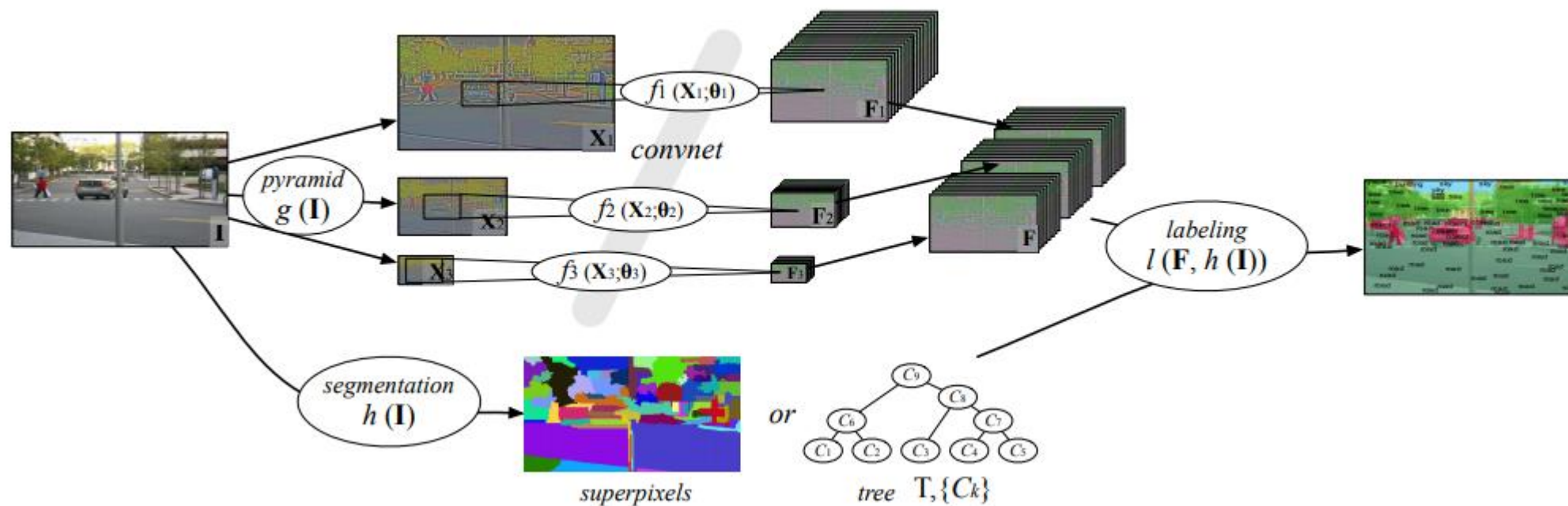
# 2. Related Work

## (1) Multi-scale superpixels



# 2. Related Work

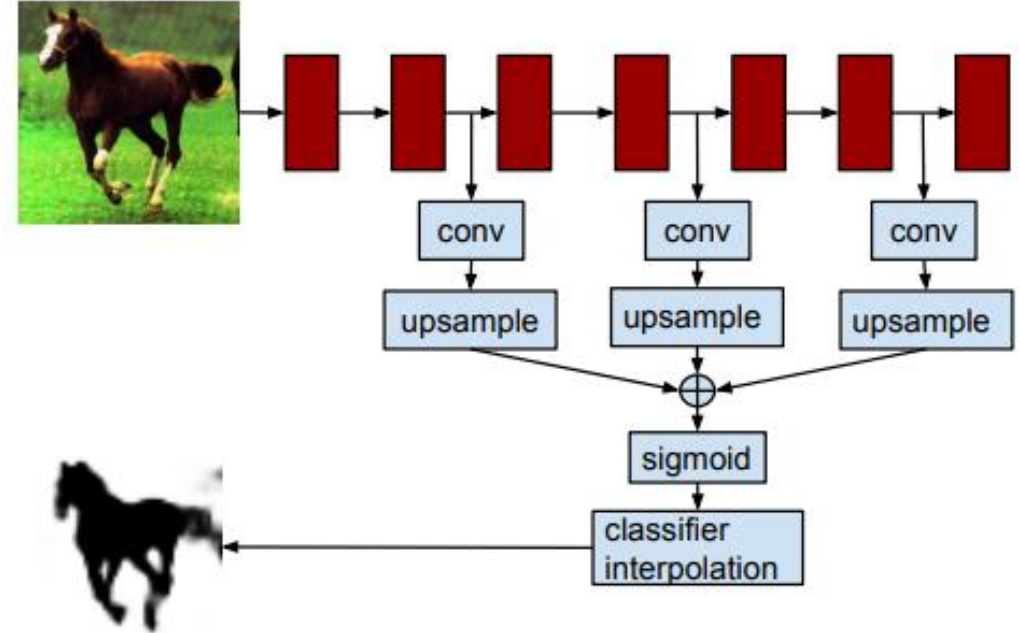
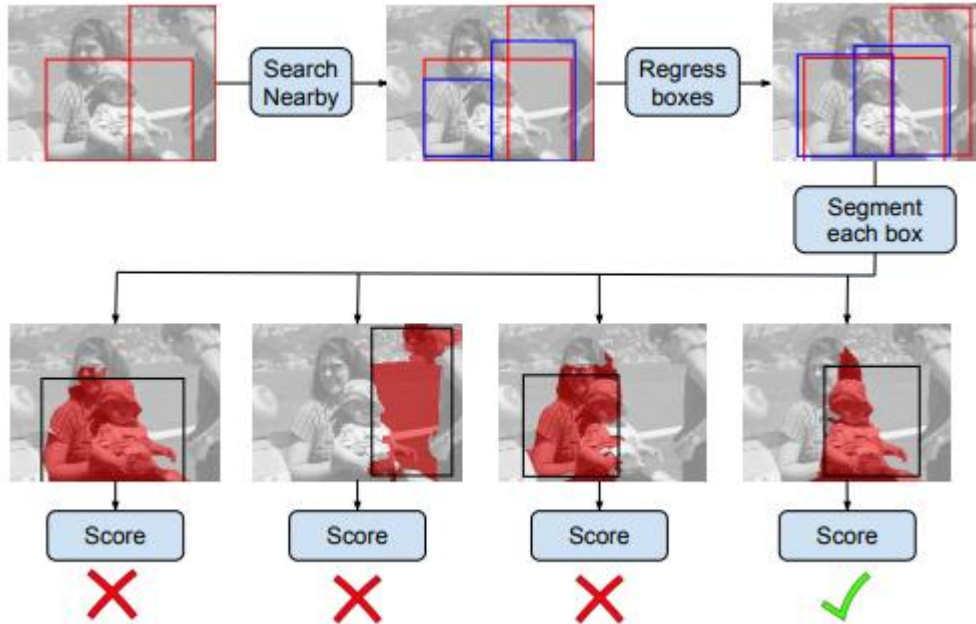
## (1) Multi-scale superpixels





# 2. Related Work

## (2) Region proposals



# 2. Related Work

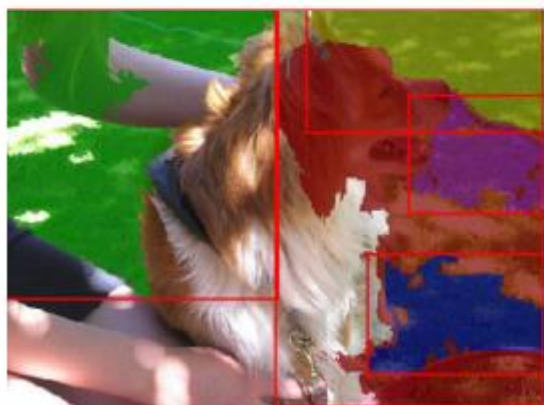
- (2) Region proposals



(a) image



(b) uniform

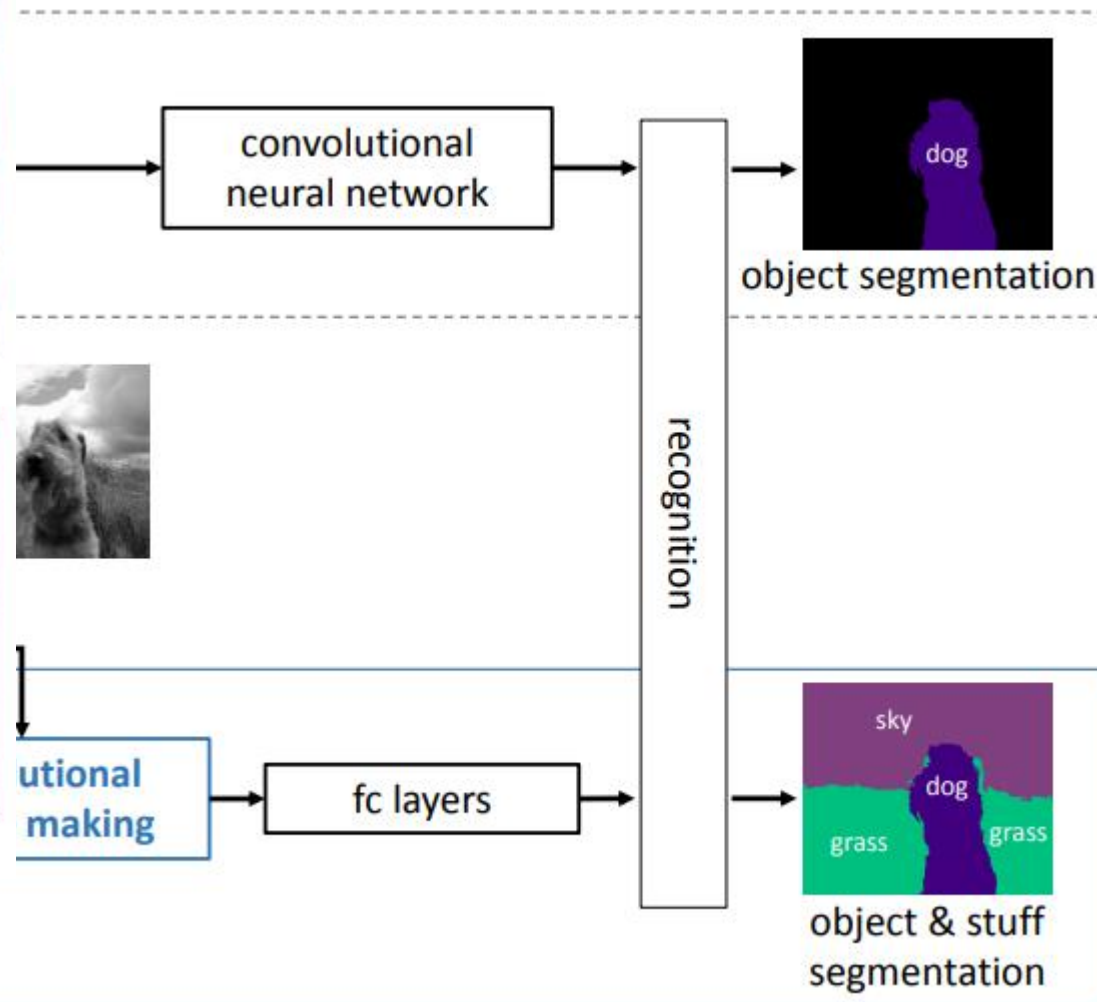


(c) deterministic segment pursuit



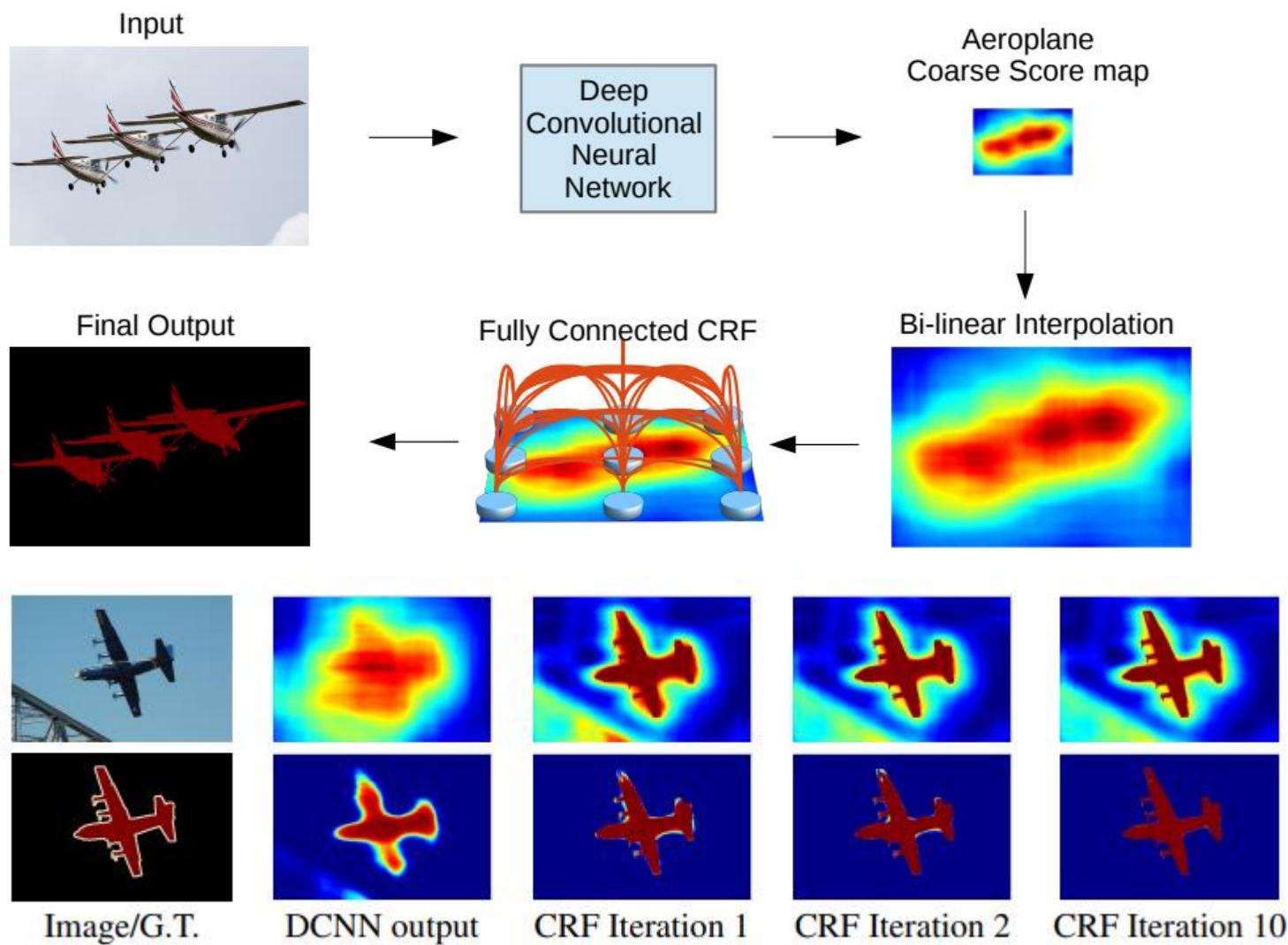
(d) stochastic segment pursuit  
feature maps

**feature-map-based (ours)**



# 2. Related Work

## • (3) Conditional Random Field (CRF)



출처 : P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In NIPS, 2011

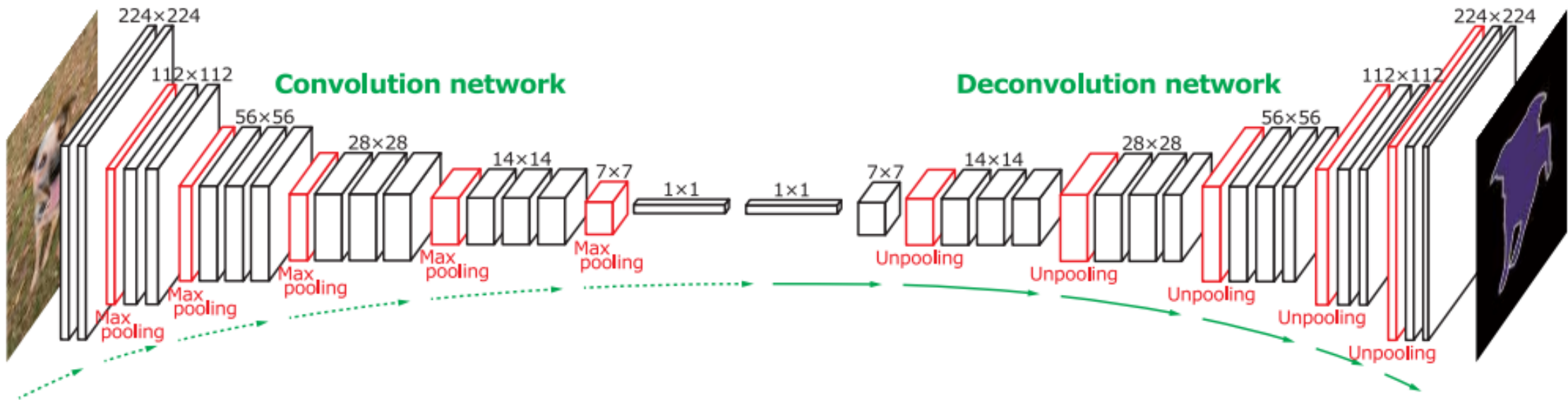
출처 : Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Chen et al. ICLR'15



# 3. System Architecture

## (1) Architecture

### ✓ DeconvNet Architecture – Convolution + Deconvolution Network

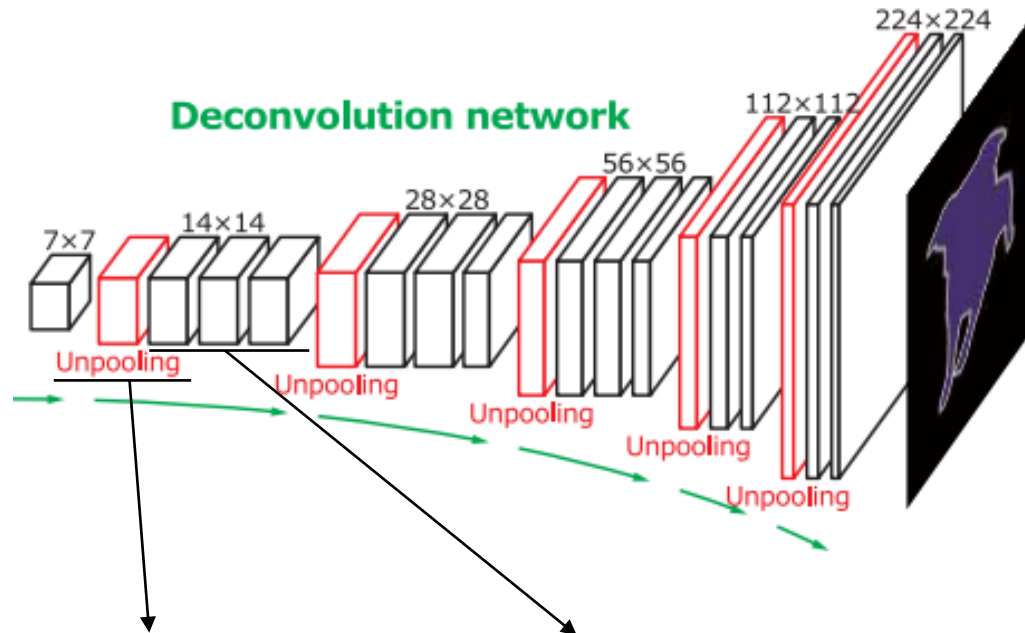


1. Convolution Network / Deconvolution Network가 대칭으로 이루어진 구조
2. Convolution Network는 VGG16을 사용
  - 13개의 층으로 이루어짐
  - RELU와 POOLING이 CONVOLUTION 사이에서 이루어짐
  - 1 X 1 Convolution 2개가 클래스의 투영을 진행함
3. Deconvolution Network는 Unpooling, deconvolution, ReLU으로 이루어짐

# 3. System Architecture

## (2) Deconvolution Network for Segmentation

### ✓ Deconvolution Network – Unpooling + Transposed Convolution



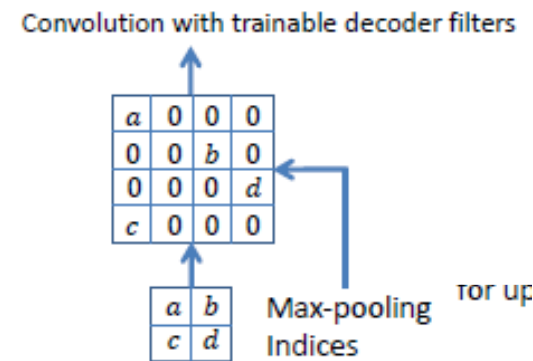
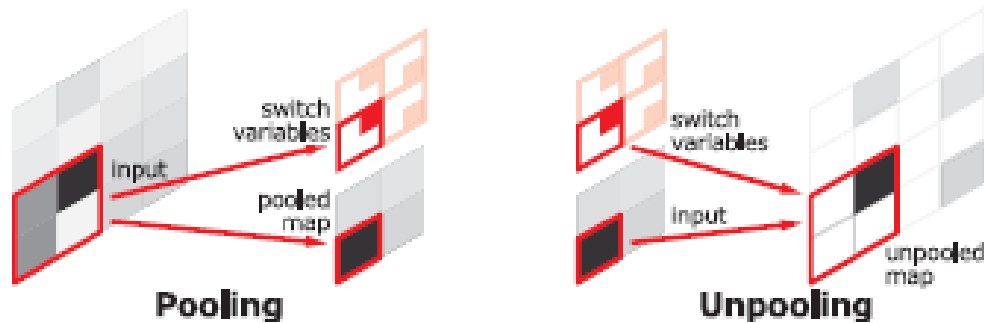
Unpooling과 Transposed Convolution이 반복적으로 이루어진 형태

- Unpooling은 Detail한 경계를 포착
- Transposed Convolution은 전반적인 모습을 포착

# 3. System Architecture

## (2) Deconvolution Network for Segmentation

### ✓ Unpooling



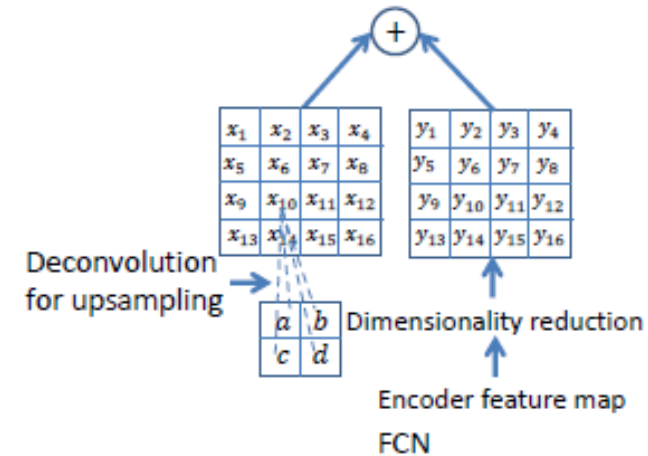
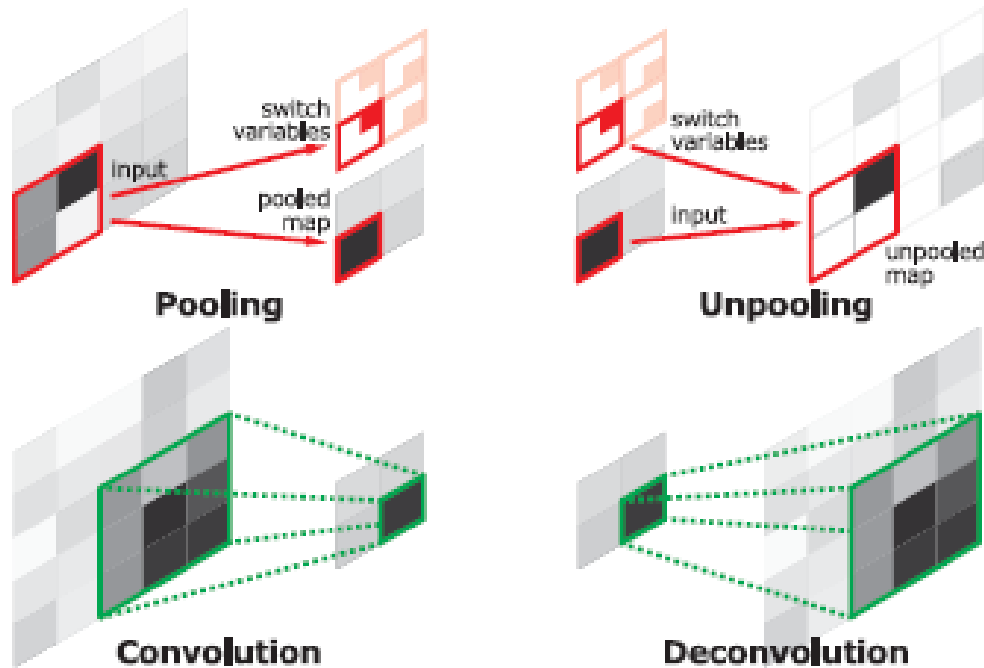
Pooling의 경우 노이즈를 제거하는 장점이 있지만 그 과정에서 정보가 손실되는 문제가 생김

- Unpooling을 통해서 Pooling시에 지워진 정보를 기록했다가 복원
- 학습이 필요 없기에 속도가 빠르고 경계를 잘 기억하는 장점이 있음
- 하지만, sparse한 activation map을 가지기에 이를 채워 줄 필요가 있음
  - 채우는 역할을 Transposed Convolution이 수행

# 3. System Architecture

## (2) Deconvolution Network for Segmentation

### ✓ Deconvolution (Transposed Convolution)



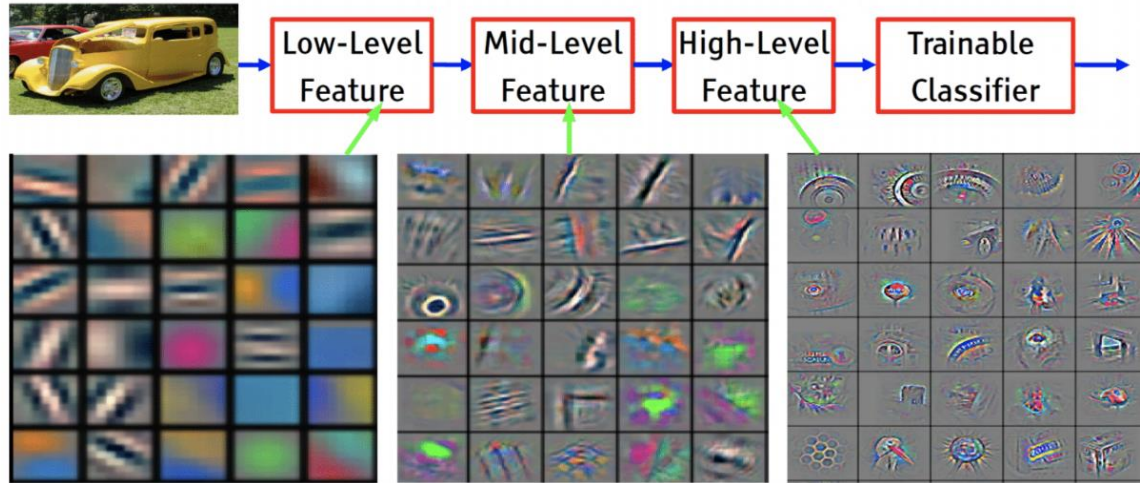
Pooling의 경우 노이즈를 제거하는 장점이 있지만 그 과정에서 정보가 손실되는 문제가 생김

- Deconvolutional layers를 통해서 input object의 모양을 복원
- 순차적인 층의 구조가 다양한 수준의 모양을 잡아냄
  - 낮은 층의 경우 전반적인 모습을
  - 깊은 층의 경우 구체적인 모습을 잡아냄

# 3. System Architecture

## (2) Deconvolution Network for Segmentation

### ✓ Analysis of Deconvolution Network



Low Layer

- 직선 및 곡선, 색상 등의 낮은 수준의 특징 (local feature)

High Layer

- 복잡하고 포괄적인 개체 정보가 활성화 (global feature)

Deconvolution Network의 Activation Map을 보면 층과 Pooling 방법에 따라 다른 특징이 있음

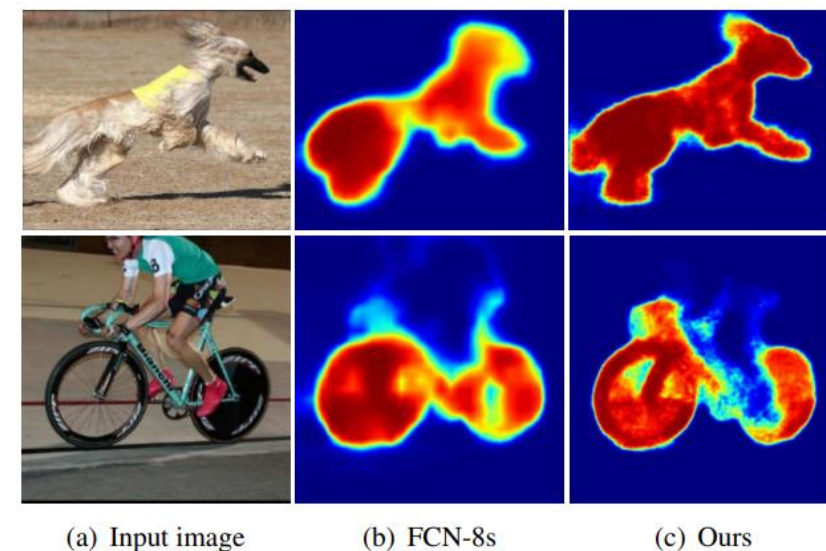
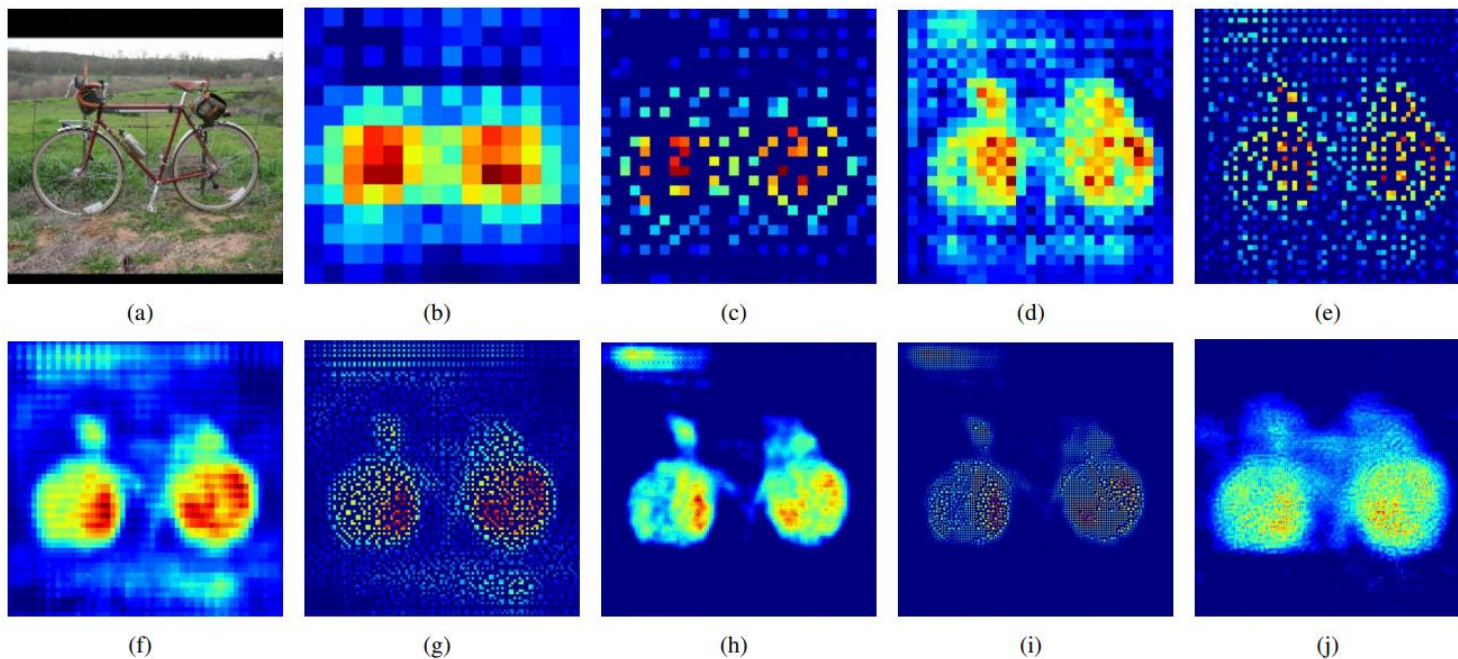
- 낮은 층의 경우 전반적인 특징 (location, shape and region) / 깊은 층의 경우 복잡한 패턴을 잡아냄



# 3. System Architecture

## (2) Deconvolution Network for Segmentation

### ✓ Analysis of Deconvolution Network



Deconvolution Network의 Activation Map을 보면 층과 Pooling 방법에 따라 다른 특징이 있음

- 낮은 층의 경우 전반적인 특징 (location, shape and region) / 깊은 층의 경우 복잡한 패턴을 잡아냄
- unpooling의 경우 "example-specific"한 구조를 잡아냄 (자세한 구조를 잡아냄)
- Transposed Conv의 경우 "class-specific"한 구조를 잡아냄 (위의 구조에 빈 부분을 채워넣음)
  - 실제로 둘을 병행시에 FCN 보다 활성화 맵이 디테일해지는 모습을 볼 수 있음

# 3. System Architecture

## (3) System Overview



### ✓ Semantic segmentation to instance wise segmentation

image로부터 object가 있을 법한 candidate를 추출해서 Segmentation을 수행

- 다양한 크기의 Object를 효과적으로 다룸
- 고정된 크기의 Receptive Field가 가지는 문제를 해결 (sub-level image의 size가 달라서인듯)
- 학습에 필요한 메모리와 복잡도를 줄일 수 있음
  - 필요한 Object에 대해서만 예측을 수행하면 됨



# 4. Training

## • (1) Batch Normalization

✓ Reduce internal covariate shift problem

**Input:** Values of  $x$  over a mini-batch:  $\mathcal{B} = \{x_1 \dots x_m\}$ ;

Parameters to be learned:  $\gamma, \beta$

**Output:**  $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

# 4. Training

## • (2) Two-stage Training

### ✓ Segmentation 테스트 대비 적은 데이터의 수를 해결

1. Batch Normalization으로 local optima에 빠지는 문제를 어느정도 해결하지만, 충분하지 않음
  - 학습수가 적어서 Deconvolution network의 장점이 감소
2. Two-stage의 학습방법을 도입해서 위의 문제를 해결
  - 1 stage에서는 쉬운 예제를 학습
    - object를 중심으로 하는 object instance를 crop함
    - 이렇게 object의 위치와 크기의 변동을 줄임으로써 적은 학습 데이터로 효과적인 성능을 낼 수 있음 ◯ □
  - 2 stage에서는 보다 어려운 예제를 통해서 fine-tuning의 과정을 거침
    - 보다 어려운 example을 통해서 fine tuning하는 과정을 거침
    - ground-truth와 overlapped이 IoU 0.5 이상인 후보자들만 추출
    - 2 stage를 통해서 네트워크가 보다 robust 해짐 (물체의 위치와 크기가 훈련의 예제에 따라 다를 수 있기에 훈련을 더 어렵게 만듦)

# 5. Inference

(0) Overview



## ✓ Inference 과정

1. Individual instance에 대해서 semantic segmentation의 학습을 수행
2. 평가 데이터에 대해, candidate proposals (individual proposals)를 생성하고 학습된 모델을 적용
3. 2에서 만들어진 모든 결과를 aggregate해서 전체 이미지를 만듦
  - 하나의 이미지에서도 여러개의 candidate가 나올 수 있고 candidat간에도 겹치는 부분이 있을 수 있음
  - 이러한 부분을 모두 모아서 aggregate를 수행
4. 추가적으로 FCN과 결합시에 성능이 좀 더 좋아짐
5. 4에 CRF를 적용시에 성능이 좀 더 좋아짐

# 5. Inference



## • (1) Aggregating Instance-wise Segmentation Maps

### ✓ Inference 과정

1. Proposal 과정에서 배경만 포함되거나 object가 잘리는 문제들이 발생
  - 해당 문제는 Proposals를 Aggregation 하면서 해결
  - Pixel에 대해 Average or Maximum 값을 취함
2. i번째 proposals에 대해서  $W * H * C$ 의 output score maps를 생성
  - $W * H$ 는 너비와 높이를 의미하고  $C$ 는 클래스의 수를 의미
  - Crop했기에 이미지의 남은 부분은 Zero-padding을 적용
  - 이후 하나의 이미지에 대해 발생한 proposals들의 Max, Mean Aggregation을 취함
  - 마지막으로 CFR를 적용

$$P(x, y, c) = \max_i G_i(x, y, c), \quad \forall i,$$

$$P(x, y, c) = \sum_i G_i(x, y, c), \quad \forall i.$$

# 5. Inference

## • (2) Ensemble with FCN

### ✓ FCN vs DeconvNet

1. DeconvNet은 fine-details를 잘 잡는 반면에 FCN은 전반적인 모양을 잘 잡음
2. Instance-wise prediction은 다양한 scales의 object 모양을 잡는데는 도움을 줌
3. Semantic Segmentation은 context 정보를 같이 잡는다는 장점이 있음
4. DeconvNet의 장점인 2와 FCN의 장점인 3을 결합시에 성능향상이 큰 폭으로 됨
  - 독립적으로 모델을 시행한 후에 둘의 결과를 평균내고 CRF를 적용

# 6. Experiments

## • (1) Implementation Details

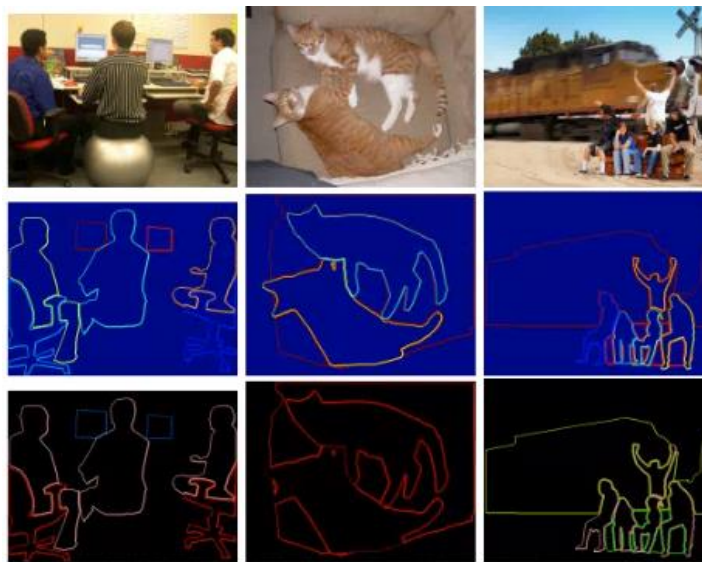
### ✓ Dataset / Optimization

[Dataset]

- PASCAL VOC 2012
- Augmented annotation[5]

[Optimization]

- SGD
  - lr : 0.01
  - momentum : 0.9
  - weight decay : 0.0005
- VGG 16-layer의 Pretrained된 모델을 사용 (ILSVRC dataset)
- dropout을 제거하고 Batch normalization 사용
- ReducelronPlateau scheduler 사용 (validation accuracy 기준)



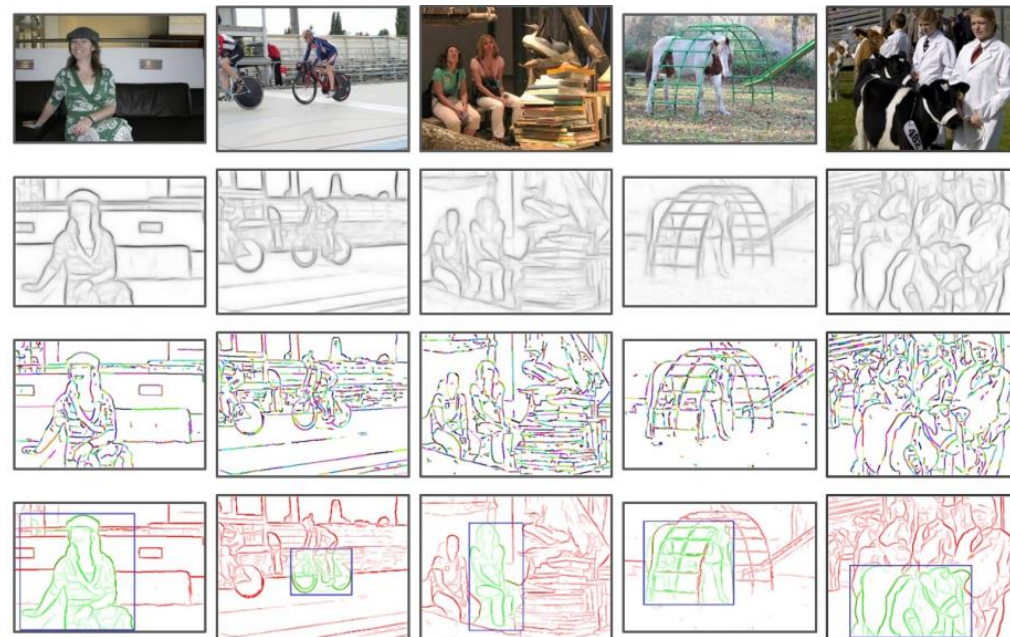
# 6. Experiments

## (1) Implementation Details

### ✓ Training Data Construction / Inference

#### [Training Data Construction]

- 1 stage training
  - 학습 데이터에 tight한 bounding box를 그린 후 local context가 추가로 포함되도록 1.2배 확장
  - Bounding box에서 square window를 crop
  - 중심에 위치한 object는 class label을 그렇지 않은 pixel은 background label을 제공
- 2 stage training
  - [6]에서 제안한 방법으로 object proposal을 생성
  - class labels는 annotation된거를 사용
  - 1 stage에서 사용한 post processing을 적용
  - 중복되는 proposals를 넣어서 class balancing을 맞춤





# 6. Experiments

## • (1) Implementation Details



### ✓ Inference 과정

1. edge-box [6]을 이용해서 test image 각각 2000개의 object proposals를 생성
2. objectness scores를 기반으로 상위 50개의 proposals만 추출
3. inference을 진행한 후에 pixel-wise maximum or mean을 계산

# 6. Experiments

## • (2) Evaluation on Pascal VOC

### ✓ Pascal VOC 2012

Table 1. Evaluation results on PASCAL VOC 2012 test set. (Asterisk (\*) denotes the algorithms that also use Microsoft COCO for training.)

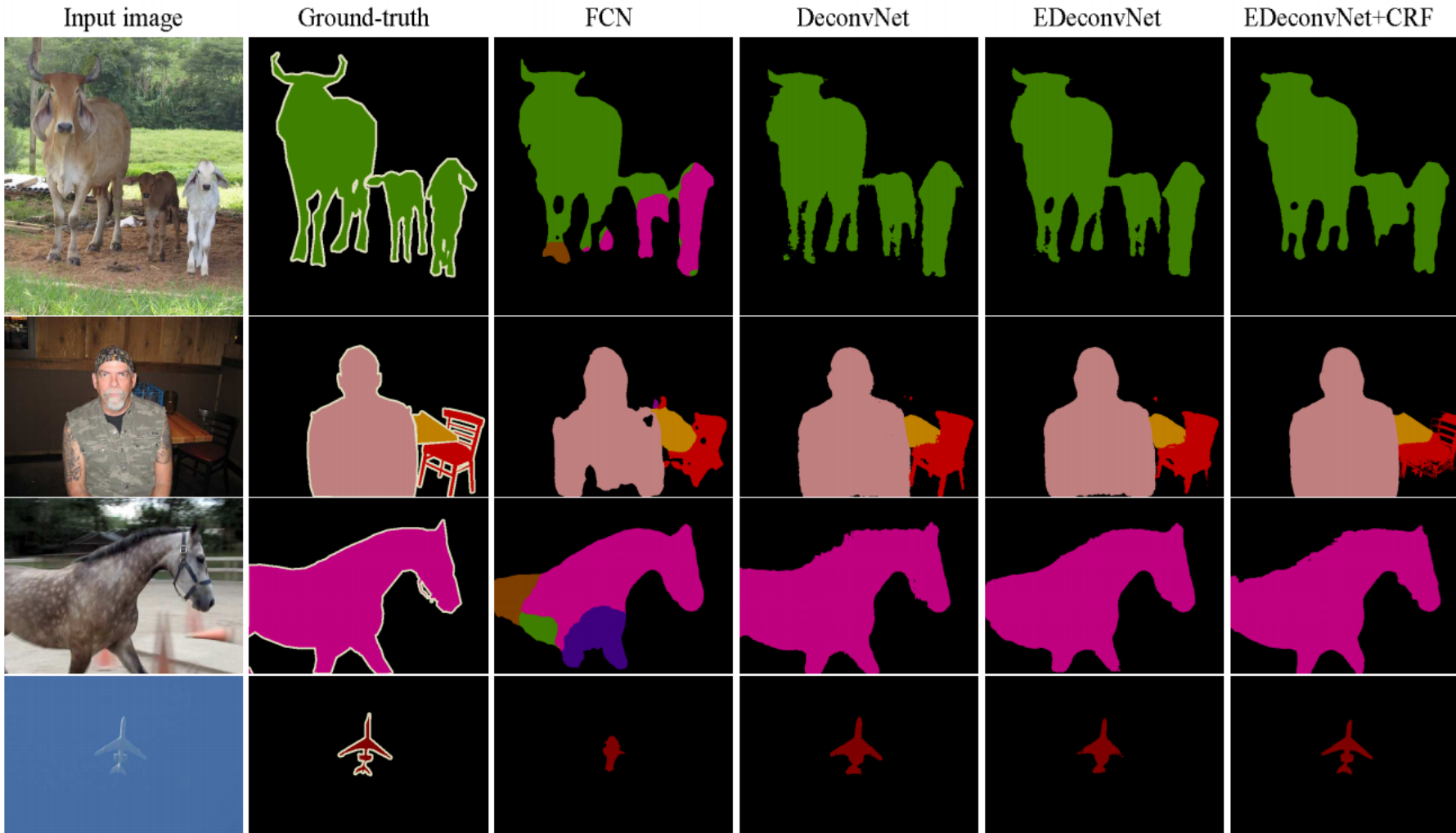
Method	bkg	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
Hypercolumn [11]	88.9	68.4	27.2	68.2	47.6	61.7	76.9	72.1	71.1	24.3	59.3	44.8	62.7	59.4	73.5	70.6	52.0	63.0	38.1	60.0	54.1	59.2
MSRA-CFM [3]	87.7	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN8s [19]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
TTI-Zoomout-16 [20]	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [1]	93.1	84.4	<b>54.5</b>	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	<b>59.7</b>	82.2	50.4	73.1	63.7	71.6
DeconvNet	92.7	85.9	42.6	78.9	62.5	66.6	87.4	77.8	79.5	26.3	73.4	60.2	70.8	76.5	79.6	77.7	58.2	77.4	52.9	75.2	59.8	69.6
DeconvNet+CRF	92.9	87.8	41.9	80.6	63.9	67.3	<b>88.1</b>	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	<b>55.0</b>	75.2	61.5	70.5
EDeconvNet	92.9	88.4	39.7	79.0	63.0	67.7	87.1	<b>81.5</b>	84.4	27.8	76.1	61.2	78.0	79.3	83.1	79.3	58.0	82.5	52.3	80.1	64.0	71.7
EDeconvNet+CRF	93.1	<b>89.9</b>	39.3	79.7	63.9	<b>68.2</b>	87.4	81.2	<b>86.1</b>	28.5	<b>77.0</b>	62.0	79.0	<b>80.3</b>	<b>83.6</b>	80.2	58.8	<b>83.4</b>	54.3	<b>80.7</b>	65.0	<b>72.5</b>
* WSSL [21]	93.2	85.3	36.2	<b>84.8</b>	61.2	67.5	84.7	81.4	81.0	<b>30.8</b>	73.8	53.8	77.5	76.5	82.3	<b>81.6</b>	56.3	78.9	52.3	76.6	63.3	70.4
* BoxSup [2]	<b>93.6</b>	86.4	35.5	79.7	<b>65.2</b>	65.2	84.3	78.5	83.7	30.5	76.2	<b>62.6</b>	<b>79.3</b>	76.1	82.1	81.3	57.0	78.2	<b>55.0</b>	72.5	<b>68.1</b>	71.0

1. DeconvNet 자체의 IoU가 69.6으로 FCN비해 많은 상승이 있었지만, DeepLab에 비해서는 낮음
2. CRF와 결합시에 1%p의 상승을 보임
3. FCN과 결합시에 2%p의 상승을 보임 (SOTA)

# 6. Experiments

- (2) Evaluation on Pascal VOC

## ✓ Pascal VOC 2012



(a) Examples that our method produces better results than FCN [19].

# 6. Experiments

## • (2) Evaluation on Pascal VOC

### ✓ Pascal VOC 2012

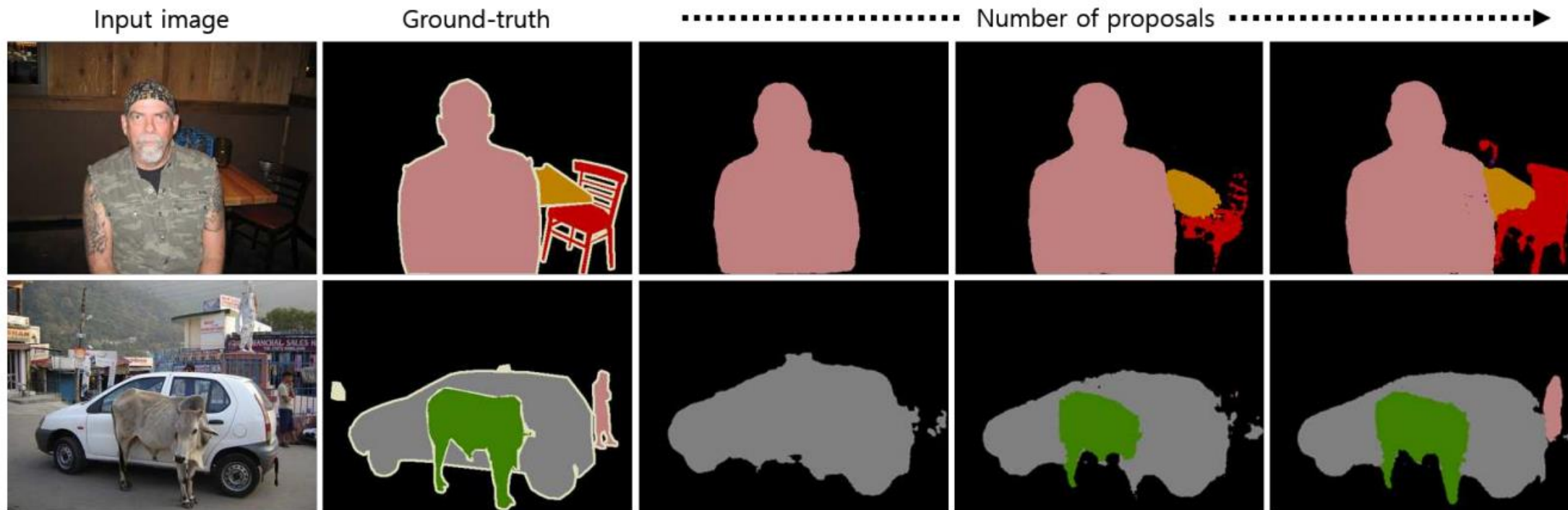


Figure 6. Benefit of instance-wise prediction. We aggregate the proposals in a decreasing order of their sizes. The algorithm identifies finer object structures through iterations by handling multi-scale objects effectively.

# 7. Conclusion



- (1) Overview

1. Deep Deconvolution network를 통해서 novel semantic segmentation을 제안
2. Dense하고 precise한 object segmentation을 수행
3. FCN이 가지는 fixed size receptive field라는 한계를 instance wise segmentation을 통해서 해결
4. FCN과 Ensemble을 통해서 더 높은 성능을 보였음
  - FCN과 DeconvNet의 성질이 다른 것을 통합
5. PASCAL VOC 2012에서 SOTA 달성

# 7. Conclusion

## • (2) Advantages

1. DeconvNet의 기여를 표현하려고 이끌어 가는 과정이 탄탄함
  - FCN의 단점을 지적하고 DeconvNet에서 해결하는 과정
  - DeconvNet의 해결책 (Unpool, Transposed Conv 등)이 왜 좋은지
    - Unpool / Transposed Conv의 장단점을 비교하고 시각화하는 과정
    - Detail한 부분이 어떻게 살아나는지
2. Instance wise segmentation을 같이 수행할 수 있는 장점
3. Multi-scale이 가지는 문제점을 Object proposal을 통해서 해결하려고 시도

# 7. Conclusion

## • (3) Disadvantages

1. Receptive Field의 문제를 해결하려고 도입한 방식이 너무 비용이 많이 드는 방법
  - Proposal을 통해서 하나의 이미지당 2000개의 후보군을 추출
  - 추출한 후보군에 대해서 상위 50개를 선택
  - 50개에 대해서 Inference를 수행후 Aggregation을 진행
  - 학습 속도에 대해서도 언급이 안되어있음
2. 학습과 추론과정이 복잡함. 사용자가 적용하기에 불편한 점이 많음
  - FCN의 경우 End-to-End 방식임
  - 1 stage 학습 + 2 stage 학습뿐만 아니라 Proposal 추출까지 필요
3. DeconvNet 단일 성능이 뛰어난 것 같지는 않음
  - DeepLab + CRF와 비교시에 오히려 낮음
  - DeepLab + CRF의 결과 이미지도 함께 보여줬으면 더 의미있지 않았을까 생각
4. 실험의 수가 다른 논문에 대비해서 많이 부족함
  - PASCAL VOC를 보통 많이 벤치마크하기는 하지만 여러 데이터에 대해서 실험을 해서 결과 비교가 필요해 보임



# 8. Appendix



## (1) 참고자료

1. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015
2. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012
3. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014
4. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014
5. B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. ´ Semantic contours from inverse detectors. In ICCV, 2011
6. C.L. Zitnick and P.Dollar Edge boxes: Locating object proposals from edges. In ECCV, 2014  
[http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016\\_slide15.pdf](http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016_slide15.pdf)
7. [http://vision.stanford.edu/teaching/cs231b\\_spring1213/slides/segmentation.pdf](http://vision.stanford.edu/teaching/cs231b_spring1213/slides/segmentation.pdf)
8. <https://hci.iwr.uni-heidelberg.de/vislearn/wp-content/uploads/2016/06/GM-lecture3.pdf>
9. <https://www.youtube.com/watch?v=58fuWVu5DVU>

# 8. Appendix

## (2) Superpixel

Superpixel 이란?

- 지각적으로 의미있는 픽셀들을 모아서 그룹화해준 것
- 즉, 비슷한 부위를 모아서 그룹화한 픽셀 그룹을 Superpixel이라고 표현함



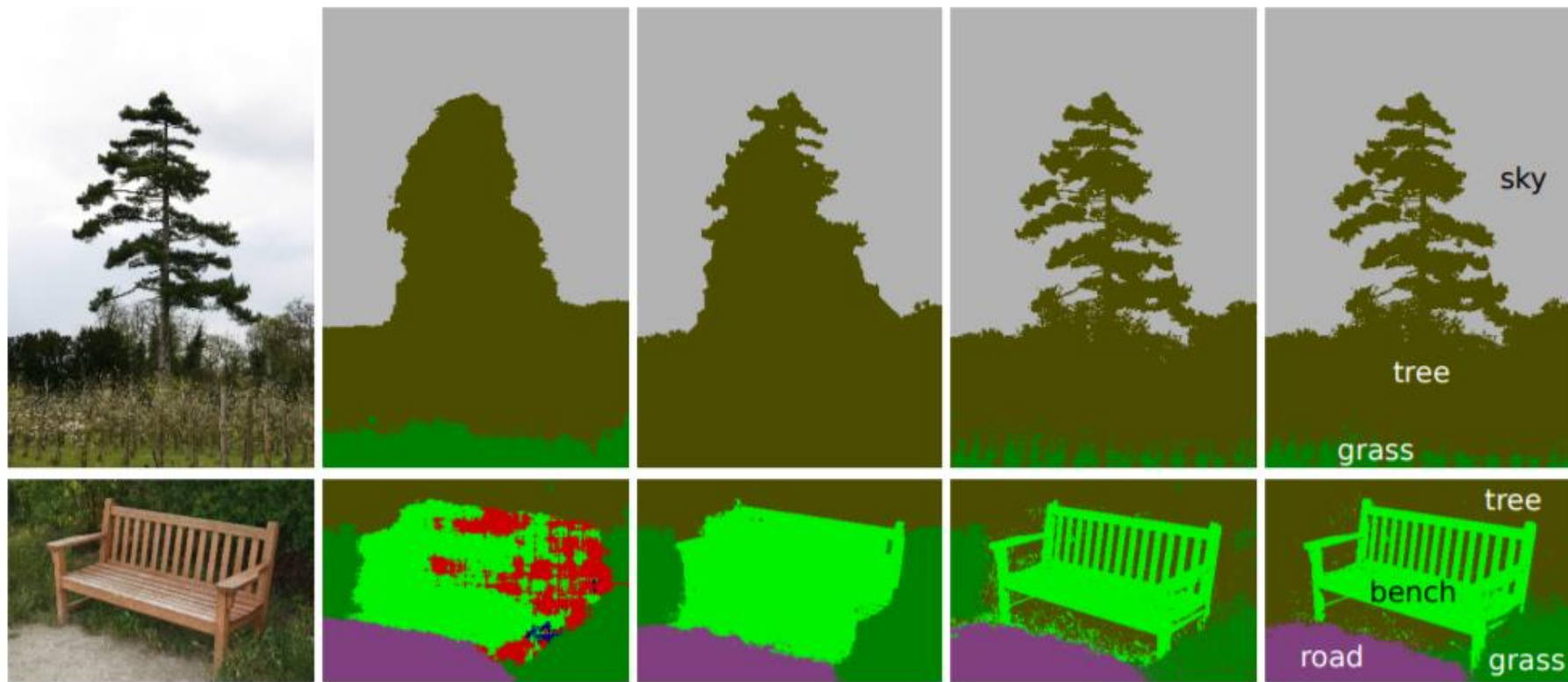
이미지 출처 : SLIC 오리지널 논문

# 8. Appendix

## • (3) Conditional Random Field (CRF)

Conditional Random Field 이란?

- Conv + Pooling을 통해서 사라진 Detail한 정보를 복원하기 위함 (이미지의 노이즈를 제거 및 경계면의 특성을 활성화)
  - FCN에서는 Skip Connection을 통해서 위의 문제를 해결
  - DeepLab v1에서는 pooling Layer을 제거하고 Atrous Convolution을 사용



(a) Image

(b) Unary classifiers

(c) Robust  $P^n$  CRF

(d) Fully connected CRF, MCMC inference, 36 hrs

(e) Fully connected CRF, our approach, 0.2 seconds

출처 : P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In NIPS, 2011

출처 : Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Chen et al. ICLR'15

# 8. Appendix

## • (3) Conditional Random Field (CRF)

Conditional Random Field 이란?

- CRF에 대한 개념 ([http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016\\_slide15.pdf](http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016_slide15.pdf))

$$E(x|I) = \sum_i \phi_u(x_i|I) + \sum_i \sum_{j \in \partial i} \phi_p(x_i, x_j|I)$$

Unary term

Pairwise term

# 8. Appendix

## • (4) Batch Normalization

Batch Normalization 이란?





# 8. 이해안가는 목록



1. Semantic Segmentation based on FCNs have a couple of critical limitations. First, the network has a predefined fixed-size receptive field (여기서 **fixed size receptive field**가 의미하는게 어떤 것). Therefore, the object that is substantially larger or smaller than the receptive field may be fragmented or mislabeled. (**fixed size에 의해서 어떤식으로 모델이 영향을 주는지에 대해서 정확한 이해가 필요**)
2. It is free from scale issues found in the original FCN-based methods and identifies finer details of an objects. (Object Proposals 방식으로 instance-wise segmentation하는게 과연 좋은 솔루션일까? 이를 통해서 scale issue를 해결했다는데 맞는지 잘 모르겠음)  
[28] C.L. Zitnick and P.Dollar Edge boxes: Locating object proposals from edges. In ECCV, 2014
3. 6.1. Implementation Details에 대한 내용을 정확히 확인.
  1. Tight한 bounding 박스를 구하기위해 Region Proposals를 추출한 방식(square window tightly enclosing ... )에 대해 잘 모르겠음 (edge-box [28] 언급이 한 줄 있기는 함)
  2. Object located at the center while other pixels are labeled as background ( 원하는 object만 class주고 나머지는 배경으로..? 이게 맞나?)
  3. Second에서는 first와는 다르게, all relevant class labels are used for annotation이라는데 이게 위와 맞물려서 배경도 20개 클래스 속하면 라벨링했다는 의미인지?
  4. Balance를 유지하기 위해서 redundant example을 넣었다는데 이게 무슨 의미이고, with limited number of examples가 의미하는바가 무엇인지? 클래스가 적은 애들때매 뭘 한거지..?
  5. 이렇게 학습을 한게 Fixed size receptive field를 어떻게 해결한지 모르겠음. 잘라도 크기는 동일한텐데, 이 후에 Resize를 한건지 ... ? 아니면 그냥 zero padding만 해서 동일하게 만든지?
4. Predefined fixed size receptive field 에 대해서 FCN의 한계 부분이 정확히 이해안하고 이게 DeconvNet에 어떻게 이어지는지
5. Conditional Random Field의 정확한 개념

감사합니다