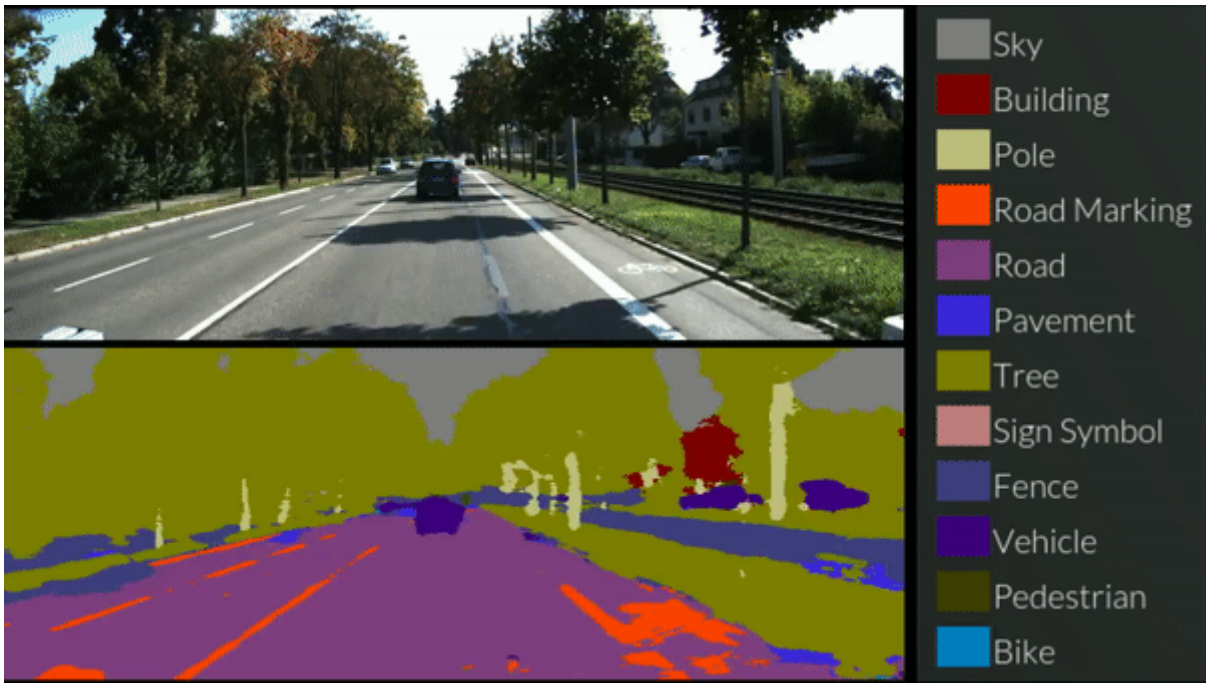
SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

Introduction

Q. 차 앞에 노인과 아이가 있다. 당신은 뭘 밟을 것인가?

- SegNet은 도로를 달리면서 촬영한 영상(road scene)에 대해 pixel-wise semantic segmentation 하기 위해 설계된 모델 (2016)
  - Segnet은 자율주행에 큰 역할을 한 모델
  - 도로와 보도는 보기에 비슷해보이지만 두 class간의 경계를 잘 구분하는 것이 매우 중요

    → 자율주행에서 도로와 보도를 구분하지 못하면 정말 큰 사고!



Appearance Background

- 기존의 sementic segmentation 모델들의 해상도가 매우 떨어지는 문제

- Max pooling, sub-sampling 연산을 수행하다보면 coarse(알맹이가 큰) feature map이 만들어지게 되는데, 이런 feature map으로는 픽셀 단위로 정교하게 segmentation을 할 수 없음
  - FCN에서 이를 해결하기 위해 Skip architecture 등 다양한 해결 방법을 시도했으나, 정교하지 못했음

- 자율주행을 위해서는 실시간으로 빠르게 segmentation이 가능해야 하는데, 정확도가 높다하더라도 계산량이 많거나 parameter 수가 많으면 빠르게 segmentation을 할 수 없음
  - memory 및 inference time 측면에서 효율적으로 동작할 수 있어야 함

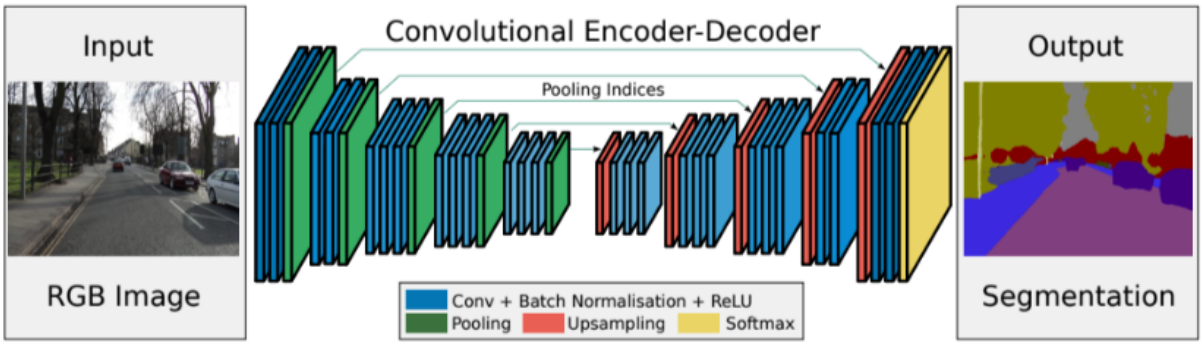
→ 이런 문제들을 해결하기 위해 등장한 모델이 SegNet

- Appearance(도로, 건물), shape(자동차, 보행자)를 잘 인식하고 다른 class간의 spatial-relationship(context)를 이해할 수 있어야 함

- Encoder에서 추출된 image representation으로부터 boundary information을 유지하는 것 중요
  - 대부분 pixel을 차지하는 도로나 건물 같은 큰 object들에 대해서는 smooth segmentation을 생성할 수 있어야 함
  - 보행자와 같은 작은 object에 대한 shape도 잘 나타낼 수 있어야 함

- 계산량 관점에서는 memory 및 inference time에서 효율적으로 동작할 수 있어야 함
  - 전체 네트워크의 학습 파라미터를 end-to-end로 한번에 학습
  - 빠르게 weight update를 반복하면서 수렴할 수 있는 SGD를 사용

Architecture



1) Encoder Network

- VGG16 구조에서 FC layer를 뺀 13개의 layer를 사용하여 연산량을 줄임
  - VGG에는 FC layer에 존재하지만, FC layer가 그리 중요한 역할을 하지 않음
  - FC layer를 제거함으로써 메모리 사용량이 줄어들고, 약 9배의 파라미터가 줄어듬

- Conv + Batch Norm + ReLU
- Max-Pooling
  - kernel size: 2x2 / stride: 2

2) Decoder Network

- 각 Encoder Network에 대응하여 존재
  - Encoder의 Pooling layer는 Up-Sampling layer에 대체
  - Up-Sampling layer 뒤에 Conv + Batch Norm + ReLU 연결

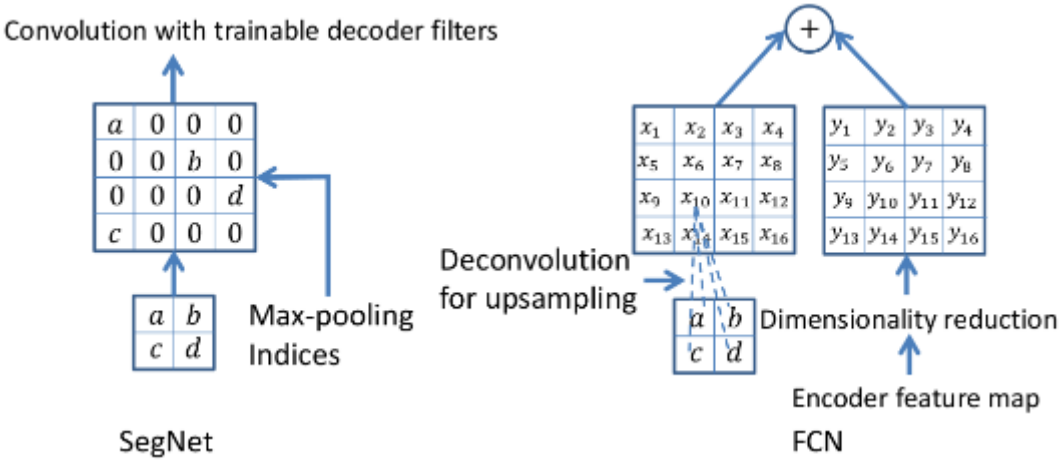
- Coarse feature map이 생기는 이유?
  - pooling 및 convolution 연산때문에 feature map의 정보가 소실되기 때문
  - Decoder에서 up-sampling을 할 때 Encoder의 feature map 정보를 decoder로 전달할 수 있다면 소실된 정보를 다시 찾는 것이므로 pixel 단위로 정교하게 segmentation 할 수 있을 것

- Encoder의 feature map 정보를 decoder로 전달하는 가장 정확한 방법

- 전체 feature map을 저장해 두었다가 Up-sampling 할 때 모두 Decoder로 전달하는 것
- 그러나 이 방식을 수행하기 위해서는 많은 메모리가 필요함

  → SegNet에서는 Max-Pooling indices(위치 정보)만을 저장해두었다가 이후 Max Unpooling

  → accuracy는 아주 조금 감소하나, memory는 크게 아낄 수 있음

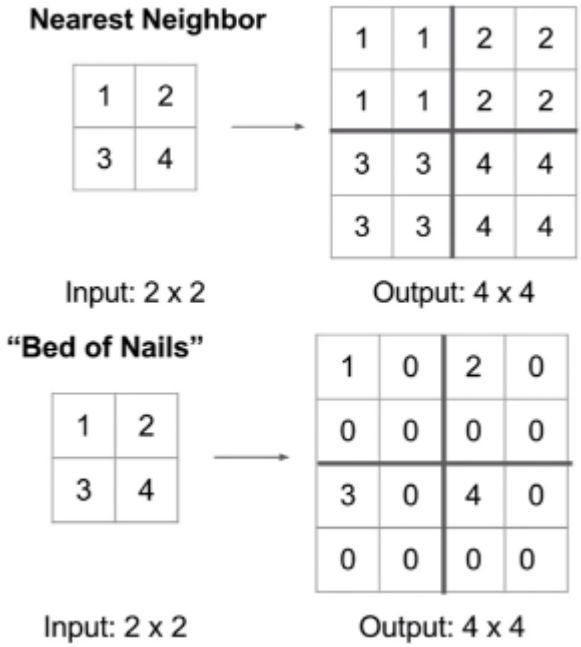


- SegNet: max-pooling indices 사용
- FCN: transposed convolution & dimensionality reduction(skip-architecture) 사용
- Max Unpooling 방식으로 Up-Sampling을 함으로써 얻는 장점
  - FCN에서는 Transposed convolution으로 upsampling을 진행했는데, SegNet은 Transposed convolution을 사용하지 않기 때문에 학습 파라미터가 없어 전체 parameter 개수를 줄일 수 있음
  - 전체 모델은 end-to-end 학습이 가능합니다.
  - 다른 encoder-decoder 형식에 응용될 수 있고, 변형도 가능

▼ **What is Max Unpooling?**

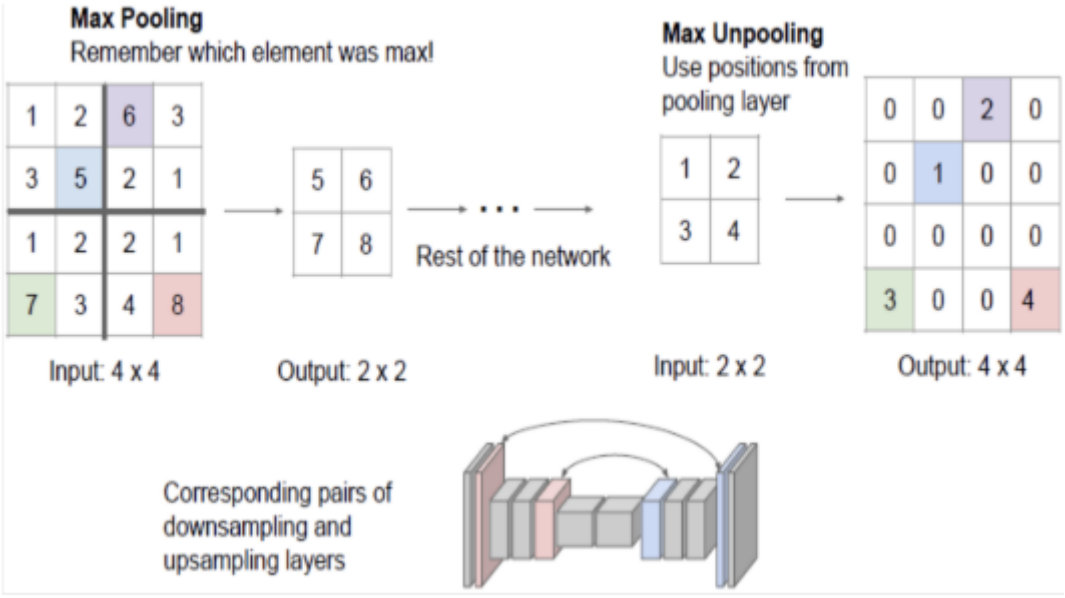
Unpooling

- Maxpooling을 거꾸로 재현하여 주변 픽셀들을 동일한 값으로 채우거나, 0으로 채우는 방식



Max Unpooling

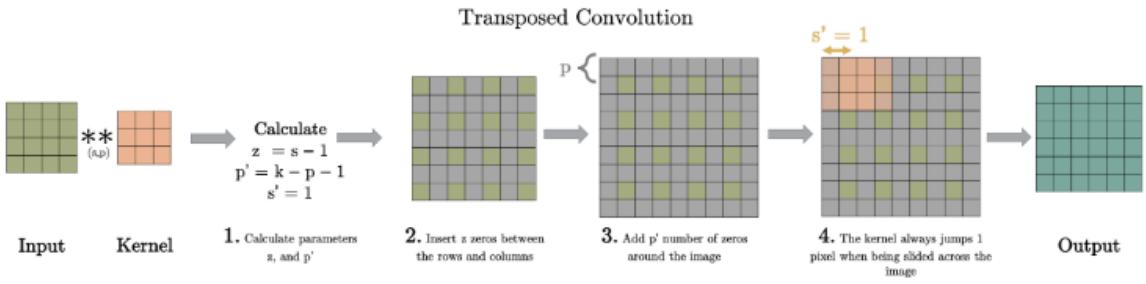
- Unpooling의 문제점
  - 원래 사이즈로 Unpooling 하게 되면 원래 Max pooled된 값의 위치를 알 수 없음
- Max Unpooling
  - Max pooling 할때의 선택된 값들의 위치를 기억해 원래 자료의 동일한 위치에 Max값을 위치시켜 Unpooling하는 기법

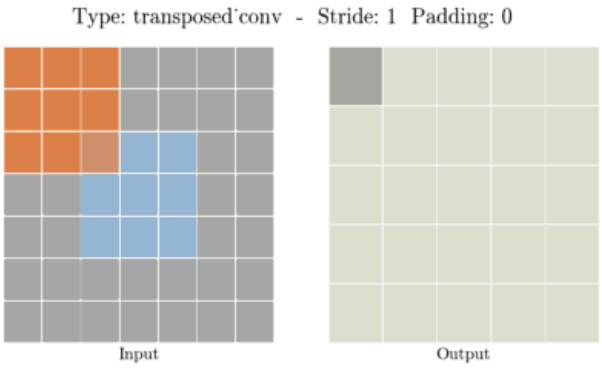


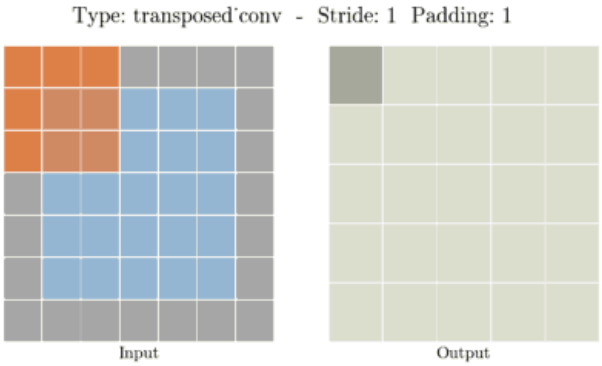
▼ **What is transposed convolution?**

Transposed Convolution

- upsamping을 실행 → input 보다 output의 공간적 차원이 더 큼
- Step 1 : 새로운 parameter z, 와 p'을 계산
  - $z=s-1$, $p'=k-p-1$, $s'=1$
- Step 2: input의 각 행과 열사이에 z 만큼의 0을 삽입
  - input의 사이즈를 $(2*i-1)$ x $(2*i-1)$만큼 증가시킴
- Step 3: 2에서 변형된 input에다 p'만큼의 0을 패딩해줌
- Step 4: 3까지 변형된 input에다 stride 1인 일반적인 convolution을 진행
- Output size: $o = (i-1)*s+k-2p$







Comparison					
Conv Type	Operation	Zero Insertions	Padding	Stride	Output Size
Standard	Downsampling	0	p	s	(i+2p-k)/s +1
Transposed	Upsampling	(s - 1)	(k-p-1)	1	(i-1)*s+k-2p

3) Softmax classifier

- Decoder의 output은 K-class softmax classifier로 들어가 최종적으로는 각 픽셀마다의 독립적인 확률값으로 계산됨
- 각 픽셀별로 가장 확률이 높은 class만을 출력하면 최종 segmentation

Performance

- 다른 Architecture(FCN, DeepLab-LargeFOV, DeconvNet 등)와 성능을 비교
- 성능을 2가지 scene segmentation benchmark에서 평가
  - Cam Vid dataset - road scene segmentation
  - SUN RGB-D dataset - indoor scene segmentation

- 평가척도
  - Global accuracy (G): 데이터셋 전체의 픽셀 수에서 올바르게 분류된 픽셀의 수
  - Class average accuracy (C): 각 클래스마다 accuracy를 계산한 뒤, 평균
  - mIoU
  - Boundary F1 Score (BF): 2 * precision * recall / (recall + precision)

Cam Vid dataset

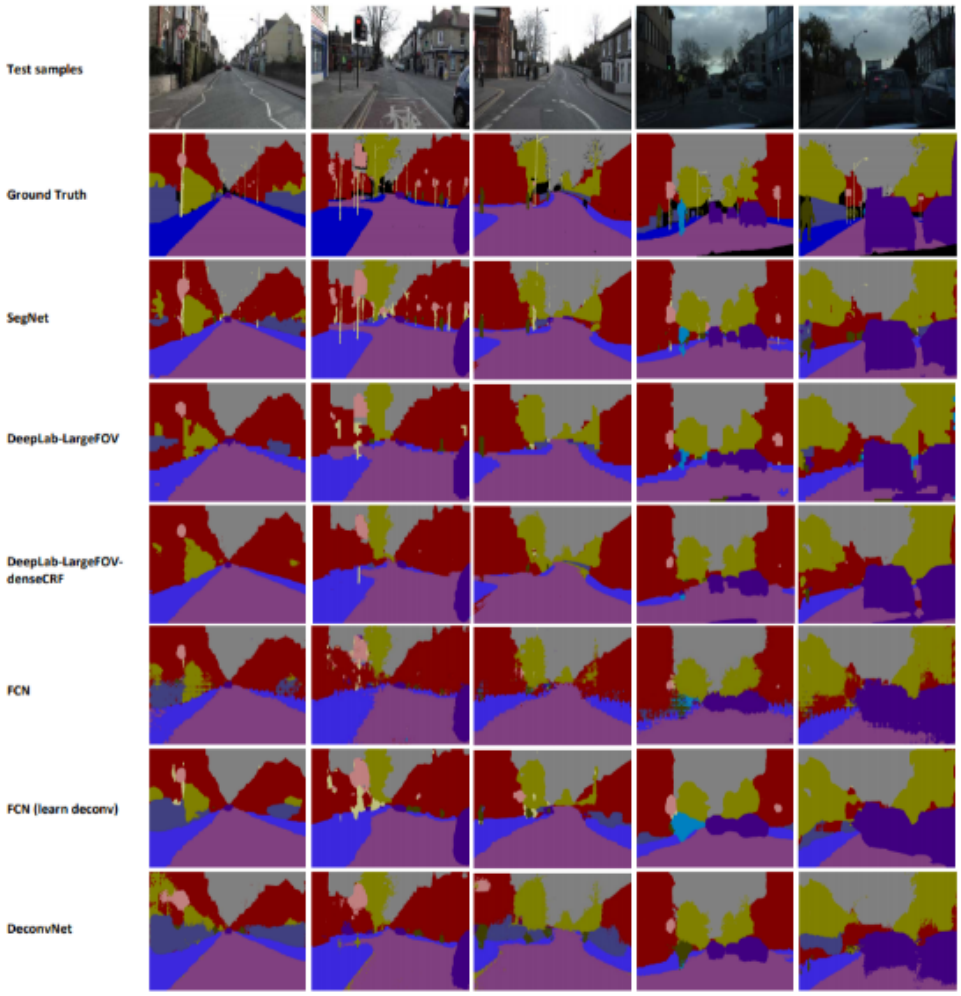


Fig. 4. Results on CamVid day and dusk test samples. SegNet shows superior performance, particularly with its ability to delineate boundaries, as compared to some of the larger models when all are trained in a controlled setting. DeepLab-LargeFOV is the most efficient model and with CRF post-processing can produce competitive results although smaller classes are lost. FCN with learnt deconvolution is clearly better. DeconvNet is the largest model with the longest training time, but its predictions loose small classes. Note that these results correspond to the model corresponding to the highest mIoU accuracy in Table 3.

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 (learnt 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.

- SegNet, DeconvNet이 가장 좋은 성능을 보임
- class간 boundary 부분에서 SegNet이 다른 모델들보다 더 정교하게 boundary를 구분하고 small 객체도 잘 잡음

- DeepLab-LargeFOV의 경우 크기가 작은 클래스는 제대로 segmentation하지 못했지만 그래도 competitive한 결과를 보여줌
- FCN with deconv의 경우 고정된 bilinear upsampling을 사용한 FCN보다 더 better한 결과를 보여줌
- DeconvNet의 경우 모델 자체가 굉장히 크고 학습이 비효율적이며 small class 객체는 segmentation하지 못함

SUN RGB-D dataset

- 실내 이미지에서 37개 클래스를 segmentation한 데이터셋
- 객체의 shape, size, pose가 굉장히 다양하고 부분적인 occlusion도 존재하기 때문에 꽤나 challenge한 task
- 본 논문에서는 이미지의 depth 정보는 제외하고 RGB 정보만 사용

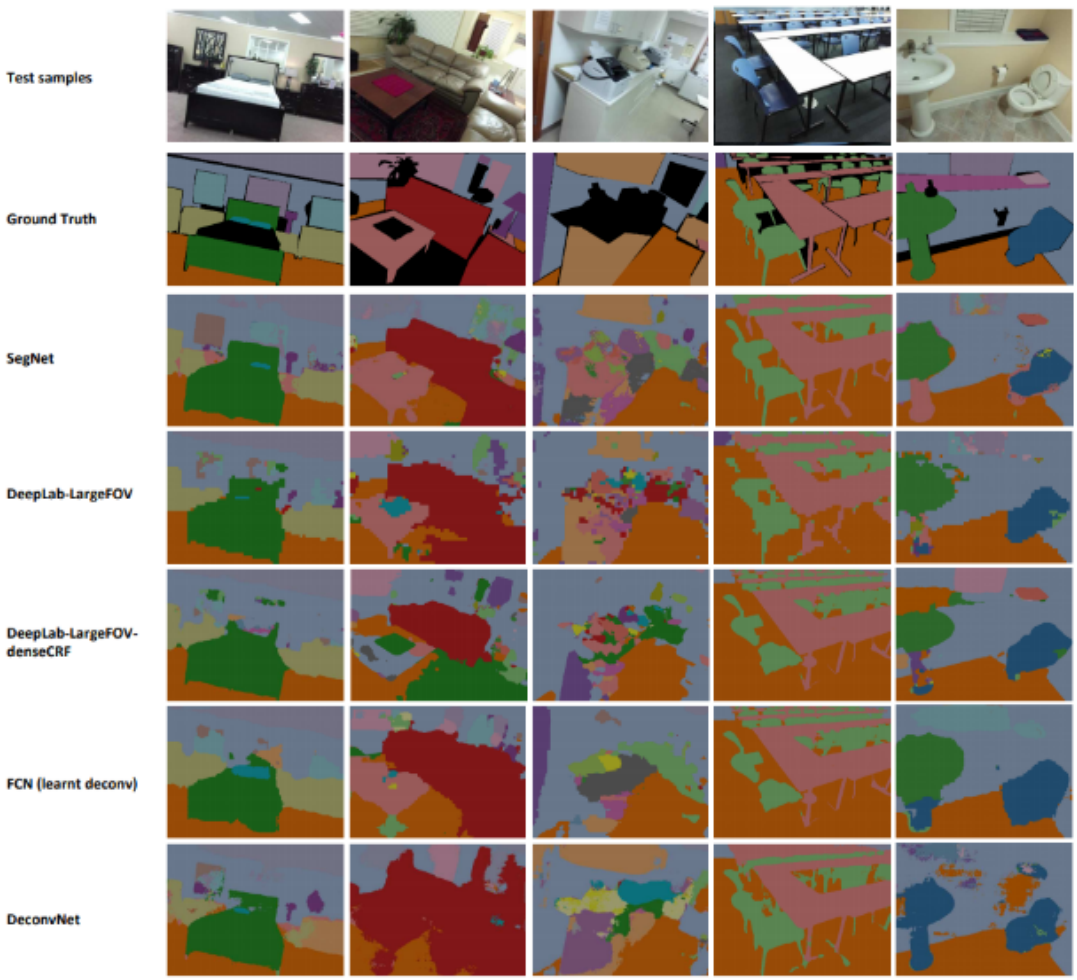


Fig. 5. Qualitative assessment of SegNet predictions on RGB indoor test scenes from the recently released SUN RGB-D dataset [23]. In this hard challenge, SegNet predictions delineate inter class boundaries well for object classes in a variety of scenes and their view-points. Overall rhe segmentation quality is better when object classes are reasonably sized but is very noisy when the scene is more cluttered. Note that often parts of an image of a scene do not have ground truth labels and these are shown in black colour. These parts are not masked in the corresponding deep model predictions that are shown. Note that these results correspond to the model corresponding to the highest mIoU accuracy in Table 4.

Network/Iterations	80K				140K				>140K				Max iter
	G	C	mIoU	BF	G	C	mIoU	BF	G	C	mIoU	BF	
SegNet	70.73	30.82	22.52	9.16	71.66	37.60	27.46	11.33	72.63	44.76	31.84	12.66	240K
DeepLab-LargeFOV [3]	70.70	41.75	30.67	7.28	71.16	42.71	31.29	7.57	71.90	42.21	32.08	8.26	240K
DeepLab-LargeFOV-denseCRF [3]	not computed								66.96	33.06	24.13	9.41	240K
FCN (learnt deconv) [2]	67.31	34.32	24.05	7.88	68.04	37.2	26.33	9.0	68.18	38.41	27.39	9.68	200K
DeconvNet [4]	59.62	12.93	8.35	6.50	63.28	22.53	15.14	7.86	66.13	32.28	22.57	10.47	380K

TABLE 4

Quantitative comparison of deep architectures on the SUNRGB-D dataset when trained on a corpus of 5250 indoor scenes. Note that only the RGB modality was used in these experiments. In this complex task with 37 classes all the architectures perform poorly, particularly because of the smaller sized classes and skew in the class distribution. DeepLab-Large FOV, the smallest and most efficient model has a slightly higher mIoU but SegNet has a better G,C,BF score. Also note that when SegNet was trained with *median frequency class balancing* it obtained 71.75, 44.85, 32.08, 14.06 (180K) as the metrics.

- SegNet은 view가 달라지더라도 크기가 큰 객체들은 잘 잡아내며 다른 모델들보다 reasonable한 결과를 보여줌
- ‘It is also useful to segment decorative objects such as paintings on the wall for AR tasks.’ 라고 주장하는데 전혀 그렇지 않아보임
- 특히 세 번째 컬럼 이미지의 경우 굉장히 난잡한 prediction이 만들어지는데, 이는 주방에 존재하는 객체들이 라벨링되지 않았기 때문에 발생하는 결과
- SegNet이 다른 모델들과 비교해서 G, C, mIoU, BF 모두에서 우수한 성능

- iteration이 적을 때도 성능이 더 잘 나옴

Conclusion

- road, indoor scene segmentation을 위한 효율적인 아키텍처 SegNet 제안
- SegNet은 메모리 사용량 연산 속도 측면에서 다른 모델보다 훨씬 효율적
- Encoder에서 모든 feature map을 저장하는 아키텍처의 경우 성능은 우수하지만 메모리 사용량이 많아지는 단점이 있었으나, SegNet은 un-maxpooling을 사용하여 메모리 사용량을 줄이면서 성능도 개선함