



# [논문 리뷰] SegNet [2017]

☀ Pixel-wise Segmentation Model인 Dep fully Convolutional Net **SegNet**을 제시합니다.

Pixel-wise classification layer에 기반하여, Encoder와 그에 상응하는 Decoder로 구성되어 있습니다.

Architecture의 Encoder는 VGG16의 13개의 layer와 동일하게 구성되어 있습니다.

Decoder는 low Resolution Feature Map을 Input Resolution feature에 Pixel-wise classification로 매핑하는 것입니다.

특히, Decoder는 non-linear Upsampling을 위해 Encoder에서 계산된 Max-pooling indices를 활용합니다. 이러한 과정은 학습에 필요 없는 요소를 Upsample을 위해 제거합니다.

자율주행과 관련된 구조들을 위해 설계된 모델이다. Upsampled maps는 드물고, 조밀한 Feature Map을 생성하기 위해 trainable filter와 결합합니다.

Memory와 Accuracy를 trade-off를 하였을 때 Segmentation Task에서 좋은 성과를 낼 수 있었습니다.

**SegNet**은 application응용을 위해 만든 것이기에, Computational 부분에서 좋은 성능을 가져다 줍니다.

또한, 상당한 추론 시간과 메모리 측면에서 다른 Architecture에 비해 좋은 성능을 낸다는 것입니다.

## Introduction

Semantic Segmentation은 장면 이해, 자율 주행, 지원 관계 등 여러 분야에서 응용되어 사용되고 있습니다.

Segmentation에서 Max Pooling과 Sub Sampling 연산을 수행하다보면, coarse feature map이 생성되는데, 이는 Pixel-wise prediction을 해야하므로, 좋은 Output을 내지 못합니다.

**SegNet**은 low Resolution feature로부터, Input 이미지와 동일한 크기로 정확한 boundary localization이 가능한 Architecture를 만들고자 제안하였습니다.

SegNet의 Encoder network는 VGG16의 Convolutional Layer로 구성되어 있습니다. **SegNet의 Key Point는 Encoder에 상응하여, 만들어진 Decoder**입니다. Encoder에서 만든 input feature map의 비선형적 upsampling을 Decoder의 Max-Pooling에서 사용합니다.

이러한 방식은 3가지 이점을 가져왔습니다.

1. 객체들간의 경계선을 강화합니다.
2. End-to-end training이 가능하기에, Parameter의 수를 대폭 감소시킵니다.
3. UpSampling은 모든 Encoder-Decoder Architecture에서 약간의 수정을 통해서 활용될 수 있습니다.

SegNet에서 FC layer를 제외한 이유는, **SegNet이 메모리 사용량 및 계산량등에서 Computational Cost측면에서 강점**을 가지기 위해서 FC layer를 제외하였다.

Decoder의 경우 Encoder의 mirrored 구조로 볼 수 있으며, Upsampling을 위해, Un-Maxpooling을 사용하였다.

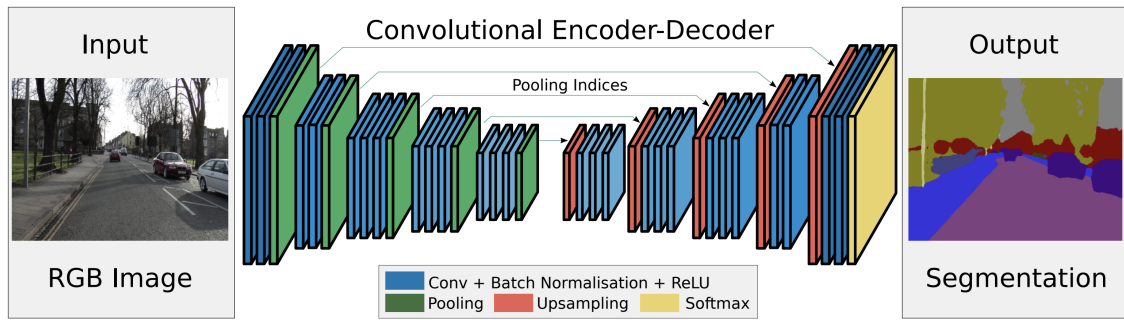


Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.

## Architecture

SegNet은 final pixelwise classification layer를 따르는 Encoder와 이에 상응하는 Decoder를 가지고 있습니다. 또한 VGG16으로 이루어져, large Dataset으로부터, pre-train하는 것이 가능합니다.

(13개의 layer로 활용하여, Encoder를 만들기에는, Decoder도 13개의 layer로 이루어져 있습니다.)

또한 가장 높은 해상도에서 Feature map을 출력하기 위해, 이전의 Fully Connected layer를 사용하지 않습니다. 최종 decoder Output은 Multi-class Softmax와 대입 후, 각각의 pixel에 대해, class를 생성합니다.

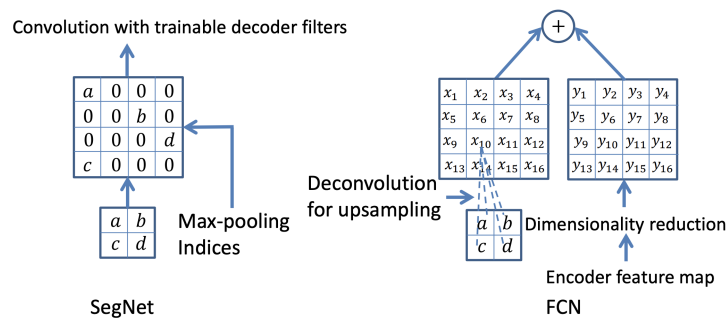


Fig. 3. An illustration of SegNet and FCN [2] decoders.  $a, b, c, d$  correspond to values in a feature map. SegNet uses the max pooling indices to upsample (without learning) the feature map(s) and convolves with a trainable decoder filter bank. FCN upsamples by learning to deconvolve the input feature map and adds the corresponding encoder feature map to produce the decoder output. This feature map is the output of the max-pooling layer (includes sub-sampling) in the corresponding encoder. Note that there are no trainable decoder filters in FCN.

**Encoder Network** : VGG16에서의 Fully Connected layer를 뺀 Convolutional Layer를 사용합니다.

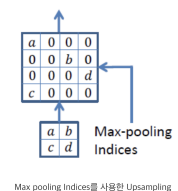
2x2 Max Pooling을 수행하면서, Max Pooling Indices(위치 정보)를 저장합니다.

### Decoder Network

Upsampling과정에서 저장했던 Pooling indices를 활용하게 됩니다. 그림을 보면, Max Pooling과정에서 선택했던 index들의 위치로 값들이 이동하는 것을 확인할 수 있습니다.

이러한 과정을 거치면 0이 많은 Sparse feature map이 만들어지게 되는데, trainable decoder filter를 활용하여, 빈 부분 채우는 작업을 시행합니다.

- Upsampling을 수행하여, 각 픽셀의 class 예측을 위한 softmax classifier가 존재한다.
- Un-Maxpooling과 Convolution을 활용합니다.
- Upsampling은 Encoder에서 저장한 Max Pooling indices를 활용합니다.



# Differences

## FCN

FCN과의 차이점은 FCN은 UpSampling과정에서 trainable decoder filter를 적용하지만, SegNet은 **UpSampling Max pooling indices**를 추가적으로 학습 없이, 만들어진 **feature map**에 **decoder filter**를 추가하는 방법을 선택합니다. 또한, FCN은 Encoder feature map에 대한 내용을 저장하고 있어, memory cost가 많이 발생하지만, SegNet은 **Max Pooling indices**만 저장하기에, **Memory cost**가 적습니다.

## DeconvNe

## U-Net

- Fully connected layer의 사용 유무 차이
- Pooling indices를 사용 유무 차이

# Experiments

☀ 기존 접근법과 비교했을 때, SegNet이 가장 좋은 성능을 낼 수 있음을 알 수 있습니다.

- Encoder feature map를 Decoder에 넣어주는 방법이 가장 좋은 성능을 나타냅니다.
- 메모리가 한정되어 있다면, Max Pooling등을 통해, Input map을 압축할 수 있고, decoder와 함께 사용하여, 성능을 향상시킬 수 있습니다.

Method	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Side-walk	Bicyclist	Class avg.	Global avg.	mIoU	BF
SfM+Appearance [28]	46.2	61.9	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1	n/a*	
Boosting [29]	61.9	67.3	91.1	71.1	58.5	92.9	49.5	37.6	25.8	77.8	24.7	59.8	76.4	n/a*	
Dense Depth Maps [32]	85.3	57.3	95.4	69.2	46.5	<b>98.5</b>	23.8	44.3	22.0	38.1	28.7	55.4	82.1	n/a*	
Structured Random Forests [31]						n/a						51.4	72.5	n/a*	
Neural Decision Forests [64]						n/a						56.1	82.1	n/a*	
Local Label Descriptors [65]	80.7	61.5	88.8	16.4	n/a	98.0	1.09	0.05	4.13	12.4	0.07	36.3	73.6	n/a*	
Super Parsing [33]	87.0	67.1	96.9	62.7	30.1	95.9	14.7	17.9	1.7	70.0	19.4	51.2	83.3	n/a*	
SegNet (3.5K dataset training - 140K)	<b>89.6</b>	<b>83.4</b>	96.1	<b>87.7</b>	52.7	96.4	<b>62.2</b>	<b>53.45</b>	<b>32.1</b>	<b>93.3</b>	<b>36.5</b>	<b>71.20</b>	<b>90.40</b>	60.10	46.84
CRF based approaches															
Boosting + pairwise CRF [29]	70.7	70.8	94.7	74.4	55.9	94.1	45.7	37.2	13.0	79.3	23.1	59.9	79.8	n/a*	
Boosting+Higher order [29]	84.5	72.6	<b>97.5</b>	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5	59.2	83.8	n/a*	
Boosting+Detectors+CRF [30]	81.5	76.6	96.2	78.7	40.2	93.9	43.0	47.6	14.3	81.5	33.9	62.5	83.8	n/a*	

TABLE 2

Quantitative comparisons of SegNet with traditional methods on the CamVid 11 road class segmentation problem [22]. SegNet outperforms all the other methods, including those using depth, video and/or CRF's on the majority of classes. In comparison with the CRF based methods SegNet predictions are more accurate in 8 out of the 11 classes. It also shows a good  $\approx 10\%$  improvement in class average accuracy when trained on a large dataset of 3.5K images. Particularly noteworthy are the significant improvements in accuracy for the smaller/thinner classes. \* Note that we could not access predictions for older methods for computing the mIoU, BF metrics.

Network/Iterations	40K				80K				>80K				Max iter
	G	C	mIoU	BF	G	C	mIoU	BF	G	C	mIoU	BF	
SegNet	88.81	59.93	50.02	35.78	89.68	69.82	57.18	42.08	90.40	71.20	60.10	46.84	140K
DeepLab-LargeFOV [3]	85.95	60.41	50.18	26.25	87.76	62.57	53.34	32.04	88.20	62.53	53.88	32.77	140K
DeepLab-LargeFOV-denseCRF [3]	not computed								89.71	60.67	54.74	40.79	140K
FCN	81.97	54.38	46.59	22.86	82.71	56.22	47.95	24.76	83.27	59.56	49.83	27.99	200K
FCN (learned deconv) [2]	83.21	56.05	48.68	27.40	83.71	59.64	50.80	31.01	83.14	64.21	51.96	33.18	160K
DeconvNet [4]	85.26	46.40	39.69	27.36	85.19	54.08	43.74	29.33	89.58	70.24	59.77	52.23	260K

TABLE 3

Quantitative comparison of deep networks for semantic segmentation on the CamVid test set when trained on a corpus of 3433 road scenes *without class balancing*. When end-to-end training is performed with the same and fixed learning rate, smaller networks like SegNet learn to perform better in a shorter time. The BF score which measures the accuracy of inter-class boundary delineation is significantly higher for SegNet, DeconvNet as compared to other competing models. DeconvNet matches the metrics for SegNet but at a much larger computational cost. Also see Table 2 for individual class accuracies for SegNet.

# Conclusion

SegNet이 연구되기까지의 동기는, road, indoor등 장면이해를 위한 효율적인 Architecture에 대한 이해가 필요하기 때문이다. SegNet은 Computational time과 Memory의 측면에서 효율적이었다.

다른 Model들과 SegNet을 비교하였을 때, **Accuracy, Memory, training time**면에서 **trade-off**시, 매우 실용적인이라는 것을 증명하였습니다.

**SegNet**은 **feature map**의 **max-pooling**하여, 효율적으로 활용하였으며, **decoder architecture**에서 사용 시, 좋은 성능을 나타내었습니다.


#### SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

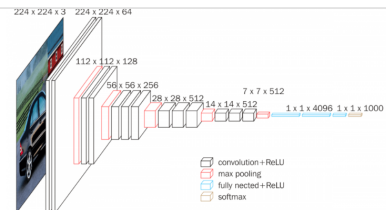
We present a novel and practical deep fully convolutional neural network architecture for semantic pixel-wise segmentation termed SegNet. This core trainable segmentation engine consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer.

 <https://arxiv.org/abs/1511.00561>




#### [Semantic Segmentation] SegNet 원리

논문 : <https://arxiv.org/pdf/1511.00561.pdf> 1. Semantic Segmentation의 목적 : 2. Semantic Segmentation 알고리즘 - SegNet <https://kuklife.tistory.com/118?category=872136> SegNet 논문은 2016년 10월 경, Vijay Badrinarayanan, Alex Kendall,  <https://kuklife.tistory.com/120>



#### All about Segmentation

Information Title: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, TPAMI 2017 Reference Review By: Junmyeong Lee (가짜연구소 논문미식회 2기) Edited by: Taeyup Song Last updated on Jan. 16, 2022 Contribution VGG-16 구조 VGG16과 위상적으로 동일함(같은 구조를 가짐) → 따라서, pre-trained baseline을 불러올 수 있고, 더욱 좋은 성능을 낼 수 있었음.

 [https://pseudo-lab.github.io/SegCrew-Book/docs/ch1/01\\_02\\_02\\_segnet.html](https://pseudo-lab.github.io/SegCrew-Book/docs/ch1/01_02_02_segnet.html)