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

Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, Senior Member, IEEE

**Abstract**

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. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network. The role of the decoder network is to map the low resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature map(s). Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps. We compare our proposed architecture with the widely adopted FCN and also with the well known DeepLab-LargeFOV, DeconvNet architectures. This comparison reveals the memory versus accuracy trade-off involved in achieving good segmentation performance.

우리는 SegNet이라 불리는 의미 픽셀 단위 세분화를 위한 새롭고 실용적인 완전 합성 신경 네트워크 아키텍처를 제시한다. 이 코어 트레이닝 가능한 세그멘테이션엔진은, 인코더 네트워크, 대응하는 디코더 네트워크, 그 후에 픽셀 단위의 분류 레이어로 구성됩니다. 인코더 네트워크의 아키텍처는 VGG16 네트워크의 13개의 컨볼루션레이어와 위상적으로 동일합니다. 디코더 네트워크의 역할은 저해상도 인코더 기능 맵을 픽셀 단위 분류를 위해 풀 입력 해상도 기능 맵에 매핑하는 것입니다. SegNet의 새로운 점은 디코더가 낮은 해상도의 입력 기능 맵을 업샘플링하는 방식에 있습니다. 구체적으로는 대응하는 인코더의 최대 풀링 스텝에서 계산된 풀링 인덱스를 사용하여 비선형 업샘플링을 실시한다. 이를 통해 업샘플링을 배울 필요가 없어집니다. 업샘플링된 맵은 희박하며 트레이닝 가능한 필터와 조합되어 고밀도 기능 맵을 생성합니다. 제안 아키텍처를 널리 채택된 FCN 및 잘 알려진 DeepLab-Large와 비교합니다.FOV, DeconvNet 아키텍처 이 비교에서는, 뛰어난 세그먼트화 퍼포먼스를 실현하기 위한 메모리 대 정확도의 트레이드 오프를 알 수 있습니다.

SegNet was primarily motivated by scene understanding applications. Hence, it is designed to be efficient both in terms of memory and computational time during inference. It is also significantly smaller in the number of trainable parameters than other competing architectures and can be trained end-to-end using stochastic gradient descent. We also performed a controlled benchmark of SegNet and other architectures on both road scenes and SUN RGB-D indoor scene segmentation tasks. These quantitative assessments show that SegNet provides good performance with competitive inference time and most efficient inference memory-wise as compared to other architectures. We also provide a Caffe implementation of SegNet and a web demo at http://mi.eng.cam.ac.uk/projects/segnet/.

SegNet은 주로 장면 이해 애플리케이션에 의해 동기 부여되었습니다. 따라서, 그것은 추론하는 동안 메모리와 계산 시간 측면에서 모두 효율적이도록 설계되었다. 또한 다른 경쟁 아키텍처보다 훈련 가능한 매개변수의 수가 상당히 적으며 확률적 경사 강하를 사용하여 엔드 투 엔드로 훈련될 수 있다. 또한 도로 장면과 SUN RGB-D 실내 장면 분할 작업 모두에서 SegNet 및 기타 아키텍처의 제어된 벤치마크를 수행했습니다. 이러한 정량적 평가는 SegNet이 다른 아키텍처에 비해 경쟁적인 추론 시간과 가장 효율적인 추론 시간을 통해 우수한 성능을 제공한다는 것을 보여줍니다. 또한 SegNet의 카페 구현 및 http://mi.eng.cam.ac.uk/projects/segnet/에서 웹 데모를 제공합니다.

1. **Introduction**

Semantic segmentation has a wide array of applications ranging from scene understanding, inferring support-relationships among objects to autonomous driving. Early methods that relied on lowlevel vision cues have fast been superseded by popular machine learning algorithms. In particular, deep learning has seen huge success lately in handwritten digit recognition, speech, categorising whole images and detecting objects in images. Now there is an active interest for semantic pixel-wise labelling. However, some of these recent approaches have tried to directly adopt deep architectures designed for category prediction to pixel-wise labelling. The results, although very encouraging, appear coarse. This is primarily because max pooling and sub-sampling reduce feature map resolution. Our motivation to design SegNet arises from this need to map low resolution features to input resolution for pixel-wise classification. This mapping must produce features which are useful for accurate boundary localization.

의미 분할은 장면 이해에서 객체 간의 지원 관계를 추론하고 자율 주행에 이르는 광범위한 응용 프로그램을 가지고 있습니다. 낮은 수준의 시각 신호에 의존했던 초기 방법은 인기 있는 기계 학습 알고리즘으로 빠르게 대체되었다. 특히 딥 러닝은 최근 손으로 쓴 숫자 인식, 음성, 전체 이미지 분류, 이미지 내 물체 감지 등에서 큰 성공을 거두고 있다. 이제 의미 픽셀 단위 라벨링에 대한 관심이 활발하다. 그러나, 이러한 최근의 접근법 중 일부는 픽셀 단위 라벨링에 대한 범주 예측을 위해 설계된 심층 아키텍처를 직접 채택하려고 시도했다. 결과는 매우 고무적이지만 거칠어 보인다. 이는 주로 최대 풀링과 서브샘플링을 통해 피처 맵의 해상도가 떨어지기 때문입니다. SegNet을 설계한 동기는 픽셀 단위 분류를 위해 저해상도 기능을 입력 해상도에 매핑해야 하는 이러한 필요성에서 비롯됩니다. 이 매핑은 정확한 경계 현지화에 도움이 되는 기능을 생성해야 합니다.

Our architecture, SegNet, is designed to be an efficient architecture for pixel-wise semantic segmentation. It is primarily motivated by road scene understanding applications which require the ability to model appearance (road, building), shape (cars, pedestrians) and understand the spatial-relationship (context) between different classes such as road and side-walk. In typical road scenes, the majority of the pixels belong to large classes such as road, building and hence the network must produce smooth segmentations. The engine must also have the ability to delineate objects based on their shape despite their small size. Hence it is important to retain boundary information in the extracted image representation. From a computational perspective, it is necessary for the network to be efficient in terms of both memory and computation time during inference. The ability to train end-to-end in order to jointly optimise all the weights in the network using an efficient weight update technique such as stochastic gradient descent (SGD) is an additional benefit since it is more easily repeatable. The design of SegNet arose from a need to match these criteria.

당사의 아키텍처인 SegNet은 픽셀 단위의 의미 분할을 위한 효율적인 아키텍처가 되도록 설계되었습니다. 이는 주로 외관(도로, 건물), 형상(자동차, 보행자)을 모델링하고 도로 및 측면 보행과 같은 다른 클래스 간의 공간 관계(콘텍스트)를 이해하는 능력이 필요한 도로 장면 이해 애플리케이션에 의해 동기 부여됩니다. 일반적인 도로 장면에서는 대부분의 픽셀이 도로, 건물 등의 대형 클래스에 속하기 때문에 네트워크는 원활한 분할을 수행해야 합니다. 또한 엔진은 작은 크기에도 불구하고 모양에 따라 물체를 묘사할 수 있어야 합니다. 따라서 추출된 화상 표현에서 경계 정보를 유지하는 것이 중요합니다. 계산의 관점에서, 네트워크는 추론 중 메모리와 계산 시간 측면에서 효율적일 필요가 있다. 확률적 경사 강하(SGD)와 같은 효율적인 가중치 업데이트 기술을 사용하여 네트워크의 모든 가중치를 공동으로 최적화하기 위해 엔드 투 엔드로 훈련하는 기능은 더 쉽게 반복할 수 있기 때문에 추가적인 이점이다. SegNet의 설계는 이러한 기준과 일치해야 하는 필요성에서 비롯되었습니다.

The encoder network in SegNet is topologically identical to the convolutional layers in VGG16. We remove the fully connected layers of VGG16 which makes the SegNet encoder network significantly smaller and easier to train than many other recent architectures. The key component of SegNet is the decoder network which consists of a hierarchy of decoders one corresponding to each encoder. Of these, the appropriate decoders use the max-pooling indices received from the corresponding encoder to perform non-linear upsampling of their input feature maps. This idea was inspired from an architecture designed for unsupervised feature learning. Reusing max-pooling indices in the decoding process has several practical advantages; (i) it improves boundary delineation , (ii) it reduces the number of parameters enabling end-to-end training, and (iii) this form of upsampling can be incorporated into any encoder-decoder architecture such as [2], [10] with only a little modification.

SegNet의 인코더 네트워크는 토폴로지적으로 VGG16의 컨볼루션 레이어와 동일합니다.이것에 의해, SegNet 인코더 네트워크는 다른 많은 최근의 아키텍처보다 큰폭으로 작아져 트레이닝이 용이하게 됩니다. SegNet의 주요 컴포넌트는 각 인코더에 대응하는 디코더 계층으로 구성된 디코더 네트워크입니다. 이 중 적절한 디코더는 대응하는 인코더로부터 수신한 최대 풀링 인덱스를 사용하여 입력 피쳐 맵의 비선형 업샘플링을 수행합니다. 이 아이디어는 감독되지 않은 기능 학습을 위해 설계된 아키텍처에서 영감을 받았습니다. 디코딩 프로세스에서 max-pooling 인덱스를 재사용하는 것은 몇 가지 실질적인 이점이 있습니다. (i) 경계 묘사를 개선하고 (ii) 엔드 투 엔드의 트레이닝을 가능하게 하는 파라미터의 수를 줄이며 (iii) 이러한 업샘플링을 약간의 수정만으로 [2], [10]과 같은 인코더-디코더 아키텍처에 통합할 수 있습니다.

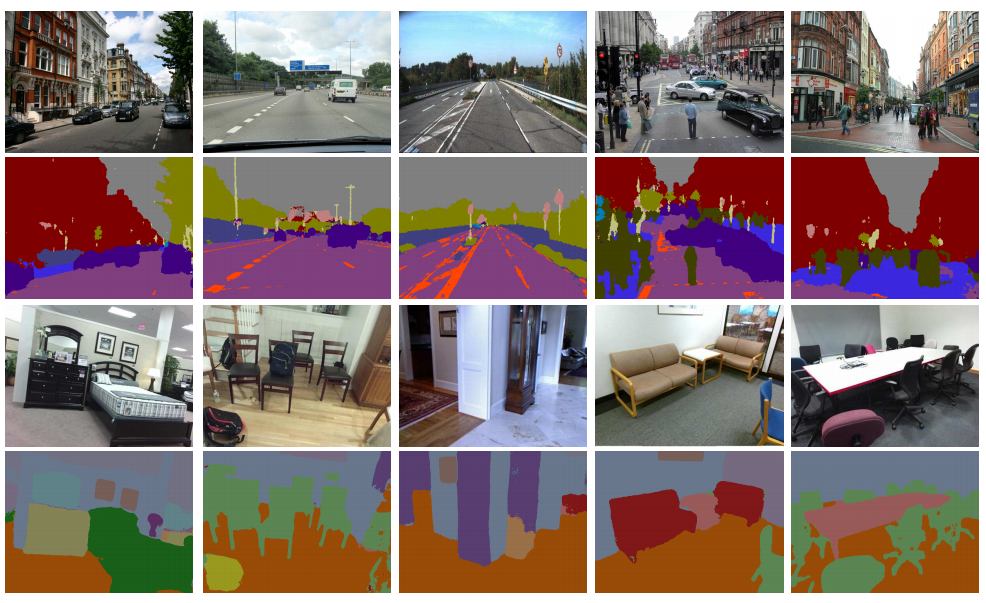


Fig. 1. SegNet predictions on road scenes and indoor scenes. To try our system yourself, please see our online web demo at http://mi.eng.cam.ac.uk/projects/segnet/.

그림 1. 도로 및 실내 장면에 대한 SegNet 예측. 델의 시스템을 직접 사용해 보시려면 http://mi.eng.cam.ac.uk/projects/segnet/의 온라인 웹 데모를 참조해 주세요.

One of the main contributions of this paper is our analysis of the SegNet decoding technique and the widely used Fully Convolutional Network (FCN). This is in order to convey the practical trade-offs involved in designing segmentation architectures. Most recent deep architectures for segmentation have identical encoder networks, i.e VGG16, but differ in the form of the decoder network, training and inference. Another common feature is they have trainable parameters in the order of hundreds of millions and thus encounter difficulties in performing end-to-end training. The difficulty of training these networks has led to multi-stage training, appending networks to a pre-trained architecture such as FCN, use of supporting aids such as region proposals for inference, disjoint training of classification and segmentation networks and use of additional training data for pre-training or for full training. In addition, performance boosting post-processing techniques have also been popular. Although all these factors improve performance on challenging benchmarks, it is unfortunately difficult from their quantitative results to disentangle the key design factors necessary to achieve good performance. We therefore analysed the decoding process used in some of these approaches and reveal their pros and cons.

이 논문의 주요 기여 중 하나는 SegNet 디코딩 기술과 널리 사용되는 완전 컨볼루션 네트워크(FCN)에 대한 분석이다. 이는 세그먼트화 아키텍처 설계와 관련된 실질적인 트레이드오프를 전달하기 위한 것입니다. 분할을 위한 대부분의 최신 딥 아키텍처는 동일한 인코더 네트워크(예: VGG16)를 가지고 있지만 디코더 네트워크, 훈련 및 추론 형태가 다릅니다. 또 다른 공통적인 특징은 훈련 가능한 파라미터가 수억 개에 달하기 때문에 엔드 투 엔드 훈련을 수행하는 데 어려움이 있다는 것입니다. 이러한 네트워크의 훈련의 어려움으로 인해 다단계 훈련, FCN과 같은 사전 훈련 아키텍처에 네트워크 추가, 추론을 위한 지역 제안, 분류 및 세분화 네트워크의 분리 훈련, 사전 훈련 또는 전체 훈련을 위한 추가 훈련 데이터 사용 등이 초래되었습니다. 또한 성능 향상 후 처리 기술도 인기를 끌고 있습니다. 이 모든 요소들이 까다로운 벤치마크의 성능을 향상시키지만, 그 정량적 결과로부터 우수한 성능을 달성하기 위해 필요한 주요 설계 요소를 분리하는 것은 안타깝게도 어렵습니다. 따라서 이러한 접근법 중 일부에 사용된 디코딩 프로세스를 분석하고 장단점을 밝혔다.

We evaluate the performance of SegNet on two scene segmentation tasks, CamVid road scene segmentation and SUN RGB-D indoor scene segmentation. Pascal VOC12 has been the benchmark challenge for segmentation over the years. However, the majority of this task has one or two foreground classes surrounded by a highly varied background. This implicitly favours techniques used for detection as shown by the recent work on a decoupled classification-segmentation network where the classification network can be trained with a large set of weakly labelled data and the independent segmentation network performance is improved. The method of [3] also use the feature maps of the classification network with an independent CRF postprocessing technique to perform segmentation. The performance can also be boosted by the use additional inference aids such as region proposals. Therefore, it is different from scene understanding where the idea is to exploit co-occurrences of objects and other spatial-context to perform robust segmentation. To demonstrate the efficacy of SegNet, we present a real-time online demo of road scene segmentation into 11 classes of interest for autonomous driving (see link in Fig. 1). Some example test results produced on randomly sampled road scene images from Google and indoor test scenes from the SUN RGB-D dataset are shown in Fig. 1.

우리는 CamVid 도로 장면 분할과 SUN RGB-D 실내 장면 분할이라는 두 가지 장면 분할 작업에서 SegNet의 성능을 평가한다. Pascal VOC12는 수년간 세그먼트화의 벤치마크 과제였습니다. 그러나 이 작업의 대부분은 매우 다양한 배경으로 둘러싸인 하나 또는 두 개의 전경 클래스를 가지고 있습니다. 이것은 분류 네트워크가 약하게 라벨이 붙여진 대량의 데이터로 훈련될 수 있고 독립적인 분할 네트워크 성능이 향상되는 분리된 분류 세분화 네트워크에 대한 최근 연구에서 보여지듯이 검출에 사용되는 기술을 암묵적으로 선호한다. 또한 [3]의 방법은 독립적인 CRF 후 처리 기술을 사용하여 분류 네트워크의 기능 맵을 사용하여 분할을 수행합니다. 성과는 또한 지역 제안과 같은 추가 추론 보조 장치를 사용함으로써 향상될 수 있다. 따라서 견고한 분할을 수행하기 위해 객체와 기타 공간 컨텍스트의 공존을 이용하는 것이 아이디어인 장면 이해와는 다릅니다. SegNet의 효과를 입증하기 위해, 우리는 자율 주행을 위한 11가지 관심 등급으로 도로 장면 분할의 실시간 온라인 데모를 제시한다(그림 1의 링크 참조). Google에서 무작위로 추출한 도로 장면 이미지와 SUN RGB-D 데이터 세트의 실내 테스트 장면에서 생성된 테스트 결과의 예는 그림 1과 같다.

The remainder of the paper is organized as follows. In Sec.2 we review related recent literature. We describe the SegNet architecture and its analysis in Sec. 3. In Sec. 4 we evaluate the performance of SegNet on outdoor and indoor scene datasets. This is followed by a general discussion regarding our approach with pointers to future work in Sec. 5. We conclude in Sec. 6.

이 문서의 나머지 내용은 다음과 같이 구성되어 있습니다. 2절에서는 관련된 최신 문헌을 검토한다. 우리는 SegNet 아키텍처와 그 분석을 3항에서 설명한다. 4절에서는 야외 및 실내 장면 데이터 세트에 대한 SegNet의 성능을 평가한다. 이어서 5항의 향후 작업에 대한 지침과 함께 당사의 접근 방식에 대한 일반적인 논의가 이루어집니다. 우리는 6항에서 결론짓는다.

1. **Literature Review**

Semantic pixel-wise segmentation is an active topic of research, fuelled by challenging datasets. Before the arrival of deep networks, the best performing methods mostly relied on hand engineered features classifying pixels independently. Typically, a patch is fed into a classifier e.g. Random3 Forest or Boosting to predict the class probabilities of the center pixel. Features based on appearance or SfM and appearance have been explored for the CamVid road scene understanding test. These per-pixel noisy predictions (often called unary terms) from the classifiers are then smoothed by using a pair-wise or higher order CRF to improve the accuracy. More recent approaches have aimed to produce high quality unaries by trying to predict the labels for all the pixels in a patch as opposed to only the center pixel. This improves the results of Random Forest based unaries but thin structured classes are classified poorly. Dense depth maps computed from the CamVid video have also been used as input for classification using Random Forests. Another approach argues for the use of a combination of popular hand designed features and spatio-temporal super-pixelization to obtain higher accuracy. The best performing technique on the CamVid test addresses the imbalance among label frequencies by combining object detection outputs with classifier predictions in a CRF framework. The result of all these techniques indicate the need for improved features for classification.

의미 픽셀 단위 세분화는 까다로운 데이터 세트에 의해 가속화된 활발한 연구 주제이다. 딥 네트워크가 등장하기 전에는 픽셀을 독립적으로 분류하는 수공학적 기능에 가장 적합한 방법이 대부분이었습니다. 일반적으로 패치는 분류자에 공급됩니다. Random3 Forest 또는 Boosting을 사용하여 중앙 픽셀의 클래스 확률을 예측합니다. CamVid 도로 장면 이해 테스트에서는 외관 또는 SFM 및 외관을 기반으로 한 특징을 살펴보았습니다. 분류기의 이러한 픽셀 단위 노이즈 예측(종종 단항이라고 함)은 쌍별 또는 고차 CRF를 사용하여 평활되어 정확도를 향상시킵니다. 최근의 접근법에서는 중앙 픽셀이 아닌 패치 내의 모든 픽셀의 라벨을 예측함으로써 고품질 유니언리 생성을 목표로 하고 있습니다. 이렇게 하면 랜덤 포레스트 기반 유니리어의 결과가 개선되지만 씬 구조화 클래스는 제대로 분류되지 않습니다. CamVid 비디오에서 계산된 고밀도 깊이 맵은 랜덤 포레스트를 사용한 분류 입력으로도 사용되었습니다. 또 다른 접근방식은 더 높은 정확도를 얻기 위해 인기 있는 수공 디자인 특징과 시공간적 초픽셀라이제이션의 조합을 사용해야 한다고 주장한다. CamVid 테스트에서 가장 뛰어난 성능을 발휘하는 기술은 객체 검출 출력과 CRF 프레임워크의 분류자 예측을 조합함으로써 라벨 주파수 간의 불균형에 대처합니다. 이러한 모든 기술의 결과는 분류를 위한 기능 개선이 필요하다는 것을 나타냅니다.

Indoor RGBD pixel-wise semantic segmentation has also gained popularity since the release of the NYU dataset. This dataset showed the usefulness of the depth channel to improve segmentation. Their approach used features such as RGB-SIFT, depth-SIFT and pixel location as input to a neural network classifier to predict pixel unaries. The noisy unaries are then smoothed using a CRF. Improvements were made using a richer feature set including LBP and region segmentation to obtain higher accuracy followed by a CRF. In more recent work, both class segmentation and support relationships are inferred together using a combination of RGB and depth based cues. Another approach focuses on real-time joint reconstruction and semantic segmentation, where Random Forests are used as the classifier. Gupta et al. use boundary detection and hierarchical grouping before performing category segmentation. The common attribute in all these approaches is the use of hand engineered features for classification of either RGB or RGBD images.

실내 RGBD 픽셀 단위 의미 분할도 NYU 데이터 집합 출시 이후 인기를 끌었다. 이 데이터 세트는 분할을 개선하기 위한 깊이 채널의 유용성을 보여주었다. 이들의 접근법은 픽셀 단항 예측에 신경망 분류기의 입력으로 RGB-SIFT, 깊이-SIFT 및 픽셀 위치와 같은 기능을 사용했습니다. 그런 다음 CRF를 사용하여 노이즈가 많은 유니언리를 평활합니다. 보다 높은 정확도를 얻기 위해 LBP 및 영역 분할을 포함한 보다 풍부한 피쳐 세트를 사용하여 개선되었습니다. 보다 최근의 연구에서는 RGB 및 깊이 기반 단서의 조합을 사용하여 클래스 분할 및 지원 관계를 함께 추론한다. 또 다른 접근방식은 실시간 공동 재구성 및 의미 분할에 초점을 맞추고 있으며, 여기서 랜덤 포레스트는 분류자로 사용된다. 굽타 등 범주 분할을 수행하기 전에 경계 탐지 및 계층적 그룹화를 사용합니다. 이러한 모든 접근법의 공통 속성은 RGB 또는 RGBD 이미지 분류에 수공학적 기능을 사용하는 것입니다.

The success of deep convolutional neural networks for object classification has more recently led researchers to exploit their feature learning capabilities for structured prediction problems such as segmentation. There have also been attempts to apply networks designed for object categorization to segmentation, particularly by replicating the deepest layer features in blocks to match image dimensions. However, the resulting classification is blocky. Another approach using recurrent neural networks merges several low resolution predictions to create input image resolution predictions. These techniques are already an improvement over hand engineered features but their ability to delineate boundaries is poor.

객체 분류를 위한 심층 컨볼루션 신경망의 성공은 최근 연구자들이 세분화와 같은 구조화된 예측 문제에 그들의 특징 학습 능력을 이용하도록 이끌었다. 오브젝트 분류용으로 설계된 네트워크를 세그먼트화에 적용하려는 시도도 있었습니다.특히 블록 내의 가장 깊은 레이어 피쳐를 이미지 치수에 맞추어 복제하는 것이 중요합니다. 단, 그 결과 분류는 블록화되어 있습니다. 반복 신경망을 사용하는 또 다른 접근방식은 입력 이미지 해상도 예측을 생성하기 위해 몇 가지 저해상도 예측을 병합합니다. 이러한 기술은 이미 수공학적 기능에 비해 개선되었지만 경계를 기술하는 능력은 떨어진다.

Newer deep architectures particularly designed for segmentation have advanced the state-of-the-art by learning to decode or map low resolution image representations to pixel-wise predictions. The encoder network which produces these low resolution representations in all of these architectures is the VGG16 classification network which has 13 convolutional layers and 3 fully connected layers. This encoder network weights are typically pre-trained on the large ImageNet object classification dataset. The decoder network varies between these architectures and is the part which is responsible for producing multi-dimensional features for each pixel for classification.

세그먼트화를 위해 특별히 설계된 새로운 딥 아키텍처는 저해상도 이미지 표현을 디코딩하거나 픽셀 단위 예측에 매핑하는 방법을 학습함으로써 최첨단 기술을 발전시켰습니다. 이러한 모든 아키텍처에서 이러한 저해상도 표현을 생성하는 인코더 네트워크는 VGG16 분류 네트워크로, 13개의 컨볼루션 층과 3개의 완전 접속 층이 있습니다. 이 인코더 네트워크의 중량은 일반적으로 대규모 ImageNet 객체 분류 데이터 세트에 대해 사전 훈련됩니다. 디코더 네트워크는 이들 아키텍처 간에 다르며 분류를 위한 각 픽셀의 다차원 기능을 생성하는 부분을 담당합니다.

Each decoder in the Fully Convolutional Network (FCN) architecture learns to upsample its input feature map(s) and combines them with the corresponding encoder feature map to produce the input to the next decoder. It is an architecture which has a large number of trainable parameters in the encoder network (134M) but a very small decoder network (0.5M). The overall large size of this network makes it hard to train end-to-end on a relevant task. Therefore, the authors use a stage-wise training process. Here each decoder in the decoder network is progressively added to an existing trained network. The network is grown until no further increase in performance is observed. This growth is stopped after three decoders thus ignoring high resolution feature maps can certainly lead to loss of edge information. Apart from training related issues, the need to reuse the encoder feature maps in the decoder makes it memory intensive in test time. We study this network in more detail as it the core of other recent architectures.

Fully Convolutional Network(FCN; 완전 컨볼루션 네트워크) 아키텍처의 각 디코더는 그 입력 피쳐 맵을 업샘플링 하는 방법을 학습하고 그것들을 대응하는 인코더 피쳐 맵과 조합하여 다음 디코더에 대한 입력을 생성합니다. 이 아키텍처는 인코더 네트워크(134M)에 트레이닝 가능한 파라미터는 많지만 디코더 네트워크(0.5M)는 매우 작다. 이 네트워크의 전체 규모가 크기 때문에 관련 작업에 대해 엔드 투 엔드로 훈련하는 것이 어렵습니다. 따라서 저자들은 단계별 훈련 과정을 사용한다. 여기서 디코더 네트워크 내의 각 디코더는 기존의 훈련된 네트워크에 단계적으로 추가된다. 네트워크는 퍼포먼스가 더 이상 향상되지 않을 때까지 확장됩니다. 이 성장은 3개의 디코더 후에 정지되므로 고해상도 기능 맵을 무시하면 에지 정보가 손실될 수 있습니다. 트레이닝 관련 문제 외에 디코더 내의 인코더 기능 맵을 재사용할 필요가 있기 때문에 테스트 시에 메모리를 대량으로 소비합니다. 이 네트워크는 다른 최신 아키텍처의 핵심으로서 보다 상세하게 연구합니다.

The predictive performance of FCN has been improved further by appending the FCN with a recurrent neural network (RNN) and fine-tuning them on large datasets. The RNN layers mimic the sharp boundary delineation capabilities of CRFs while exploiting the feature representation power of FCN’s. They show a significant improvement over FCN-8 but also show that this difference is reduced when more training data is used to train FCN-8. The main advantage of the CRF-RNN is revealed when it is jointly trained with an architecture such as the FCN8. The fact that joint training helps is also shown in other recent results. Interestingly, the deconvolutional network performs significantly better than FCN although at the cost of a more complex training and inference. This however raises the question as to whether the perceived advantage of the CRF-RNN would be reduced as the core feed-forward segmentation engine is made better. In any case, the CRF-RNN network can be appended to any deep segmentation architecture including SegNet.

FCN의 예측 성능은 FCN에 RNN(Recurrent Neural Network)을 추가하고 이를 대규모 데이터셋에서 미세 조정하여 더욱 향상되었습니다. RNN 레이어는 FCN의 기능 표현 능력을 이용하면서 CRF의 날카로운 경계 기술 기능을 모방합니다. 이는 FCN-8에 비해 크게 개선되었지만 FCN-8 훈련에 더 많은 훈련 데이터를 사용할 경우 이 차이가 줄어든다는 것을 보여줍니다. CRF-RNN의 주요 장점은 FCN8과 같은 아키텍처와 함께 훈련을 받을 때 드러납니다. 공동훈련이 도움이 된다는 사실은 최근의 다른 결과에서도 드러난다. 흥미롭게도, 디콘볼루션 네트워크는 더 복잡한 훈련과 추론을 희생하더라도 FCN보다 훨씬 더 나은 성능을 발휘합니다. 그러나 이는 핵심 피드포워드 분할 엔진이 개선됨에 따라 CRF-RNN의 인식된 이점이 감소할지에 대한 의문을 제기한다. 어느 경우든 CRF-RNN 네트워크는 SegNet을 포함한 모든 딥 세그멘테이션아키텍처에 추가할 수 있습니다.

Multi-scale deep architectures are also being pursued. They come in two flavours, (i) those which use input images at a few scales and corresponding deep feature extraction networks, and (ii) those which combine feature maps from different layers of a single deep architecture. The common idea is to use features extracted at multiple scales to provide both local and global context and the using feature maps of the early encoding layers retain more high frequency detail leading to sharper class boundaries. Some of these architectures are difficult to train due to their parameter size. Thus a multi-stage training process is employed along with data augmentation. The inference is also expensive with multiple convolutional pathways for feature extraction. Others append a CRF to their multi-scale network and jointly train them. However, these are not feed-forward at test time and require optimization to determine the MAP labels.

멀티스케일 딥아키텍처도 추진되고 있습니다. 그것들은 두 가지 맛, 즉 (i) 몇 개의 스케일로 입력 이미지를 사용하고 대응하는 심층 특징 추출 네트워크를 사용하는 것과 (ii) 단일 심층 아키텍처의 다른 레이어에서 특징 맵을 결합하는 것으로 나타난다. 일반적인 생각은 로컬 및 글로벌컨텍스트를 모두 제공하기 위해 여러 스케일로 추출된 기능을 사용하는 것입니다.또, 초기 부호화 레이어의 기능 맵을 사용하면, 보다 높은 주파수의 상세를 유지할 수 있기 때문에, 클래스 경계가 선명해집니다. 이들 아키텍처 중 일부는 파라미터 크기 때문에 트레이닝이 어렵습니다. 따라서 데이터 확대와 함께 다단계 교육 프로세스가 사용됩니다. 특징 추출을 위한 다중 컨볼루션 경로에서도 추론이 비싸다. 다른 기업은 멀티스케일 네트워크에 CRF를 추가하여 공동으로 훈련합니다. 단, 이들은 테스트 시 피드포워딩되지 않으므로 MAP 라벨을 판별하기 위해 최적화가 필요합니다.

Several of the recently proposed deep architectures for segmentation are not feed-forward in inference time. They require either MAP inference over a CRF or aids such as region proposals for inference. We believe the perceived performance increase obtained by using a CRF is due to the lack of good decoding techniques in their core feed-forward segmentation engine. SegNet on the other hand uses decoders to obtain features for accurate pixel-wise classification.

분할을 위해 최근에 제안된 심층 아키텍처 중 일부는 추론 시간에 피드포워드가 아니다. 그들은 CRF에 대한 MAP 추론 또는 추론을 위한 지역 제안과 같은 보조를 필요로 한다. CRF를 사용하여 얻을 수 있는 인식된 성능 증가는 핵심 피드 포워드 세그먼트화 엔진에 좋은 디코딩 기술이 부족하기 때문이라고 생각합니다. 한편, SegNet은 디코더를 사용하여 정확한 픽셀 단위 분류 기능을 제공합니다.

The recently proposed Deconvolutional Network and its semi-supervised variant the Decoupled network use the max locations of the encoder feature maps (pooling indices) to perform non-linear upsampling in the decoder network. The authors of these architectures, independently of SegNet (first submitted to CVPR 2015), proposed this idea of decoding in the decoder network. However, their encoder network consists of the fully connected layers from the VGG-16 network which consists of about 90% of the parameters of their entire network. This makes training of their network very difficult and thus require additional aids such as the use of region proposals to enable training. Moreover, during inference these proposals are used and this increases inference time significantly. From a benchmarking point of view, this also makes it difficult to evaluate the performance of their architecture (encoder-decoder network) without other aids. In this work we discard the fully connected layers of the VGG16 encoder network which enables us to train the network using the relevant training set using SGD optimization. Another recent method shows the benefit of reducing the number of parameters significantly without sacrificing performance, reducing memory consumption and improving inference time.

최근에 제안된 디콘볼루션네트워크와 그 반감시형 디콘볼루션네트워크는 인코더 피쳐 맵(풀링 인덱스)의 최대 위치를 사용하여 디코더 네트워크에서 비선형 업샘플링을 수행합니다. 이러한 아키텍처의 저자들은 (CVPR 2015에 처음 제출된) 세그넷과는 독립적으로, 디코더 네트워크에서 디코딩하는 아이디어를 제안하였습니다. 단, 인코더 네트워크는 네트워크 전체의 파라미터의 약 90%를 차지하는 VGG-16 네트워크의 완전 접속 레이어로 구성됩니다. 이로 인해 네트워크의 트레이닝이 매우 어려워지기 때문에 트레이닝을 가능하게 하기 위한 지역 제안서 등의 추가 지원이 필요합니다. 더욱이, 추론하는 동안 이러한 제안들이 사용되며 이는 추론 시간을 크게 증가시킨다. 벤치마킹의 관점에서 보면, 이것은 다른 지원 없이는 아키텍처(인코더-디코더 네트워크)의 성능을 평가하는 것을 어렵게 합니다. 이 작업에서는 SGD 최적화를 사용하여 관련 훈련 세트를 사용하여 네트워크를 훈련할 수 있는 VGG16 인코더 네트워크의 완전히 연결된 레이어를 폐기합니다. 또 다른 최근의 방법은 성능을 희생하지 않고 파라미터 수를 대폭 줄이고 메모리 소비를 줄이고 추론 시간을 개선하는 이점을 보여준다.

Our work was inspired by the unsupervised feature learning architecture proposed by Ranzato et al. The key learning module is an encoder-decoder network. An encoder consists of convolution with a filter bank, element-wise tanh non-linearity, max-pooling and sub-sampling to obtain the feature maps. For each sample, the indices of the max locations computed during pooling are stored and passed to the decoder. The decoder upsamples the feature maps by using the stored pooled indices. It convolves this upsampled map using a trainable decoder filter bank to reconstruct the input image. This architecture was used for unsupervised pre-training for classification. A somewhat similar decoding technique is used for visualizing trained convolutional networks for classification. The architecture of Ranzato et al. mainly focused on layer-wise feature learning using small input patches. This was extended by Kavukcuoglu et. al. to accept full image sizes as input to learn hierarchical encoders. Both these approaches however did not attempt to use deep encoder-decoder networks for unsupervised feature training as they discarded the decoders after each encoder training. Here, SegNet differs from these architectures as the deep encoder-decoder network is trained jointly for a supervised learning task and hence the decoders are an integral part of the network in test time.

우리의 연구는 Ranzato 등이 제안한 감독되지 않은 기능 학습 아키텍처에서 영감을 얻었다. 키 러닝 모듈은 인코더/디코더 네트워크입니다. 인코더는 필터 뱅크와의 컨볼루션, 요소별 tanh 비선형성, max-pooling 및 서브샘플링으로 구성되어 피쳐 맵을 얻을 수 있습니다. 각 샘플에 대해 풀링 중에 계산된 최대 로케이션의 인덱스가 저장되어 디코더에 전달됩니다. 디코더는 저장된 풀링된 인덱스를 사용하여 기능 맵을 업샘플링합니다. 트레이닝 가능한 디코더 필터 뱅크를 사용하여 이 업샘플링된 맵을 합성하여 입력 이미지를 재구성합니다. 이 아키텍처는 분류를 위한 감독되지 않은 사전 훈련에 사용되었다. 분류를 위해 훈련된 컨볼루션 네트워크를 가시화하기 위해 다소 유사한 복호화 기술이 사용된다. Ranzato 등의 아키텍처는 주로 작은 입력 패치를 사용한 계층별 기능 학습에 초점을 맞췄다. 이는 Kavukcuoglu 등에 의해 확장되어 계층적 인코더를 학습하기 위한 입력으로 전체 영상 크기를 수용했다. 단, 이들 접근법 모두 각 인코더 훈련 후에 디코더를 폐기했기 때문에 비감독 기능 훈련에 딥 인코더/디코더 네트워크를 사용하려고 하지 않았습니다. 여기서 SegNet은 이러한 아키텍처와는 다릅니다.이는 딥 인코더/디코더 네트워크가 감독되는 학습 태스크를 위해 공동으로 훈련되기 때문입니다.따라서 디코더는 테스트 시 네트워크의 필수적인 부분입니다.

Other applications where pixel wise predictions are made using deep networks are image super-resolution and depth map prediction from a single image. The authors in discuss the need for learning to upsample from low resolution feature maps which is the central topic of this paper.

딥 네트워크를 이용해 픽셀 와이즈 예측을 하는 다른 응용 프로그램으로는 이미지 초해상도 및 단일 이미지에서의 깊이 지도 예측이 있습니다. 의 저자들은 이 문서의 핵심 주제인 저해상도 기능 맵에서 업샘플링을 배울 필요가 있다고 설명합니다.

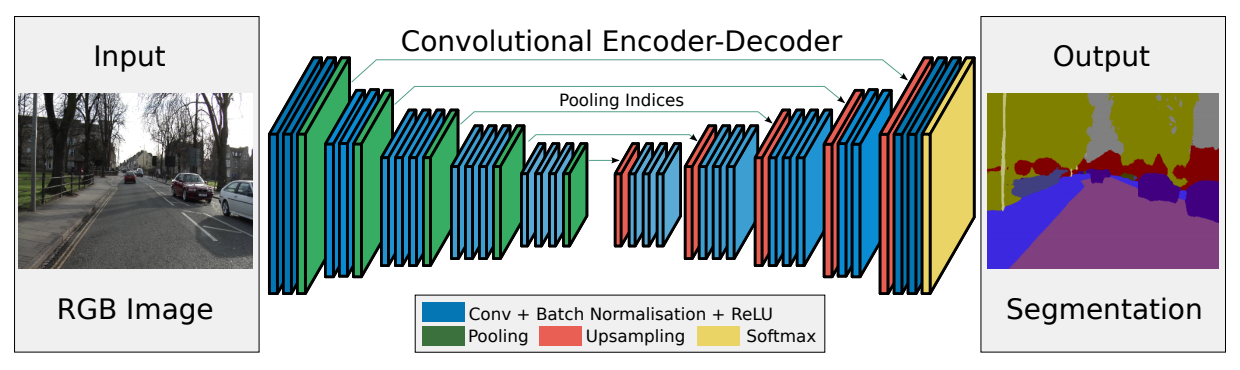


Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to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

그림 2. SegNet 아키텍처의 그림. 완전히 연결된 레이어는 없으므로 컨볼루션일 뿐입니다. A디코더의 입력은 인코더에서 희박한 형상 map(s)을 생산하여 편입을 수영장 지수를 사용하여 upsamples. 그런 다음 트레이닝 가능한 필터 뱅크와 컨볼루션하여 기능 맵을 조밀하게 만듭니다. 최종 디코더 출력 피쳐 맵은 픽셀 단위 분류를 위해 소프트 최대 분류기에 공급됩니다.

1. **Architecture**

SegNet has an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer. This architecture is illustrated in Fig. 3. The encoder network consists of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network designed for object classification. We can therefore initialize the training process from weights trained for classification on large datasets. We can also discard the fully connected layers in favour of retaining higher resolution feature maps at the deepest encoder output. This also reduces the number of parameters in the SegNet encoder network significantly (from 134M to 14.7M) as compared to other recent architectures (see. Table 6). Each encoder layer has a corresponding decoder layer and hence the decoder network has 13 layers. The final decoder output is fed to a multi-class soft-max classifier to produce class probabilities for each pixel independently.

SegNet은 인코더 네트워크와 대응하는 디코더 네트워크를 가지며, 최종 픽셀 단위 분류 레이어가 이어진다. 이 아키텍처는 그림 3에 나타나 있습니다. 인코더 네트워크는 객체 분류용으로 설계된 VGG16 네트워크의 첫 번째 13개의 컨볼루션 층에 대응하는 13개의 컨볼루션 층으로 구성됩니다. 따라서 대규모 데이터 세트에 대한 분류를 위해 훈련한 체중에서 훈련 과정을 초기화할 수 있다. 또, 완전 접속 레이어를 폐기하고, 가장 깊은 인코더 출력으로 고해상도 기능 맵을 유지할 수도 있습니다. 이것에 의해, 다른 최근의 아키텍처에 비해, SegNet 인코더 네트워크의 파라메타의 수도 큰폭(134M에서14.7M)으로 감소합니다( 를 참조). 표 6). 각 인코더 층은 대응하는 디코더 층을 가지며, 따라서 디코더 네트워크는 13개의 층을 가진다. 최종 디코더 출력은 멀티클래스 소프트맥스 분류기에 공급되어 각 픽셀에 대해 독립적으로 클래스 확률을 생성한다.

Each encoder in the encoder network performs convolution with a filter bank to produce a set of feature maps. These are then batch normalized). Then an element-wise rectifiedlinear non-linearity (ReLU) max(0, x) is applied. Following that, max-pooling with a 2 × 2 window and stride 2 (non-overlapping window) is performed and the resulting output is sub-sampled by a factor of 2. Max-pooling is used to achieve translation invariance over small spatial shifts in the input image. Sub-sampling results in a large input image context (spatial window) for each pixel in the feature map. While several layers of max-pooling and sub-sampling can achieve more translation invariance for robust classification correspondingly there is a loss of spatial resolution of the feature maps. The increasingly lossy (boundary detail) image representation is not beneficial for segmentation where boundary delineation is vital. Therefore, it is necessary to capture and store boundary information in the encoder feature maps before sub-sampling is performed. If memory during inference is not constrained, then all the encoder feature maps (after subsampling) can be stored. This is usually not the case in practical applications and hence we propose a more efficient way to store this information. It involves storing only the max-pooling indices, i.e, the locations of the maximum feature value in each pooling window is memorized for each encoder feature map. In principle, this can be done using 2 bits for each 2 × 2 pooling window and is thus much more efficient to store as compared to memorizing feature map(s) in float precision. As we show later in this work, this lower memory storage results in a slight loss of accuracy but is still suitable for practical applications.

인코더 네트워크 내의 각 인코더는 필터 뱅크와 컨볼루션하여 기능 맵 세트를 생성한다. 그런 다음 배치 정규화됩니다). 그런 다음 요소별 정류 선형 비선형성(ReLU) max(0, x)가 적용됩니다. 그 후, 2×2의 창과 스트라이드 2(중복하지 않는 창)에 의한 최대 풀링을 실시해, 결과 출력을 2의 배수로 서브 샘플링 한다.최대 풀링을 사용하여 입력 화상의 작은 공간 시프트에 대해 변환 불변성을 실현한다. 하위 샘플링을 수행하면 피쳐 맵의 각 픽셀에 대해 큰 입력 영상 컨텍스트(공간 창)가 생성됩니다. max-pooling 및 sub-sampling의 여러 레이어가 강력한 분류를 위해 더 많은 변환 불변성을 달성할 수 있지만 그에 따라 피쳐 맵의 공간 분해능이 손실됩니다. 점점 손실되는(경계 상세) 영상 표현은 경계 묘사가 중요한 분할에 도움이 되지 않습니다. 따라서 서브샘플링을 실행하기 전에 경계정보를 캡처하여 인코더 피쳐 맵에 저장해야 합니다. 추론 중 메모리가 구속되지 않으면 모든 인코더 기능 맵(서브샘플링 후)을 저장할 수 있습니다. 이는 일반적으로 실제 응용 프로그램에서는 해당되지 않으므로 이 정보를 보다 효율적으로 저장할 수 있는 방법을 제안합니다. 여기에는 max-pooling 인덱스만 저장됩니다.즉, 각 풀링 창의 최대 피쳐 값의 위치가 각 인코더 피쳐 맵에 대해 기억됩니다. 원칙적으로 이것은 각 2 × 2 풀링 윈도우에 대해 2비트를 사용하여 수행할 수 있으며, 따라서 float precision으로 피쳐 맵을 기억하는 것에 비해 저장하는 것이 훨씬 효율적입니다. 이 작업의 후반부에서 볼 수 있듯이, 이 낮은 메모리 스토리지로 인해 정확도가 약간 떨어지게 되지만, 여전히 실용적인 애플리케이션에 적합합니다.

The appropriate decoder in the decoder network upsamples its input feature map(s) using the memorized max-pooling indices from the corresponding encoder feature map(s). This step produces sparse feature map(s). This SegNet decoding technique is illustrated in Fig. 3. These feature maps are then convolved with a trainable decoder filter bank to produce dense feature maps. A batch normalization step is then applied to each of these maps. Note that the decoder corresponding to the first encoder (closest to the input image) produces a multi-channel feature map, although its encoder input has 3 channels (RGB). This is unlike the other decoders in the network which produce feature maps with the same number of size and channels as their encoder inputs. The high dimensional feature representation at the output of the final decoder is fed to a trainable soft-max classifier. This soft-max classifies each pixel independently. The output of the soft-max classifier is a K channel image of probabilities where K is the number of classes. The predicted segmentation corresponds to the class with maximum probability at each pixel.

디코더 네트워크 내의 적절한 디코더는 대응하는 인코더 피쳐 맵에서 기억된 최대 풀링 인덱스를 사용하여 입력 피쳐 맵을 업샘플링합니다. 이 절차에서는 sparse 기능 맵이 생성됩니다. 이 SegNet 디코딩 기법은 그림 3에 나타나 있다. 그런 다음 이러한 기능 맵을 트레이닝 가능한 디코더 필터 뱅크와 컨볼루션하여 고밀도 기능 맵을 만듭니다. 그런 다음 배치 정규화 단계가 각 맵에 적용됩니다. 첫 번째 인코더(입력 이미지에 가장 가까운)에 대응하는 디코더는 멀티채널피처 맵을 생성하는 것에 주의해 주세요.단, 인코더 입력에는 3채널(RGB)이 있습니다. 이는 인코더 입력과 동일한 수의 크기 및 채널로 기능 맵을 생성하는 네트워크 내의 다른 디코더와는 다릅니다. 최종 디코더의 출력에서의 고차원 피쳐 표현은 훈련 가능한 소프트맥스 분류기에 공급됩니다. 이 soft-max는 각 픽셀을 독립적으로 분류합니다. soft-max 분류기의 출력은 확률의 K채널 이미지입니다.K는 클래스 수입니다. 예측된 분할은 각 픽셀에서 최대 확률을 갖는 클래스에 해당합니다.

We add here that two other architectures, DeconvNet and U-Net share a similar architecture to SegNet but with some differences. DeconvNet has a much larger parameterization, needs more computational resources and is harder to train end-to-end (Table 6), primarily due to the use of fully connected layers (albeit in a convolutional manner) We report several comparisons with DeconvNet later in the paper Sec. 4.

여기에 DeconvNet과 U-Net이라는 두 개의 다른 아키텍처가 SegNet과 유사한 아키텍처를 공유하지만 약간의 차이가 있다는 점을 덧붙입니다. DeconvNet은 매개변수화가 훨씬 더 크고, 더 많은 계산 리소스가 필요하며, 엔드 투 엔드(표 6)를 교육하는 것이 더 어렵습니다. 주로 완전히 연결된 계층(컨볼루션 방식이지만)의 사용 때문입니다. 우리는 DeconvNet과의 몇 가지 비교를 논문의 뒷부분에 보고합니다.

As compared to SegNet, U-Net (proposed for the medical imaging community) does not reuse pooling indices but instead transfers the entire feature map (at the cost of more memory) to the corresponding decoders and concatenates them to upsampled (via deconvolution) decoder feature maps. There is no conv5 and max-pool 5 block in U-Net as in the VGG net architecture. SegNet, on the other hand, uses all of the pre-trained convolutional layer weights from VGG net as pre-trained weights.

SegNet과 비교하여 U-Net(의료 영상 커뮤니티용으로 제안됨)은 풀링 인덱스를 재사용하지 않고 대신 전체 기능 맵을 해당하는 디코더로 전송하고 업샘플링(디콘볼루션) 디코더 기능 맵에 연결합니다. VGG 네트워크 아키텍처와 같이 U-Net에는 conv5 및 max-pool 5 블록이 없습니다. 반면, SegNet은 VGG 넷에서 사전 훈련한 모든 컨볼루션 레이어 가중치를 사전 훈련한 가중치로 사용한다.

* 1. **Decoder Variants**

Many segmentation architectures share the same encoder network and they only vary in the form of their decoder network. Of these we choose to compare the SegNet decoding technique with the widely used Fully Convolutional Network (FCN) decoding technique.

많은 세그멘테이션아키텍처는 같은 인코더 네트워크를 공유하며 디코더 네트워크의 형태만 다릅니다. 이 중 SegNet 디코딩 기술을 널리 사용되는 FCN(Fully Convolutional Network) 디코딩 기술과 비교하기로 선택한다.

In order to analyse SegNet and compare its performance with FCN (decoder variants) we use a smaller version of SegNet, termed SegNet-Basic 1, which has 4 encoders and 4 decoders. All the encoders in **SegNet-Basic** perform max-pooling and subsampling and the corresponding decoders upsample its input using the received max-pooling indices. Batch normalization is used after each convolutional layer in both the encoder and decoder network. No biases are used after convolutions and no ReLU nonlinearity is present in the decoder network. Further, a constant kernel size of 7 × 7 over all the encoder and decoder layers is chosen to provide a wide context for smooth labelling i.e. a pixel in the deepest layer feature map (layer 4) can be traced back to a context window in the input image of 106×106 pixels. This small size of SegNet-Basic allows us to explore many different variants (decoders) and train them in reasonable time. Similarly we create **FCN-Basic**, a comparable version of FCN for our analysis which shares the same encoder network as SegNet-Basic but with the FCN decoding technique (see Fig. 3) used in all its decoders.

SegNet을 분석하여 FCN(디코더 바리안트)과 성능을 비교하기 위해 SegNet-Basic 1이라고 하는 작은 버전의 SegNet을 사용합니다.이 SegNet은 4개의 인코더와 4개의 디코더를 갖추고 있습니다. **SegNet-Basic**의 모든 인코더는 max-pooling 및 서브샘플링을 실행하고 대응하는 디코더는 수신된 max-pooling 인덱스를 사용하여 입력을 업샘플링합니다. 배치 정규화는 인코더와 디코더 네트워크 양쪽의 각 컨볼루션층 후에 사용됩니다. 컨볼루션 후에는 바이어스가 사용되지 않으며 디코더 네트워크에는 ReLU 비선형성이 존재하지 않습니다. 또한 모든 인코더 및 디코더 레이어에 걸쳐 7×7의 일정한 커널 사이즈를 선택하여 부드러운 라벨링을 위한 넓은 콘텍스트를 제공한다.즉, 가장 깊은 레이어 특징 맵(레이어 4)의 화소는 106×106 화소의 입력 화상의 콘텍스트 창으로 트레이스 할 수 있다. 이 작은 크기의 SegNet-Basic을 통해 다양한 변종(디코더)을 탐색하고 적절한 시간에 훈련할 수 있습니다. 마찬가지로 우리는 분석을 위해 FCN의 동등한 버전인 **FCN-Basic**을 만듭니다. FCN은 SegNet-Basic과 동일한 인코더 네트워크를 공유하지만 모든 디코더에 사용되는 FCN 디코딩 기술(그림 3 참조)을 사용합니다.

On the left in Fig. 3 is the decoding technique used by SegNet (also SegNet-Basic), where there is no learning involved in the upsampling step. However, the upsampled maps are convolved with trainable multi-channel decoder filters to densify its sparse inputs. Each decoder filter has the same number of channels as the number of upsampled feature maps. A smaller variant is one where the decoder filters are single channel, i.e they only convolve their corresponding upsampled feature map. This variant (**SegNetBasic-SingleChannelDecoder**) reduces the number of trainable parameters and inference time significantly.

그림 3의 왼쪽은 SegNet(SegNet-Basic도 마찬가지)에서 사용하는 디코딩 기법이며, 업샘플링 단계에서는 학습이 필요하지 않습니다. 그러나 업샘플링된 맵은 훈련 가능한 멀티채널 디코더 필터로 컨볼루션되어 스파스 입력을 조밀하게 합니다. 각 디코더 필터에는 업샘플링된 기능 맵의 수와 같은 수의 채널이 있습니다. 더 작은 변종으로는 디코더 필터가 단일 채널인 경우, 즉 대응하는 업샘플링 기능 맵만 컨버전합니다. 이 바리안트(**SegNetBasic-SingleChannelDecoder**)는 트레이닝 가능한 파라미터의 수와 추론 시간을 대폭 단축합니다.

On the right in Fig. 3 is the FCN (also FCN-Basic) decoding technique. The important design element of the FCN model is dimensionality reduction step of the encoder feature maps. This compresses the encoder feature maps which are then used in the corresponding decoders. Dimensionality reduction of the encoder feature maps, say of 64 channels, is performed by convolving them with 1 × 1 × 64 × K trainable filters, where K is the number of classes. The compressed K channel final encoder layer feature maps are the input to the decoder network. In a decoder of this network, upsampling is performed by inverse convolution using a fixed or trainable multi-channel upsampling kernel. We set the kernel size to 8 × 8. This manner of upsampling is also termed as deconvolution. Note that, in comparison, SegNet the multi-channel convolution using trainable decoder filters is performed after upsampling to densifying feature maps. The upsampled feature map in FCN has K channels. It is then added element-wise to the corresponding resolution encoder feature map to produce the output decoder feature map. The upsampling kernels are initialized using bilinear interpolation weights.

그림 3의 오른쪽은 FCN(또한 FCN-Basic) 디코딩 기법입니다. FCN 모델의 중요한 설계 요소는 인코더 기능 맵의 치수 축소 단계입니다. 이것에 의해, 인코더 기능 맵이 압축되어 대응하는 디코더로 사용됩니다. 인코더 피쳐 맵의 치수 축소(예를 들어 64 채널)는 1 × 1 × 64 × K의 훈련 가능한 필터를 사용하여 컴볼레이션함으로써 실행됩니다.여기서 K는 클래스의 수입니다. 압축된 K채널 최종 인코더 레이어피처 맵은 디코더 네트워크에 대한 입력입니다. 이 네트워크의 디코더에서 업샘플링은 고정 또는 트레이닝 가능한 멀티채널 업샘플링 커널을 사용하여 역컨볼루션에 의해 이루어진다. 커널 크기를 8 × 8로 설정했습니다. 이러한 업샘플링 방식을 디콘볼루션이라고도 합니다. 이에 비해 SegNet은 트레이닝 가능한 디코더 필터를 사용한 멀티채널 컨볼루션은 고밀도 기능 맵으로 업샘플링한 후에 실행됩니다. FCN의 업샘플링 기능 맵에는 K채널이 있습니다. 그런 다음 대응하는 해상도 인코더 기능 맵에 요소별로 추가되어 출력 디코더 기능 맵이 생성됩니다. 업샘플링 커널은 쌍선형 보간 가중치를 사용하여 초기화됩니다.

The FCN decoder model requires storing encoder feature maps during inference. This can be memory intensive for embedded applications; for e.g. storing 64 feature maps of the first layer of FCN-Basic at 180 × 240 resolution in 32 bit floating point precision takes 11MB. This can be made smaller using dimensionality reduction to the 11 feature maps which requires ≈ 1.9MB storage. SegNet on the other hand requires almost negligible storage cost for the pooling indices (.17MB if stored using 2 bits per 2 × 2 pooling window). We can also create a variant of the FCN-Basic model which discards the encoder feature map addition step and only learns the upsampling kernels (**FCN-Basic-NoAddition**).

FCN 디코더 모델에서는 추론 중에 인코더 기능 맵을 저장해야 합니다. 내장형 어플리케이션에서는 메모리 부하가 높은 경우가 있습니다.예를 들어 FCN-Basic의 첫 번째 레이어 기능 맵 64개를 해상도 180×240으로 32비트 부동소수점 정밀도로 저장하는 경우 11MB가 소요됩니다. 이는 치수 축소를 통해 feature1.9가 필요한 11개의 기능 맵으로 축소할 수 있습니다.MB 스토리지 반면 SegNet은 풀링 인덱스에 대해 거의 무시해도 될 정도의 스토리지 비용이 필요합니다(2 × 2 풀링 창당 2비트를 사용하여 저장하는 경우 17MB). 인코더 기능 맵의 추가 스텝은 폐기하고 업샘플링 커널(**FCN-Basic-NoAddition**)만 학습하는 FCN-Basic 모델의 배리언트를 작성할 수도 있습니다.

In addition to the above variants, we study upsampling using fixed bilinear interpolation weights which therefore requires no learning for upsampling (**Bilinear-Interpolation**). At the other extreme, we can add 64 encoder feature maps at each layer to the corresponding output feature maps from the SegNet decoder to create a more memory intensive variant of SegNet (**SegNet-BasicEncoderAddition**). Here both the pooling indices for upsampling are used, followed by a convolution step to densify its sparse input. This is then added element-wise to the corresponding encoder feature maps to produce a decoders output.

위의 변형 외에도, 우리는 업샘플링에 대한 학습이 필요 없는 고정 **쌍선형 보간** 가중치를 사용한 업샘플링을 연구한다. 또, SegNet 디코더로부터의 대응하는 출력 피처 맵에, 각 레이어에서 64개의 인코더 피처 맵을 추가하고, 보다 메모리 부하가 높은 SegNet(**SegNet-BasicEncoderAddition**)을 작성할 수 있습니다. 여기서는 업샘플링을 위한 풀링 지수를 모두 사용하고, 이어서 희박한 입력을 조밀하게 하는 컨볼루션 단계를 사용한다. 다음으로 대응하는 인코더 기능 맵에 요소별로 추가되어 디코더 출력이 생성됩니다.

Another and more memory intensive FCN-Basic variant (**FCN-Basic-NoDimReduction**) is where there is no dimensionality reduction performed for the encoder feature maps. This implies that unlike FCN-Basic the final encoder feature map is not compressed to K channels before passing it to the decoder network. Therefore, the number of channels at the end of each decoder is the same as the corresponding encoder (i.e 64).

메모리 부하가 높은 FCN-Basic 배리언트(**FCN-Basic-NoDimReduction**)는 인코더 기능 맵에 대해 치수 축소가 실행되지 않는 경우입니다. 이는 FCN-Basic과 달리 최종 인코더 기능 맵은 디코더 네트워크에 전달되기 전에 K채널로 압축되지 않음을 의미합니다. 따라서 각 디코더 끝에 있는 채널 수는 대응하는 인코더(64)와 동일합니다.

We also tried other generic variants where feature maps are simply upsampled by replication, or by using a fixed (and sparse) array of indices for upsampling. These performed quite poorly in comparison to the above variants. A variant without max-pooling and sub-sampling in the encoder network (decoders are redundant) consumes more memory, takes longer to converge and performs poorly. Finally, please note that to encourage reproduction of our results we release the Caffe implementation of all the variants.

또한 기능 맵이 복제 또는 업샘플링을 위해 고정된(그리고 희박한) 인덱스 배열을 사용하여 단순히 업샘플링되는 다른 일반적인 변형도 시도했다. 위의 변형에 비해 성능이 상당히 떨어졌습니다. 인코더 네트워크에서 max-pooling 및 서브샘플링이 없는 배리언트(디코더는 용장)는 메모리 소비량이 많아 컨버전스에 시간이 걸리고 퍼포먼스가 저하됩니다. 마지막으로, 결과 재현을 장려하기 위해 모든 변종 Cafe 구현을 출시합니다.

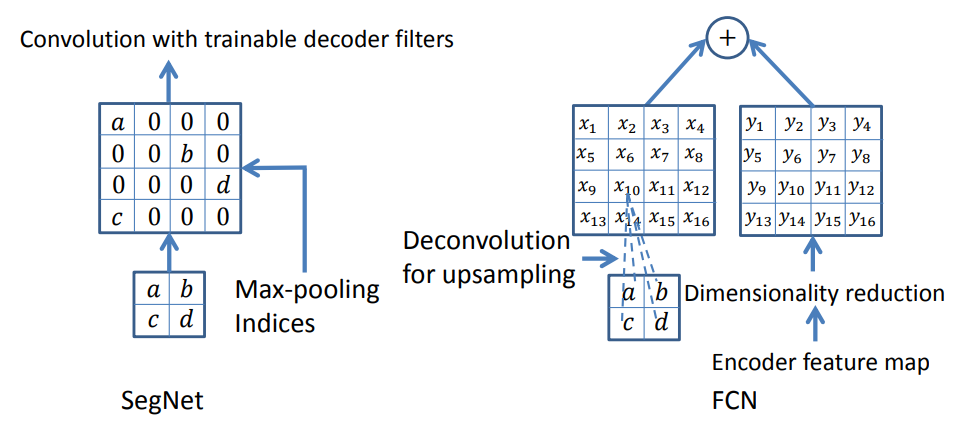


Fig. 3. An illustration of SegNet and FCN decoders. a, b, c, d correspond to values in a feature map. SegNet uses the max pooling indices to upsample (without learning) the feature map(s) and convolves with a trainable decoder filter bank. FCN upsamples by learning to deconvolve the input feature map 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.

그림 3. SegNet 및 FCN 디코더의 그림 a, b, c, d는 피처 맵의 값에 대응합니다. SegNet은 최대 풀링 인덱스를 사용하여 (학습하지 않고) 기능 맵을 업샘플링하고 트레이닝 가능한 디코더 필터 뱅크를 사용하여 컨버전합니다. FCN은 입력 기능 맵의 디콘볼루션 방법을 학습함으로써 업샘플링을 수행하고 대응하는 인코더 기능 맵을 추가하여 디코더 출력을 생성합니다. 이 기능 맵은 대응하는 인코더 내의 max-pooling 레이어(서브샘플링 포함)의 출력입니다. FCN에는 트레이닝 가능한 디코더 필터가 없습니다.

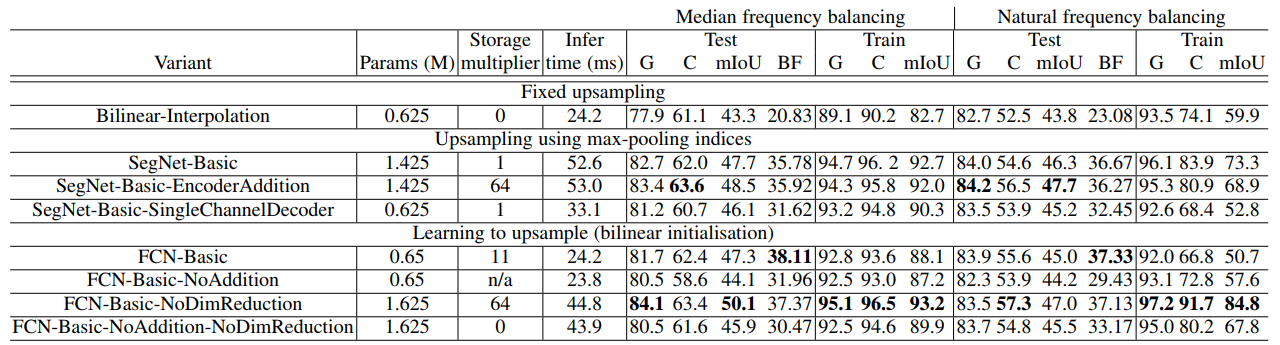


TABLE 1 Comparison of decoder variants. We quantify the performance using global (G), class average (C), mean of intersection over union (mIoU) and a semantic contour measure(BF). The testing and training accuracies are shown as percentages for both natural frequency and median frequency balanced training loss function. SegNet-Basic performs at the same level as FCN-Basic but requires only storing max-pooling indices and is therefore more memory efficient during inference. Note that the theoretical memory requirement reported is based only on the size of the first layer encoder feature map. FCN-Basic, SegNet-Basic, SegNet-Basic-EncoderAddition all have high BF scores indicating the need to use information in encoder feature maps for better class contour delineation. Networks with larger decoders and those using the encoder feature maps in full perform best, although they are least efficient in terms of inference time and memory.

표 1 디코더 종류 비교 우리는 글로벌(G), 클래스 평균(C), 결합 교차로 평균(mIoU) 및 의미 등고선 측정(BF)을 사용하여 성능을 정량화한다. 테스트 및 훈련 정확도는 자연 주파수 및 중앙 주파수 균형 훈련 손실 함수에 대한 백분율로 표시됩니다. SegNet-Basic은 FCN-Basic과 동일한 수준으로 동작하지만 최대 풀링 인덱스만 저장하면 되므로 추론 시 메모리 효율이 향상됩니다. 보고되는 이론적인 메모리 요건은 첫 번째 레이어 인코더 기능 맵의 크기만을 기반으로 합니다. FCN-Basic, SegNet-Basic, SegNet-Basic-EncoderAddition은 모두 높은 BF 점수를 가지고 있어 보다 나은 클래스 윤곽선 묘사를 위해 인코더 피처 맵의 정보를 사용할 필요가 있습니다. 디코더가 큰 네트워크 및 인코더 기능 맵을 모두 사용하는 네트워크는 추론 시간과 메모리 측면에서 효율이 떨어지지만 성능이 가장 우수합니다.

* 1. **Training**

We use the CamVid road scenes dataset to benchmark the performance of the decoder variants. This dataset is small, consisting of 367 training and 233 testing RGB images (day and dusk scenes) at 360×480 resolution. The challenge is to segment 11 classes such as road, building, cars, pedestrians, signs, poles, side-walk etc. We perform local contrast normalization to the RGB input.

CamVid 도로 장면 데이터 세트를 사용하여 디코더 변형의 성능을 벤치마킹합니다. 이 데이터 세트는 360×480 해상도로 367개의 훈련과 233개의 테스트 RGB 이미지(낮 및 해질녘 장면)로 구성됩니다. 도로, 건물, 자동차, 보행자, 표지판, 폴, 사이드워크 등 11개 클래스를 세분화하는 것이 과제다. RGB 입력에 대해 로컬 콘트라스트 정규화를 수행합니다.

The encoder and decoder weights were all initialized using the technique described in He et al. To train all the variants we use stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and momentum of 0.9 using our Caffe implementation of SegNet-Basic. We train the variants until the training loss converges. Before each epoch, the training set is shuffled and each mini-batch (12 images) is then picked in order thus ensuring that each image is used only once in an epoch. We select the model which performs highest on a validation dataset.

인코더와 디코더 무게는 모두 He 등에 기술된 기술을 사용하여 초기화되었다. 모든 변형을 훈련하기 위해 우리는 SegNet-Basic의 Caffe 구현을 사용하여 0.1의 고정 학습률과 0.9의 운동량을 가진 확률적 경사 강하(SGD)를 사용한다. 우리는 훈련 손실이 수렴될 때까지 변형을 훈련시킨다. 각 에폭 전에 트레이닝 세트를 쉐이핑하고 각 미니 배치(12 화상)를 선택해, 각 화상을 에폭에 1회만 사용하도록 한다. 검증 데이터 세트에서 가장 높은 성능을 발휘하는 모델을 선택합니다.

We use the cross-entropy loss as the objective function for training the network. The loss is summed up over all the pixels in a mini-batch. When there is large variation in the number of pixels in each class in the training set (e.g road, sky and building pixels dominate the CamVid dataset) then there is a need to weight the loss differently based on the true class. This is termed class balancing. We use median frequency balancing where the weight assigned to a class in the loss function is the ratio of the median of class frequencies computed on the entire training set divided by the class frequency. This implies that larger classes in the training set have a weight smaller than 1 and the weights of the smallest classes are the highest. We also experimented with training the different variants without class balancing or equivalently using natural frequency balancing.

우리는 교차 엔트로피 손실을 네트워크 훈련을 위한 목적 함수로 사용한다. 손실은 미니 배치의 모든 픽셀에 걸쳐 집계됩니다. 트레이닝 세트의 각 클래스의 픽셀 수에 큰 변동이 있는 경우(예: 도로와 건물 픽셀이 CamVid 데이터 세트를 지배함)에는 실제 클래스에 따라 손실의 가중치를 달리할 필요가 있습니다. 이를 클래스 밸런싱이라고 합니다. 손실 함수에서 클래스에 할당된 가중치가 전체 교육 세트에서 계산된 클래스 주파수의 중앙값을 클래스 주파수로 나눈 비율인 중앙 주파수 밸런싱을 사용한다. 즉, 교육 집합에서 큰 클래스의 가중치는 1보다 작고 가장 작은 클래스의 가중치는 가장 높다는 것을 의미합니다. 또한 클래스 밸런싱 없이 또는 자연 주파수 밸런싱을 사용하여 다양한 변형을 훈련하는 실험도 실시했습니다.

* 1. **Analysis**

To compare the quantitative performance of the different decoder variants, we use three commonly used performance measures: global accuracy (G) which measures the percentage of pixels7 correctly classified in the dataset, class average accuracy (C) is the mean of the predictive accuracy over all classes and mean intersection over union (mIoU) over all classes as used in the Pascal VOC12 challenge. The mIoU metric is a more stringent metric than class average accuracy since it penalizes false positive predictions. However, mIoU metric is not optimized for directly through the class balanced cross-entropy loss.

서로 다른 디코더 변형의 정량적 성능을 비교하기 위해 일반적으로 사용되는 세 가지 성능 척도를 사용한다. 데이터셋에서 올바르게 분류된 픽셀7의 백분율을 측정하는 글로벌 정확도(G), 클래스 평균 정확도(C)는 모든 클래스에 대한 예측 정확도 평균 및 MioU에 걸친 평균 교차점(MioU)이다. 모든 클래스가 Pascal VOC12 챌린지에 사용되었습니다. mIoU 메트릭은 잘못된 긍정 예측에 불이익을 주므로 클래스 평균 정확도보다 더 엄격한 메트릭입니다. 그러나 mIoU 메트릭은 클래스 균형 교차 엔트로피 손실을 통해 직접 최적화되지 않았습니다.

The mIoU metric otherwise known as the Jacard Index is most commonly used in benchmarking. However, Csurka et al. note that this metric does not always correspond to human qualitative judgements (ranks) of good quality segmentation. They show with examples that mIoU favours region smoothness and does not evaluate boundary accuracy, a point also alluded to recently by the authors of FCN. Hence they propose to complement the mIoU metric with a boundary measure based on the Berkeley contour matching score commonly used to evaluate unsupervised image segmentation quality. Csurka et al. simply extend this to semantic segmentation and show that the measure of semantic contour accuracy used in conjunction with the mIoU metric agrees more with human ranking of segmentation outputs.

Jacard Index로 알려진 mIoU 메트릭은 벤치마킹에서 가장 일반적으로 사용됩니다. 그러나 Csurka 등은 이 지표가 항상 양호한 품질 세분화에 대한 인간의 정성적 판단(순위)에 해당하는 것은 아니라고 지적한다. 이들은 예를 들어 mIoU가 지역 평활성을 선호하고 경계 정확도를 평가하지 않는다는 것을 보여주며, FCN의 저자들이 최근 언급하기도 했다. 따라서 이들은 감독되지 않은 영상 분할 품질을 평가하는 데 일반적으로 사용되는 버클리 등고선 일치 점수에 기초한 경계 측정으로 mIoU 메트릭을 보완할 것을 제안한다. Csurka 등은 단순히 이것을 의미 분할로 확장하고 mIoU 메트릭과 함께 사용되는 의미 등고선 정확도 측정이 분할 출력의 인간 순위와 더 일치함을 보여준다.

The key idea in computing a semantic contour score is to evaluate the F1-measure which involves computing the precision and recall values between the predicted and ground truth class boundary given a pixel tolerance distance. We used a value of 0.75% of the image diagonal as the tolerance distance. The F1-measure for each class that is present in the ground truth test image is averaged to produce an image F1-measure. Then we compute the whole test set average, denoted the boundary F1-measure (BF) by average the image F1 measures.

의미 등고선 점수 계산의 핵심 아이디어는 픽셀 공차 거리가 주어진 예측 및 지상 진실 등급 경계 사이의 정밀도와 호출 값을 계산하는 것을 포함하는 F1 측정을 평가하는 것이다. 공차 거리로 영상 대각선의 0.75% 값을 사용했습니다. 실측값 테스트 영상에 존재하는 클래스별 F1 측정값을 평균화하여 이미지 F1 측정값을 생성한다. 그런 다음 이미지 F1 측정 평균으로 경계 F1 측정(BF)을 나타내는 전체 테스트 세트 평균을 계산합니다.

We test each architectural variant after each 1000 iterations of optimization on the CamVid validation set until the training loss converges. With a training mini-batch size of 12 this corresponds to testing approximately every 33 epochs (passes) through the training set. We select the iteration wherein the global accuracy is highest amongst the evaluations on the validation set. We report all the three measures of performance at this point on the held-out CamVid test set. Although we use class balancing while training the variants, it is still important to achieve high global accuracy to result in an overall smooth segmentation. Another reason is that the contribution of segmentation towards autonomous driving is mainly for delineating classes such as roads, buildings, side-walk, sky. These classes dominate the majority of the pixels in an image and a high global accuracy corresponds to good segmentation of these important classes. We also observed that reporting the numerical performance when class average is highest can often correspond to low global accuracy indicating a perceptually noisy segmentation output.

CamVid 검증 세트에서 최적화를 1000회 반복한 후 훈련 손실이 수렴될 때까지 각 아키텍처 변형을 테스트합니다. 트레이닝 미니 배치 사이즈가 12인 경우 이는 트레이닝 세트를 통해 약 33에폭(합격)마다 테스트하는 것에 해당합니다. 검증 세트에 대한 평가 중에서 글로벌 정확도가 가장 높은 반복을 선택한다. 보류된 CamVid 테스트세트로 이 시점에서 퍼포먼스의 3가지 척도를 모두 보고합니다. 변종을 훈련하는 동안 클래스 밸런싱을 사용하지만 전체적인 원활한 분할을 위해서는 높은 글로벌 정확도를 달성하는 것이 여전히 중요합니다. 또 다른 이유는 자율 주행에 대한 세분화의 기여가 주로 도로, 건물, 옆길, 하늘과 같은 계층을 묘사하는 데 있다는 것이다. 이러한 클래스는 이미지 내의 픽셀의 대부분을 차지하고 있으며, 높은 글로벌 정밀도는 이러한 중요한 클래스의 양호한 분할에 해당합니다. 또한 클래스 평균이 가장 높을 때 수치 성능을 보고하는 것은 종종 지각적으로 노이즈가 많은 분할 출력을 나타내는 낮은 글로벌 정확도에 해당될 수 있다는 것을 관찰했다.

In Table 1 we report the numerical results of our analysis. We also show the size of the trainable parameters and the highest resolution feature map or pooling indices storage memory, i.e, of the first layer feature maps after max-pooling and sub-sampling. We show the average time for one forward pass with our Caffe implementation, averaged over 50 measurements using a 360 × 480 input on an NVIDIA Titan GPU with cuDNN v3 acceleration. We note that the upsampling layers in the SegNet variants are not optimised using cuDNN acceleration. We show the results for both testing and training for all the variants at the selected iteration. The results are also tabulated without class balancing (natural frequency) for training and testing accuracies. Below we analyse the results with class balancing.

표 1에서는 분석의 수치 결과를 보고한다. 또한 훈련 가능한 매개변수의 크기와 최대 풀링 및 하위 샘플링 후 첫 번째 레이어 피쳐 맵의 최고 해상도 피쳐 맵 또는 풀링 인덱스 저장 메모리도 보여준다. Caffe 구현으로 1회 전진 패스의 평균 시간을 보여주며, cuDNN v3 가속으로 NVIDIA Titan GPU에서 360 × 480 입력을 사용하여 50회 이상의 측정치를 평균했습니다. SegNet 변형에서 업샘플링 계층은 cuDNN 가속을 사용하여 최적화되지 않습니다. 선택한 반복에서 모든 변형에 대한 테스트 및 교육 결과를 보여 줍니다. 또한 훈련과 테스트의 정확성을 위해 클래스 밸런싱(자연 빈도)을 사용하지 않고 결과를 표로 작성합니다. 아래에서는 클래스 밸런싱을 사용하여 결과를 분석합니다.

From the Table 1, we see that bilinear interpolation based upsampling without any learning performs the worst based on all the measures of accuracy. All the other methods which either use learning for upsampling (FCN-Basic and variants) or learning decoder filters after upsampling (SegNet-Basic and its variants) perform significantly better. This emphasizes the need to learn decoders for segmentation. This is also supported by experimental evidence gathered by other authors when comparing FCN with SegNet-type decoding techniques.

표 1에서 학습이 없는 쌍선형 보간 기반 업샘플링은 모든 정확도 측정에 기초해 가장 나쁜 성능을 발휘한다는 것을 알 수 있다. 업샘플링을 위한 학습(FCN-Basic 및 변종) 또는 업샘플링 후 학습 디코더 필터(SegNet-Basic 및 그 변종)를 사용하는 다른 모든 방법은 훨씬 더 나은 성능을 발휘합니다. 이는 분할을 위해 디코더를 배울 필요가 있음을 강조합니다. 이는 FCN을 SegNet 유형의 디코딩 기술과 비교할 때 다른 저자가 수집한 실험 증거에서도 입증됩니다.

When we compare SegNet-Basic and FCN-Basic we see that both perform equally well on this test over all the measures of accuracy. The difference is that SegNet uses less **memory** during inference since it only stores max-pooling indices. On the other hand FCN-Basic stores **encoder feature maps** in full which consumes much more memory (11 times more). SegNet-Basic has a decoder with 64 feature maps in each decoder layer. In comparison FCN-Basic, which uses dimensionality reduction, has fewer (11) feature maps in each decoder layer. This reduces the number of convolutions in the decoder network and hence FCN-Basic is faster during inference (forward pass). From another perspective, the decoder network in SegNet-Basic makes it overall a larger network than FCN-Basic. This endows it with more flexibility and hence achieves higher training accuracy than FCN-Basic for the same number of iterations. Overall we see that SegNet-Basic has an advantage over FCN-Basic when inference time memory is constrained but where inference time can be compromised to some extent.

SegNet-Basic과 FCN-Basic을 비교하면 이 테스트에서는 모든 정밀도 측정에서 동일한 성능을 발휘합니다. 차이점은 SegNet은 최대 풀링 인덱스만 저장하기 때문에 추론 중에 사용하는 **메모리**가 적다는 것입니다. 한편 FCN-Basic에는 **인코더 기능 맵**이 완전히 저장되어 메모리 소비량이 11배 증가합니다. SegNet-Basic은 각 디코더 레이어에 64개의 기능 맵을 가진 디코더를 갖추고 있습니다. 이에 비해 FCN-Basic은 치수 축소를 사용합니다.각 디코더 레이어에서 기능 맵이 (11)개 적습니다. 이것에 의해, 디코더 네트워크내의 컨버전스수가 감소하기 때문에, FCN-Basic 는 추론(포워드 패스)중에 고속이 됩니다. 다른 관점에서 보면, SegNet-Basic의 디코더 네트워크는 전체적으로 FCN-Basic보다 더 큰 네트워크를 만듭니다. 이를 통해 유연성이 향상되어 동일한 횟수의 반복에 대해 FCN-Basic보다 높은 교육 정확도를 달성할 수 있습니다. 전반적으로 SegNet-Basic은 추론 시간 메모리가 제한되지만 추론 시간이 어느 정도 저하될 수 있는 경우 FCN-Basic보다 유리하다는 것을 알 수 있다.

SegNet-Basic is most similar to FCN-Basic-NoAddition in terms of their decoders, although the decoder of SegNet is larger. Both learn to produce dense feature maps, either directly by learning to perform deconvolution as in FCN-Basic-NoAddition or by first upsampling and then convolving with trained decoder filters. The performance of SegNet-Basic is superior, in part due to its larger decoder size. The accuracy of FCN-Basic-NoAddition is also lower as compared to FCN-Basic. This shows that it is vital to capture the information present in the encoder feature maps for better performance. In particular, note the large drop in the BF measure between these two variants. This can also explain the part of the reason why SegNet-Basic outperforms FCN-BasicNoAddition.

SegNet-Basic은 디코더 면에서 FCN-Basic-NoAddition과 가장 유사하지만 SegNet의 디코더는 더 큽니다. 둘 다 FCN-Basic-NoAddition에서와 같이 디콘볼루션 수행 방법을 직접 배우거나 먼저 업샘플링을 수행한 후 훈련된 디코더 필터를 사용하여 컨볼루션을 수행함으로써 고밀도 기능 맵을 생성하는 방법을 배웁니다. SegNet-Basic의 성능은 부분적으로 디코더 크기가 크기 때문에 우수합니다. FCN-Basic-NoAddition의 정확도도 FCN-Basic에 비해 낮습니다. 이는 성능을 향상시키기 위해 인코더 기능 맵에 있는 정보를 캡처하는 것이 중요하다는 것을 나타냅니다. 특히 이들 두 가지 변형 사이의 BF 측정치가 크게 떨어진 점에 유의하십시오. SegNet-Basic이 FCN-BasicNoAddition을 능가하는 이유도 여기에 있습니다.

The size of the FCN-Basic-NoAddition-NoDimReduction model is slightly larger than SegNet-Basic since the final encoder feature maps are not compressed to match the number of classes K. This makes it a fair comparison in terms of the size of the model. The performance of this FCN variant is poorer than SegNet-Basic in test but also its training accuracy is lower for the same number of training epochs. This shows that using a larger decoder is not enough but it is also important to capture encoder feature map information to learn better, particular the fine grained contour information (notice the drop in the BF measure). Here it is also interesting to see that SegNet-Basic has a competitive training accuracy when compared to larger models such FCNBasic-NoDimReduction.

FCN-Basic-NoAddition-NoDimReduction 모델의 크기는 SegNet-Basic보다 약간 큽니다.이는 최종 인코더 기능 맵이 클래스 K의 수와 일치하도록 압축되지 않기 때문입니다. 이를 통해 모델의 크기 면에서 공정한 비교를 할 수 있습니다. 이 FCN 변종의 성능은 테스트에서 SegNet-Basic보다 낮지만 같은 수의 트레이닝 에폭에서는 트레이닝 정확도가 낮습니다. 이것은, 보다 큰 디코더를 사용하는 것으로는 충분하지 않지만, 인코더 기능 맵 정보, 특히 세밀한 윤곽 정보(BF 측정값의 저하에 주의)를 취득하는 것도 중요하다는 것을 나타내고 있습니다. 여기서도 SegNet-Basic이 FCNBasic-NoDimReduction과 같은 대형 모델과 비교할 때 훈련의 정확도가 경쟁력이 있다는 것을 알 수 있습니다.

Another interesting comparison between FCN-BasicNoAddition and SegNet-Basic-SingleChannel Decoder shows that using max-pooling indices for upsampling and an overall larger decoder leads to better performance. This also lends evidence to SegNet being a good architecture for segmentation, particularly when there is a need to find a compromise between storage cost, accuracy versus inference time. In the best case, when both memory and inference time is not constrained, larger models such as FCN-Basic-NoDimReduction and SegNet-EncoderAddition are both more accurate than the other variants. Particularly, discarding dimensionality reduction in the FCN-Basic model leads to the best performance amongst the FCN-Basic variants with a high BF score. This once again emphasizes the trade-off involved between memory and accuracy in segmentation architectures.

FCN-BasicNoAddition과 SegNet-Basic-SingleChannelDecoder의 또 다른 흥미로운 비교에서는 업샘플링에 max-pooling 인덱스를 사용하고 전체적으로 큰 디코더를 사용하면 퍼포먼스가 향상됩니다. 이는 특히 스토리지 비용, 정확도 및 추론 시간 사이에서 타협점을 찾아야 하는 경우 SegNet이 세그먼트화에 적합한 아키텍처라는 증거이기도 합니다. 최적의 경우 메모리와 추론 시간이 모두 제한되지 않는 경우 FCN-Basic-NoDimReduction 및 SegNet-EncoderAddition과 같은 대형 모델이 다른 모델보다 정확합니다. 특히 FCN-Basic 모델에서 차원 축소를 폐기하면 BF 점수가 높은 FCN-Basic 변형 중 최고의 성능을 얻을 수 있습니다. 이는 세그먼트화 아키텍처에서 메모리와 정확성의 균형을 다시 한번 강조합니다.

The last two columns of Table 1 show the result when no class balancing is used (natural frequency). Here, we can observe that without weighting the results are poorer for all the variants, particularly for class average accuracy and mIoU metric. The global accuracy is the highest without weighting since the majority of the scene is dominated by sky, road and building pixels. Apart from this all the inference from the comparative analysis of variants holds true for natural frequency balancing too, including the trends for the BF measure. SegNetBasic performs as well as FCN-Basic and is better than the larger FCN-Basic-NoAddition-NoDimReduction. The bigger but less efficient models FCN-Basic-NoDimReduction and SegNetEncoderAddition perform better than the other variants.

표 1의 마지막 두 열은 클래스 밸런싱을 사용하지 않은 경우의 결과를 나타냅니다(자연 주파수). 여기서 가중치를 부여하지 않으면 모든 변형, 특히 클래스 평균 정확도와 mIoU 메트릭에서 결과가 더 좋지 않다는 것을 알 수 있습니다. 장면의 대부분이 하늘, 도로, 건물 픽셀로 구성되어 있기 때문에 가중치 없이 글로벌 정확도가 가장 높습니다. 이와는 별도로, BF 측정의 추세를 포함한 자연 주파수 균형에도 변종 비교 분석의 모든 추론이 적용된다. SegNetBasic은 FCN-Basic과 마찬가지로 뛰어난 성능을 발휘하며 더 큰 FCN-Basic-NoAddition-NoDimReduction보다 우수합니다. 더 크지만 효율성이 낮은 FCN-Basic-NoDimReduction 및 SegNetEncoderAddition 모델은 다른 모델보다 성능이 우수합니다.

We can now summarize the above analysis with the following general points.

1) The best performance is achieved when encoder feature maps are stored in full. This is reflected in the semantic contour delineation metric (BF) most clearly.

2) When memory during inference is constrained, then compressed forms of encoder feature maps (dimensionality reduction, max-pooling indices) can be stored and used with an appropriate decoder (e.g. SegNet type) to improve performance.

3) Larger decoders increase performance for a given encoder network.

이제 위의 분석을 다음과 같은 일반적인 점으로 요약할 수 있습니다.

1) 최고의 퍼포먼스는 인코더 기능 맵이 완전히 저장되어 있을 때 실현됩니다. 이는 의미 등고선 묘사 메트릭(BF)에 가장 명확하게 반영된다.

2) 추론 중 메모리가 제한되면 압축된 형태의 인코더 기능 맵(치수 감소, 최대 풀링 인덱스)을 적절한 디코더(예: SegNet 유형)와 함께 저장하고 사용할 수 있습니다.

3) 디코더가 클수록 특정 인코더 네트워크의 퍼포먼스가 향상됩니다.

1. **Benchmarking**

We quantify the performance of SegNet on two scene segmentation benchmarks using our Caffe implementation 3. The first task is road scene segmentation which is of current practical interest for various autonomous driving related problems. The second task is indoor scene segmentation which is of immediate interest to several augmented reality (AR) applications. The input RGB images for both tasks were 360 × 480.

우리는 Caffe 구현 3을 사용하여 두 장면 분할 벤치마크에서 SegNet의 성능을 정량화합니다. 첫 번째 작업은 다양한 자율 주행 관련 문제에 대해 현재 실질적으로 관심이 있는 도로 장면 분할입니다. 두 번째 작업은 여러 증강 현실(AR) 애플리케이션에서 즉시 관심 있는 실내 장면 분할이다. 두 작업 모두 입력 RGB 이미지는 360 × 480이었습니다.

We benchmarked SegNet against several other well adopted deep architectures for segmentation such as FCN, DeepLabLargFOV and DeconvNet. Our objective was to understand the performance of these architectures when trained end-to-end on the same datasets. To enable end-to-end training we added batch normalization layers after each convolutional layer. For DeepLab-LargeFOV, we changed the max pooling 3 stride to 1 to achieve a final predictive resolution of 45 × 60. We restricted the feature size in the fully connnected layers of DeconvNet to 1024 so as to enable training with the same batch size as other models. Here note that the authors of DeepLab-LargeFOV have also reported little loss in performance by reducing the size of the fully connected layers.

SegNet을 FCN, DeepLabLarg 등 세그먼트화를 위해 잘 채택된 다른 몇 가지 심층 아키텍처와 비교하여 벤치마킹했습니다.FOV 및 DeconvNet. 우리의 목표는 동일한 데이터셋에서 엔드 투 엔드로 교육을 받은 경우 이러한 아키텍처의 성능을 이해하는 것이었습니다. 엔드 투 엔드 훈련을 가능하게 하기 위해 각 컨볼루션 레이어 뒤에 배치 정규화 레이어를 추가했다. DeepLab-Large용FOV, 우리는 45 × 60의 최종 예측 분해능을 달성하기 위해 최대 풀링을 1로 변경했다. 다른 모델과 동일한 배치 크기의 훈련을 가능하게 하기 위해 DeconvNet의 완전 접속 레이어 기능 사이즈를 1024로 제한했습니다. 여기에서는 DeepLab-Large의 저자가또한 FOV는 완전히 연결된 레이어의 크기를 줄임으로써 성능 저하가 거의 보고되지 않았습니다.

In order to perform a controlled benchmark we used the same SGD solver with a fixed learning rate of 10−3 and momentum of 0.9. The optimization was performed for more than 100 epochs through the dataset until no further performance increase was observed. Dropout of 0.5 was added to the end of deeper convolutional layers in all models to prevent overfitting (see http://mi.eng.cam.ac.uk/projects/ segnet/tutorial.html for example caffe prototxt). For the road scenes which have 11 classes we used a mini-batch size of 5 and for indoor scenes with 37 classes we used a mini-batch size of 4.

제어된 벤치마크를 수행하기 위해 동일한 SGD 솔버를 사용하여 10-3의 고정 학습률과 0.9의 모멘텀을 얻었습니다. 최적화는 더 이상의 성능 향상이 관찰되지 않을 때까지 데이터셋을 통해 100년 이상 수행되었습니다. 과적합을 방지하기 위해 모든 모델에서 더 깊은 컨볼루션 층 끝에 0.5의 드롭아웃이 추가되었다(예: 카페 prototxt 참조). 11클래스 로드씬은 미니배치 사이즈 5를, 37클래스 실내씬은 미니배치 사이즈 4를 사용했습니다.

**4.1. Road Scene Segmentation**

A number of road scene datasets are available for semantic parsing. Of these we choose to benchmark SegNet using the CamVid dataset as it contains video sequences. This enables us to compare our proposed architecture with those which use motion and structure and video segments. We also combine to form an ensemble of 3433 images to train SegNet for an additional benchmark. For a web demo (see footnote 3) of road scene segmentation, we include the CamVid test set to this larger dataset. Here, we would like to note that another recent and independent segmentation benchmark on road scenes has been performed for SegNet and the other competing architectures used in this paper. However, the benchmark was not controlled, meaning that each architecture was trained with a separate recipe with varying input resolutions and sometimes with a validation set included. Therefore, we believe our more controlled benchmark can be used to complement their efforts.

의미 해석을 위해 여러 도로 장면 데이터 세트를 사용할 수 있습니다. 이 중 비디오 시퀀스가 포함되어 있기 때문에 CamVid 데이터 세트를 사용하여 SegNet을 벤치마킹할 것을 선택합니다. 이를 통해 제안된 아키텍처를 모션, 구조 및 비디오 세그먼트를 사용하는 아키텍처와 비교할 수 있습니다. 또한 3433개의 이미지의 앙상블을 형성하여 SegNet을 트레이닝하여 추가 벤치마크를 실시합니다. 도로 장면 세분화의 웹 데모(각주 3 참조)를 위해 이 더 큰 데이터 세트에 CamVid 테스트 세트를 포함합니다. 여기서는 SegNet 및 본 문서에서 사용된 기타 경쟁 아키텍처에 대해 도로 환경에 대한 또 다른 최근 독립적인 세그먼트화 벤치마크가 수행되었음을 주목합니다. 그러나 벤치마크는 제어되지 않았습니다. 즉, 각 아키텍처는 다양한 입력 분해능을 가진 별도의 레시피를 사용하여 교육되었으며, 때로는 검증 세트를 포함하기도 했습니다. 따라서 델의 관리 대상 벤치마크를 사용하여 고객의 노력을 보완할 수 있다고 생각합니다.

The qualitative comparisons of SegNet predictions with other deep architectures can be seen in Fig. 4. The qualitative results show the ability of the proposed architecture to segment smaller classes in road scenes while producing a smooth segmentation of the overall scene. Indeed, under the controlled benchmark setting, SegNet shows superior performance as compared to some of the larger models. 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 than with fixed bilinear upsampling. DeconvNet is the largest model and the most inefficient to train. Its predictions do not retain small classes.

SegNet 예측과 다른 심층 아키텍처의 정성적 비교는 그림 4에서 확인할 수 있습니다. 정성적 결과는 제안된 아키텍처가 도로 장면에서 더 작은 클래스를 분할하는 동시에 전체 장면을 부드럽게 분할할 수 있는 능력을 보여준다. 실제로 제어된 벤치마크 설정 하에서 SegNet은 일부 대형 모델에 비해 뛰어난 성능을 보여줍니다. DeepLab-대형FOV는 가장 효율적인 모델이며 CRF 후처리를 통해 더 작은 클래스가 손실되더라도 경쟁력 있는 결과를 얻을 수 있습니다. 학습된 디콘볼루션 FCN은 고정 쌍선형 업샘플링보다 확실히 우수합니다. DeconvNet은 가장 규모가 크고 교육 효율성이 가장 낮은 모델입니다. 그것의 예측은 작은 클래스를 유지하지 않는다.

We also use this benchmark to first compare SegNet with several non deep-learning methods including Random Forests, Boosting in combination with CRF based methods. This was done to give the user a perspective of the improvements in accuracy that has been achieved using deep networks compared to classical feature engineering based techniques.

또한 이 벤치마크를 사용하여 먼저 SegNet을 CRF 기반 방법과 조합하여 랜덤 포레스트, 부스팅을 포함한 여러 비 딥 러닝 방법과 비교합니다. 이는 사용자가 딥 네트워크를 사용하여 달성한 기존의 기능 엔지니어링 기반 기술에 비해 정확도가 향상되었다는 관점을 제공하기 위해 수행되었습니다.

The results in Table 2 show SegNet-Basic, SegNet obtain competitive results when compared with methods which use CRFs. This shows the ability of the deep architecture to extract meaningful features from the input image and map it to accurate and smooth class segment labels. The most interesting result here is the large performance improvement in class average and mIOU metrics that is obtained when a large training dataset, obtained by combining [22], [26], [60], [61], is used to train SegNet. Correspondingly, the qualitative results of SegNet (see Fig. 4) are clearly superior to the rest of the methods. It is able to segment both small and large classes well. We remark here that we used median frequency class balancing in training SegNet-Basic and SegNet. In addition, there is an overall smooth quality of segmentation much like what is typically obtained with CRF post-processing. Although the fact that results improve with larger training sets is not surprising, the percentage improvement obtained using pre-trained encoder network and this training set indicates that this architecture can potentially be deployed for practical applications. Our random testing on urban and highway images from the internet (see Fig. 1) demonstrates that SegNet can absorb a large training set and generalize well to unseen images. It also indicates the contribution of the prior (CRF) can be lessened when sufficient amount of training data is made available.

표 2의 결과에 따르면 SegNet-Basic은 CRF를 사용하는 방법과 비교하여 경쟁력 있는 결과를 얻을 수 있습니다. 이는 입력 이미지에서 의미 있는 기능을 추출하여 정확하고 부드러운 클래스 세그먼트 라벨에 매핑하는 딥 아키텍처의 기능을 보여줍니다. 여기서 가장 흥미로운 결과는 [22], [26], [60], [61]을 결합하여 얻은 대규모 훈련 데이터 세트를 SegNet 훈련에 사용할 때 얻을 수 있는 클래스 평균 및 mIOU 메트릭의 큰 성능 향상입니다. 따라서 SegNet의 정성적 결과(그림 4 참조)는 나머지 방법보다 확실히 우수하다. 그것은 작은 학급과 큰 학급 모두를 잘 나눌 수 있다. 여기에서 우리는 SegNet-Basic과 SegNet 훈련에서 중앙 주파수 클래스 밸런싱을 사용했음을 언급합니다. 또한 일반적으로 CRF 후처리를 통해 얻을 수 있는 것과 같이 전체적으로 부드러운 분할 품질이 있습니다. 대규모 트레이닝 세트에 의해 결과가 향상되는 것은 놀랄 일이 아니지만, 사전 트레이닝된 인코더 네트워크와 이 트레이닝 세트를 사용하여 얻을 수 있는 비율의 향상은 이 아키텍처가 실용적인 애플리케이션에 잠재적으로 도입될 수 있음을 나타냅니다. 인터넷에서 도시 및 고속도로 이미지에 대한 무작위 테스트(그림 1 참조)는 SegNet이 대규모 교육 세트를 흡수하고 보이지 않는 이미지로 잘 일반화할 수 있음을 보여준다. 또한 충분한 양의 훈련 데이터를 이용할 수 있을 때 이전(CRF)의 기여도를 낮출 수 있음을 나타낸다.

In Table 3 we compare SegNet’s performance with now widely adopted fully convolutional architectures for segmentation. As compared to the experiment in Table 2, we did not use any class blancing for training any of the deep architectures including SegNet. This is because we found it difficult to train larger models such as DeconvNet with median frequency balancing. We benchmark performance at 40K, 80K and >80K iterations which given the mini-batch size and training set size approximately corresponds to 50, 100 and >100 epochs. For the last test point we also report the maximum number of iterations (here atleast 150 epochs) beyond which we observed no accuracy improvements or when over-fitting set in. We report the metrics at three stages in the training phase to reveal how the metrics varied with training time, particularly for larger networks. This is important to understand if additional training time is justified when set against accuracy increases. Note also that for each evaluation we performed a complete run through the dataset to obtain batch norm statistics and then evaluated the test model with this statistic (see http://mi.eng.cam.ac.uk/projects/segnet/tutorial.html for code.). These evaluations are expensive to perform on large training sets and hence we only report metrics at three time points in the training phase.

표 3에서는 세그넷의 성능을 세그먼트화를 위해 널리 채택되고 있는 완전 컨볼루션 아키텍처와 비교합니다. 표 2의 실험과 비교하여 SegNet을 포함한 심층 아키텍처 훈련에는 클래스 블랭싱을 사용하지 않았습니다. 이는 중앙 주파수 밸런싱을 사용하여 DeconvNet과 같은 대형 모델을 교육하는 것이 어렵다는 것을 발견했기 때문입니다. 퍼포먼스를 40,000, 80,000, 80,000 이상의 반복으로 벤치마킹합니다.미니 배치 사이즈와 트레이닝 세트 사이즈는 약 50, 100, 100 에폭에 대응합니다. 마지막 테스트 포인트에 대해서도 정확도가 개선되지 않았거나 과적합이 설정되었을 때 관찰된 최대 반복 횟수(여기서는 최소 150에폭)를 보고한다. 트레이닝 단계의 3단계에서 메트릭을 보고하고 트레이닝 시간에 따라 메트릭이 어떻게 변화하는지 확인합니다.특히 대규모 네트워크에 대해서는요. 정확도가 증가하지 않도록 설정할 때 추가 교육 시간이 충분한지 이해하는 것이 중요합니다. 또한 각 평가에서 데이터 세트를 완전히 실행하여 배치 표준 통계를 얻은 다음 이 통계로 테스트 모델을 평가했습니다(코드에 대해서는 http://mi.eng.cam.ac.uk/projects/segnet/ tutorial.html 참조). 이러한 평가는 대규모 트레이닝 세트에서는 비용이 많이 들기 때문에 트레이닝 단계의 3가지 시점에서만 메트릭을 보고합니다.

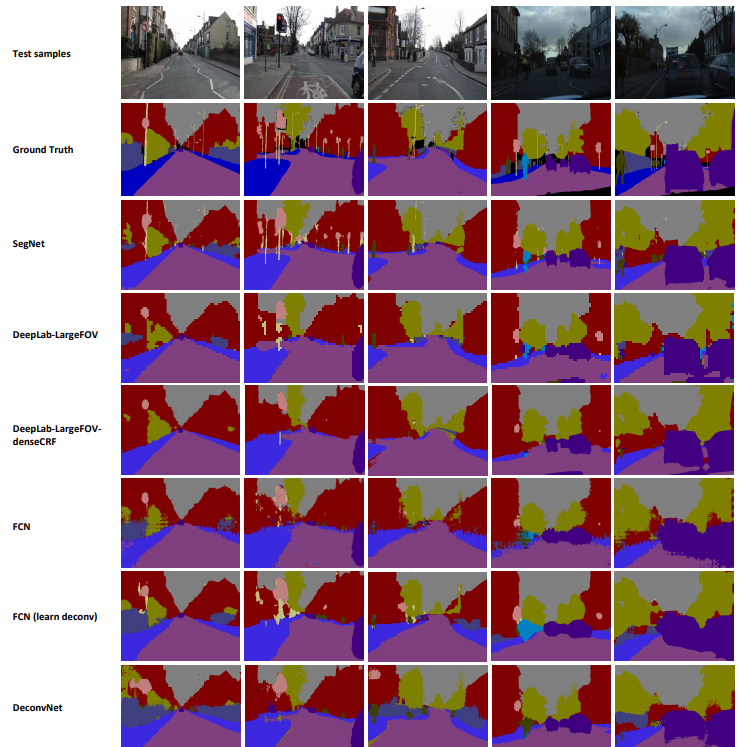


Fig. 4. Results on CamVid day and dusk test samples. SegNet shows superior performance, particularly with its ability to delineate boundaries, as compared to some of the larger models when all are trained in a controlled setting. DeepLab-LargeFOV is the most efficient model and with CRF post-processing 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.

그림 4. CamVid 주간 및 해질녘 시료에 대한 결과 SegNet은 특히 모든 것이 제어된 환경에서 훈련될 때 일부 대형 모델에 비해 경계를 기술하는 기능을 통해 우수한 성능을 보여줍니다. DeepLab-대형FOV는 가장 효율적인 모델이며 CRF 후처리를 통해 더 작은 클래스가 손실되더라도 경쟁력 있는 결과를 얻을 수 있습니다. 학습된 디콘볼루션 FCN이 확실히 낫다. DeconvNet은 가장 긴 교육 시간을 가진 가장 큰 모델이지만, 그 예측은 소규모 클래스를 느슨하게 한다. 이 결과는 표 3의 가장 높은 mIoU 정확도에 해당하는 모델에 해당합니다.

From Table 3 we immediately see that SegNet, DeconvNet achieve the highest scores in all the metrics as compared to other models. DeconvNet has a higher boundary delineation accuracy but SegNet is much more efficient as compared to DeconvNet. This can be seen from the compute statistics in Table 6. FCN, DeconvNet which have fully connected layers (turned into convolutional layers) train much more slowly and have comparable or higher forward-backward pass time with reference to SegNet. Here we note also that over-fitting was not an issue in training these larger models, since at comparable iterations to SegNet their metrics showed an increasing trend.

표 3에서 우리는 SegNet, DeconvNet이 다른 모델과 비교하여 모든 지표에서 가장 높은 점수를 획득했음을 즉시 알 수 있다. DeconvNet은 경계 묘사 정확도가 높지만, SegNet은 DeconvNet에 비해 훨씬 더 효율적입니다. 이는 표 6의 계산 통계에서 알 수 있습니다. FCN, DeconvNet은 완전히 연결된 계층(컨볼루션 계층으로 전환됨)의 트레이닝 속도가 훨씬 느리고 SegNet에 비해 포워드-백워드 패스 시간이 비슷하거나 더 높습니다. 또한 SegNet과 유사한 반복에서 측정 기준이 증가 추세를 나타냈기 때문에 이러한 대형 모델을 교육하는 데 있어 과적합은 문제가 되지 않았습니다.

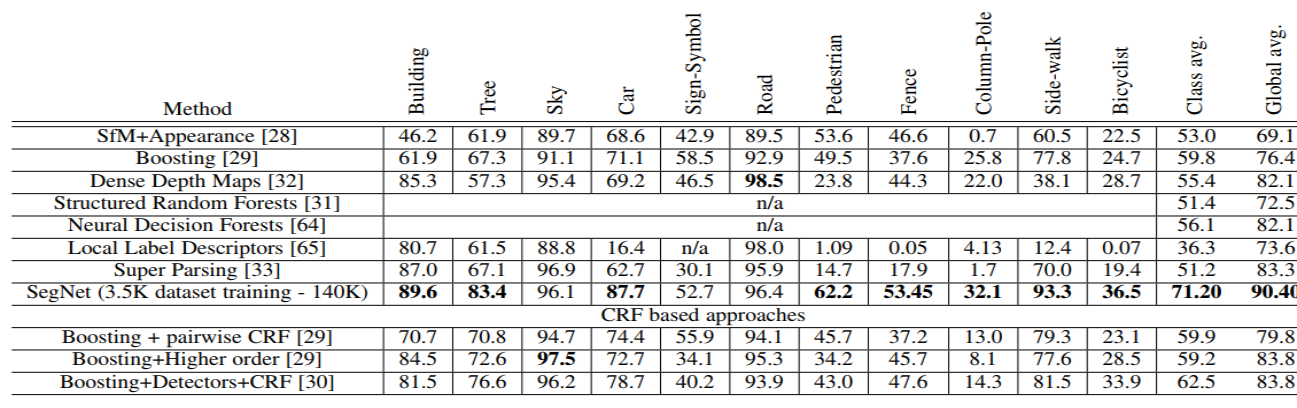


TABLE 2 Quantitative comparisons of SegNet with traditional methods on the CamVid 11 road class segmentation problem. SegNet outperforms all the other methods, including those using depth, video and/or CRF’s on the majority of classes. In comparison with the CRF based methods SegNet predictions are more accurate in 8 out of the 11 classes. It also shows a good ≈ 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.

표 2 CamVid 11 도로 등급 분할 문제에 대한 기존 방법과 SegNet의 정량적 비교. SegNet은 대부분의 클래스에서 깊이, 비디오 및/또는 CRF를 사용하는 방법을 포함하여 다른 모든 방법을 능가합니다. CRF 기반 메서드와 비교하여 SegNet 예측은 11개 클래스 중 8개에서 더 정확합니다. 또한 3.5K 이미지의 대규모 데이터 세트에 대해 교육을 받은 경우 클래스 평균 정확도가 10% 향상되었습니다. 특히 주목할 점은 더 작거나 더 얇은 클래스의 정확도가 크게 향상되었다는 것입니다. \* mIoU, BF 메트릭을 계산하기 위한 이전 방법의 예측에 액세스할 수 없었습니다.

For the FCN model learning the deconvolutional layers as opposed to fixing them with bi-linear interpolation weights improves performance particularly the BF score. It also achieves higher metrics in a far lesser time. This fact agrees with our earlier analysis in Sec. 3.3.

FCN 모델을 학습하는 경우, 2선형 보간 가중치로 고정하는 대신 디콘볼루션 레이어를 학습하는 것은 특히 BF 점수를 향상시킨다. 또한 훨씬 짧은 시간에 더 높은 메트릭을 달성할 수 있습니다. 이 사실은 3.3항의 이전 분석과 일치한다.

Surprisingly, DeepLab-LargeFOV which is trained to predict labels at a resolution of 45×60 produces competitive performance given that it is the smallest model in terms of parameterization and also has the fastest training time as per Table 6. However, the boundary accuracy is poorer and this is shared by the other architectures. DeconvNet’s BF score is higher than the other networks when trained for a very long time. Given our analysis in Sec. 3.3 and the fact that it shares a SegNet type architecture.

의외로 Deep Lab-Large는45×60 해상도의 라벨을 예측하도록 훈련받은 FOV는 매개변수화 측면에서 가장 작은 모델이며 표 6에 따라 가장 빠른 교육 시간을 가진다는 점에서 경쟁력 있는 성능을 발휘한다. 그러나 경계 정확도는 더 낮으며 이는 다른 아키텍처에 의해 공유됩니다. DeconvNet의 BF 점수는 매우 오랜 시간 훈련했을 때 다른 네트워크보다 높습니다. 3.3항의 분석과 SegNet 유형의 아키텍처를 공유한다는 사실을 고려할 때,

**4.2. Sun RGB-D Indoor Scenes**

SUN RGB-D is a very challenging and large dataset of indoor scenes with 5285 training and 5050 testing images. The images are captured by different sensors and hence come in various resolutions. The task is to segment 37 indoor scene classes including wall, floor, ceiling, table, chair, sofa etc. This task is made hard by the fact that object classes come in various shapes, sizes and in different poses. There are frequent partial occlusions since there are typically many different classes present in each of the test images. These factors make this one of the hardest segmentation challenges. We only use the RGB modality for our training and testing. Using the depth modality would necessitate architectural modifications/redesign. Also the quality of depth images from current cameras require careful post-processing to fill-in missing measurements. They may also require using fusion of many frames to robustly extract features for segmentation. Therefore we believe using depth for segmentation merits a separate body of work which is not in the scope of this paper. We also note that an earlier benchmark dataset NYUv2 is included as part of this dataset.

SUN RGB-D는 5285개의 트레이닝과 5050개의 테스트 이미지를 갖춘 매우 도전적이고 대규모 실내 장면 데이터 세트입니다. 이미지는 다른 센서에 의해 캡처되므로 다양한 해상도로 표시됩니다. 작업은 벽, 바닥, 천장, 테이블, 의자, 소파 등을 포함한 37개의 실내 장면 클래스를 세그먼트화하는 것이다. 이 작업은 객체 클래스가 다양한 모양, 크기 및 포즈로 제공되기 때문에 어렵습니다. 일반적으로 각 테스트 영상에 여러 클래스가 존재하기 때문에 부분 폐색이 자주 발생합니다. 이러한 요소들은 이 문제를 가장 어려운 세분화 과제 중 하나로 만듭니다. 훈련과 테스트에는 RGB 모달리티만 사용합니다. 깊이 모달리티를 사용하려면 아키텍처 수정/재설정이 필요합니다. 또한 현재 카메라의 깊이 영상의 품질은 누락된 측정을 채우기 위해 신중한 후 처리가 필요합니다. 또한 분할을 위해 기능을 확실하게 추출하기 위해 많은 프레임의 융합을 사용해야 할 수도 있습니다. 따라서, 분할에 깊이를 사용하는 것은, 이 백서의 범위에 포함되지 않는 별개의 작업에 가치가 있다고 생각합니다. 또한 이전 벤치마크 데이터 집합인 NYUv2가 이 데이터 집합의 일부로 포함되어 있습니다.

Road scene images have limited variation, both in terms of the classes of interest and their spatial arrangements. When captured from a moving vehicle where the camera position is nearly always parallel to the road surface limiting variability in view points. This makes it easier for deep networks to learn to segment them robustly. In comparison, images of indoor scenes are more complex since the view points can vary a lot and there is less regularity in both the number of classes present in a scene and their spatial arrangement. Another difficulty is caused by the widely varying sizes of the object classes in the scene. Some test samples from the recent SUN RGB-D dataset are shown in Fig. 5. We observe some scenes with few large classes and some others with dense clutter (bottom row and right). The appearance (texture and shape) can also widely vary in indoor scenes. Therefore, we believe this is the hardest challenge for segmentation architectures and methods in computer vision. Other challenges, such as Pascal VOC12 salient object segmentation have occupied researchers more, but we believe indoor scene segmentation is more challenging and has more current practical applications such as in AR and robotics. To encourage more research in this direction we compared well known deep architectures on the large SUN RGB-D dataset.

도로 풍경 이미지는 관심 클래스 및 공간 배열 측면에서 제한된 변동성을 가집니다. 카메라 위치가 거의 항상 노면과 평행한 이동 차량에서 캡처될 경우 시야의 변동이 제한됩니다. 이것에 의해, 딥 네트워크는, 보다 간단하게 세그먼트화하는 것을 학습할 수 있습니다. 이에 비해 실내 장면의 이미지는 관점이 많이 다를 수 있고 장면에 존재하는 클래스의 수와 공간 배치 모두에서 규칙성이 낮기 때문에 더 복잡하다. 또 다른 어려움은 씬(scene)에서 객체 클래스의 크기가 매우 다양합니다. 최근 SUN RGB-D 데이터 집합의 일부 테스트 샘플은 그림 5와 같다. 큰 클래스가 적은 장면과 밀도가 높은 혼란이 있는 장면(아래 행과 오른쪽)을 관찰합니다. 실내 장면에서도 외관(질감 및 모양)이 크게 다를 수 있습니다. 따라서 이는 컴퓨터 비전의 세그먼트화 아키텍처와 방법에서 가장 어려운 과제라고 생각합니다. Pascal VOC12 돌출 객체 분할과 같은 다른 과제가 연구자들을 더 많이 점유하고 있지만[66] 실내 장면 분할이 더 어렵고 AR이나 로봇과 같은 현재 더 실용적인 응용 분야를 가지고 있다고 생각합니다. 이 방향에서 더 많은 연구를 장려하기 위해 대규모 SUN RGB-D 데이터 세트에 대해 잘 알려진 심층 아키텍처를 비교했습니다.

The qualitative results of SegNet on samples of indoor scenes of different types such as bedroom, living room, laboratory, meeting room, bathroom are shown in Fig. 5. We see that SegNet obtains reasonable predictions when the size of the classes are large under different view points. This is particularly interesting since the input modality is only RGB. RGB images are also useful to segment thinner structures such as the legs of chairs and tables, lamps which is difficult to achieve using depth images from currently available sensors. This can be seen from the results of SegNet, DeconvNet in Fig. 5. It is also useful to segment decorative objects such as paintings on the wall for AR tasks. However as compared to outdoor scenes the segmentation quality is clearly more noisy. The quality drops significantly when clutter is increased (see the result sample in the middle column).

침실, 거실, 실험실, 회의실, 욕실 등 다양한 유형의 실내 장면 샘플에 대한 SegNet의 정성적 결과는 그림 5와 같다. SegNet은 다양한 뷰 포인트에서 클래스의 크기가 클 때 합리적인 예측을 얻을 수 있음을 알 수 있습니다. 입력 양식이 RGB이기 때문에 특히 흥미롭다. RGB 이미지는 현재 사용 가능한 센서의 깊이 이미지를 사용하여 달성하기 어려운 의자와 테이블 다리, 램프와 같은 얇은 구조를 세그먼트화하는 데도 유용하다. 이는 그림 5의 SegNet, DeconvNet의 결과에서 확인할 수 있다. 또한 AR 작업을 위해 벽에 있는 그림과 같은 장식 객체를 분할하는 데도 유용합니다. 그러나 야외 장면에 비해 분할 품질은 확실히 노이즈가 더 많습니다. 잡동사니가 증가하면 품질이 크게 떨어집니다(가운데 열의 결과 샘플 참조).

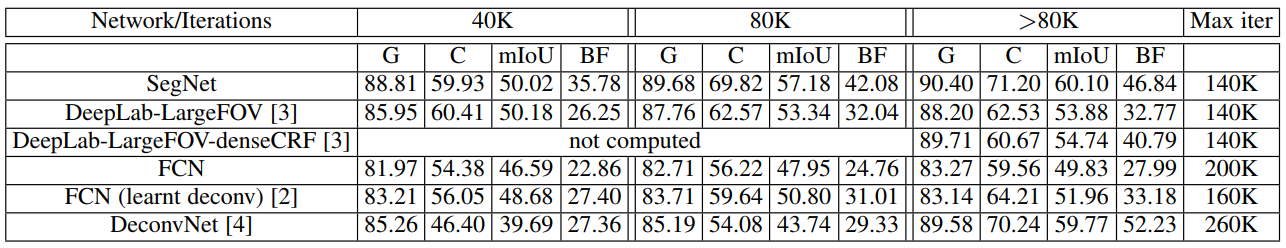


TABLE 3 Quantitative comparison of deep networks for semantic segmentation on the CamVid test set when trained on a corpus of 3433 road scenes without class balancing. When end-to-end training is performed with the same and fixed learning rate, smaller networks like SegNet learn to perform better in 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.

표 3 클래스 밸런싱 없이 3433개의 도로 장면의 말뭉치에 대해 교육을 받은 경우 CamVid 테스트 세트의 의미 분할을 위한 심층 네트워크의 정량적 비교. 동일한 고정 학습률로 엔드 투 엔드 트레이닝을 실시하면, SegNet 와 같은 소규모 네트워크는 단시간에 보다 뛰어난 퍼포먼스를 얻을 수 있습니다. 클래스 간 경계 설명의 정확도를 측정하는 BF 점수는 다른 경쟁 모델에 비해 SegNet, DeconvNet에서 상당히 높다. DeconvNet은 SegNet에 대한 메트릭과 일치하지만 계산 비용이 훨씬 높습니다. 또한 SegNet의 개별 클래스 정밀도에 대해서는 표2를 참조해 주십시오.

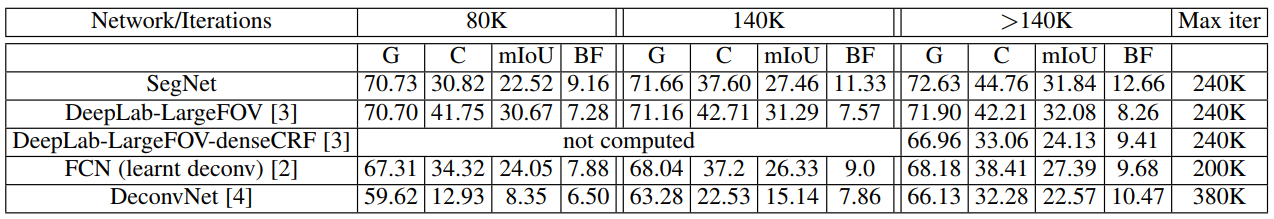


TABLE 4 Quantitative comparison of deep architectures on the SUNRGB-D dataset when trained on a corpus of 5250 indoor scenes. Note that only the RGB modality was used in these experiments. In this complex task with 37 classes all the architectures perform poorly, particularly because of the smaller sized 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.

표 4 5250개의 실내 장면 말뭉치에 대해 교육을 받은 경우 SUNRGB-D 데이터 세트의 심층 아키텍처의 정량적 비교. 이러한 실험에서는 RGB 형식만 사용되었습니다. 37개의 클래스가 있는 이 복잡한 태스크에서는 모든 아키텍처의 퍼포먼스가 저조합니다.특히 클래스 분산의 크기가 작고 스큐가 작기 때문입니다. DeepLab-Large FOV, 가장 작고 효율적인 모델은 mIoU가 약간 높지만 SegNet은 G, C, BF 점수가 더 높습니다. 또한 SegNet이 중앙 주파수 클래스 밸런싱을 통해 훈련되었을 때 71.75, 44.85, 32.08, 14.06 (180K)을 메트릭으로 얻었습니다.

The quantitative results in Table 4 show that all the deep architectures share low mIoU and boundary metrics. The global and class averages (correlates well with mIou) are also small. SegNet outperforms all other methods in terms of G,C, BF metrics and has a slightly lower mIoU than DeepLab-LargeFOV. As a stand alone experiment we trained SegNet with median frequency class balancing and the metrics were higher (see Table 4) and this agrees with our analysis in Sec. 3.3. Interestingly, using the grid search based optimal hyperparameters for the dense-CRF worsened all except the BF score metric for DeepLab-LargeFOVdenseCRF. More optimal settings could perhaps be found but the grid search process was too expensive given the large inference time for dense-CRFs.

표 4의 정량적 결과에 따르면 모든 심층 아키텍처는 낮은 mIoU 및 경계 메트릭을 공유합니다. 글로벌 및 클래스 평균(mIou와 잘 관련됨)도 작습니다. SegNet은 G, C, BF 메트릭 측면에서 다른 모든 방법을 능가하며 DeepLab-LargeFOV보다 mIoU가 약간 낮습니다. 독립형 실험으로서 우리는 중앙 주파수 클래스 밸런싱을 사용하여 SegNet을 교육했고 메트릭은 더 높았다(표 4 참조). 이는 3.3항의 분석과 일치한다. 흥미롭게도, DeepLab-Large에 대한 BF 점수 메트릭을 제외한 모든 것이 악화되었다.FOVdense CRF 보다 최적의 설정을 찾을 수 있을지는 모르지만, 고밀도 CRF에 대한 큰 추론 시간을 고려할 때 그리드 검색 프로세스는 너무 비쌌다.

One reason for the overall poor performance is the large number of classes in this segmentation task, many of which occupy a small part of the image and appear infrequently. The accuracies reported in Table 5 clearly show that larger classes have reasonable accuracy and smaller classes have lower accuracies. This can be improved with larger sized datasets and class distribution aware training techniques. Another reason for poor performance could lie in the inability of these deep architectures (all are based on the VGG architecture) to large variability in indoor scenes . This conjecture on our part is based on the fact that the smallest model DeepLab-LargeFOV produces the best accuracy in terms of mIoU and in comparison, larger parameterizations in DeconvNet, FCN did not improve perfomance even with much longer training (DeconvNet). This suggests there could lie a common reason for poor performance across all architectures. More controlled datasets are needed to verify this hypothesis.

전체적인 퍼포먼스가 저조한 이유 중 하나는 이 세그멘테이션 태스크의 클래스 수가 많기 때문입니다.이러한 클래스 중 상당수는 이미지의 작은 부분을 차지하고 있어 자주 표시되지 않습니다. 표 5에 보고된 정확도는 세분류가 클수록 정확도가 높고, 세분류가 작을수록 정확도가 낮다는 것을 명확히 보여준다. 대규모 데이터 세트와 클래스 분산 대응 교육 기술을 통해 이를 개선할 수 있습니다. 퍼포먼스 저하의 또 다른 이유는 이러한 깊은 아키텍처(모두 VGG 아키텍처에 근거하고 있다)가 실내 장면의 큰 변동에 대응할 수 없다는 것입니다. 이 추측은 가장 작은 DeepLab-Large 모델이FOV는 mIoU 측면에서 최고의 정확도를 제공하며, 그에 비해 DeconvNet FCN의 매개 변수화가 더 길어도 성능이 향상되지 않았다(DevonNet). 이는 모든 아키텍처에서 성능 저하가 발생하는 공통적인 이유가 있을 수 있음을 시사합니다. 이 가설을 검증하려면 보다 통제된 데이터 세트가 필요합니다.

1. **Discussion and Future Work**

Deep learning models have often achieved increasing success due to the availability of massive datasets and expanding model depth and parameterisation. However, in practice factors like memory and computational time during training and testing are important factors to consider when choosing a model from a large bank of models. Training time becomes an important consideration particularly when the performance gain is not commensurate with increased training time as shown in our experiments. Test time memory and computational load are important to deploy models on specialised embedded devices, for example, in AR applications. From an overall efficiency viewpoint, we feel less attention has been paid to smaller and more memory, time efficient models for real-time applications such as road scene understanding and AR. This was the primary motivation behind the proposal of SegNet, which is significantly smaller and faster than other competing architectures, but which we have shown to be efficient for tasks such as road scene understanding.

딥 러닝 모델은 대규모 데이터셋의 가용성과 모델 깊이 및 매개변수화 확대로 인해 종종 더 큰 성공을 거두었습니다. 그러나 실습에서는 대규모 모델 뱅크에서 모델을 선택할 때 훈련 및 테스트 중 메모리와 계산 시간과 같은 요소를 고려해야 합니다. 훈련 시간은 특히 우리의 실험에서 보여지듯이 성능 향상이 훈련 시간의 증가에 비례하지 않을 때 중요한 고려 사항이 됩니다. 테스트 시간 메모리와 계산 로드는 AR 애플리케이션 등 특수 임베디드 장치에 모델을 배치하는 데 중요합니다. 전체적인 효율의 관점에서 보면, 보다 작고, 보다 많은 메모리, 시간 효율이 뛰어난 모델, 예를 들면 도로 상황 이해나 AR에 대한 관심이 줄어들고 있는 것을 알 수 있습니다. 이는 다른 경쟁 아키텍처보다 훨씬 작고 빠르지만 도로 장면 이해와 같은 작업에 효율적이라는 것을 보여주는 SegNet의 제안의 이면에 있는 주요 동기였습니다.

Segmentation challenges such as Pascal and MS-COCO are object segmentation challenges wherein a few classes are present in any test image. Scene segmentation is more challenging due to the high variability of indoor scenes and a need to segment a larger number of classes simultaneously. The task of outdoor and indoor scene segmentation are also more practically oriented with current applications such as autonomous driving, robotics and AR.

Pascal 및 MS-COCO와 같은 분할 과제는 테스트 이미지에 몇 개의 클래스가 존재하는 객체 분할 과제입니다. 장면 분할은 실내 장면의 변동성이 크고 동시에 더 많은 클래스를 분할해야 하기 때문에 더 어렵습니다. 실외 및 실내 장면 분할 작업은 또한 자율 주행, 로봇 공학 및 AR과 같은 현재 애플리케이션에 더 실용적이다.

The metrics we chose to benchmark various deep segmentation architectures like the boundary F1-measure (BF) was done to complement the existing metrics which are more biased towards region accuracies. It is clear from our experiments and other independent benchmarks that outdoor scene images captured from a moving car are easier to segment and deep architectures perform robustly. We hope our experiments will encourage researchers to engage their attention towards the more challenging indoor scene segmentation task.

경계 F1 측정(BF)과 같은 다양한 심층 세분화 아키텍처를 벤치마킹하기 위해 선택한 메트릭은 지역 정확도에 더 치우친 기존 메트릭을 보완하기 위해 수행되었습니다. 우리의 실험과 다른 독립적인 벤치마크를 통해, 움직이는 자동차에서 캡처한 야외 장면 이미지는 더 쉽게 분할할 수 있고 심층 아키텍처는 강력한 성능을 발휘한다는 것이 명백합니다. 우리는 우리의 실험이 연구자들이 더 어려운 실내 장면 분할 작업에 관심을 기울이도록 격려하기를 바란다.

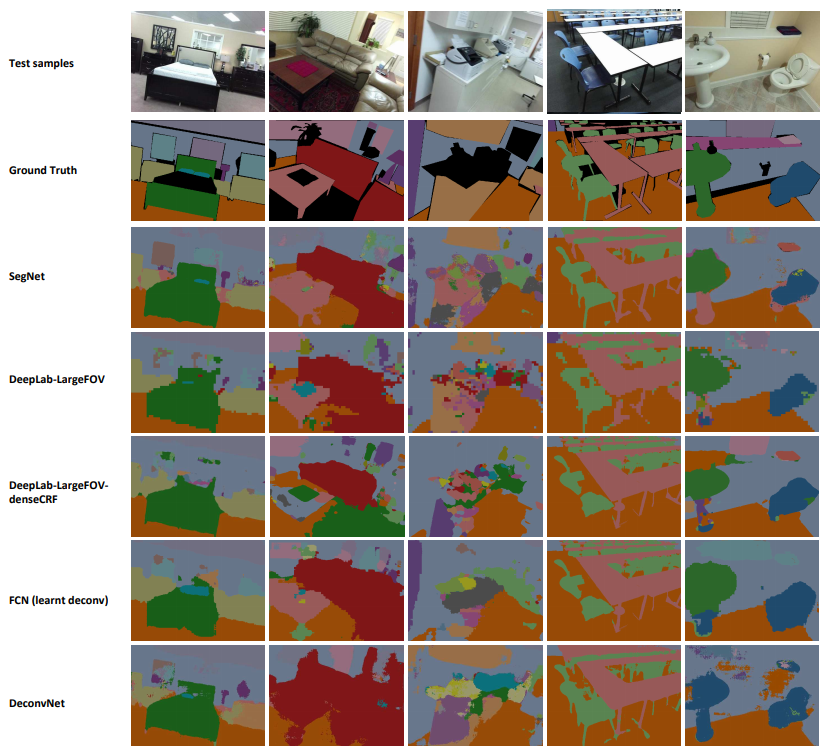


Fig. 5. Qualitative assessment of SegNet predictions on RGB indoor test scenes from the recently released SUN RGB-D dataset. In this hard challenge, SegNet predictions delineate inter class boundaries well for object classes in a variety of scenes and their view-points. Overall the segmentation quality 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.

그림 5. 최근 출시된 SUN RGB-D 데이터 세트에서 RGB 실내 테스트 장면에 대한 SegNet 예측의 정성적 평가. 이 어려운 과제에서 SegNet 예측은 다양한 장면에서 객체 클래스의 클래스 간 경계와 해당 뷰포인트를 잘 묘사합니다. 전반적으로 객체 클래스의 크기가 적당한 경우 분할 품질이 향상되지만 씬(scene)이 더 복잡할 경우 노이즈가 매우 큽니다. 씬(scene) 영상의 일부에는 접지 진실 라벨이 없고 검은색으로 표시되는 경우가 많습니다. 이러한 부품은 표시된 해당 심층 모델 예측에서 마스킹되지 않습니다. 이 결과는 표 4의 가장 높은 mIoU 정확도에 해당하는 모델에 해당합니다.

An important choice we had to make when benchmarking different deep architectures of varying parameterization was the manner in which to train them. Many of these architectures have used a host of supporting techniques and multi-stage training recipes to arrive at high accuracies on datasets but this makes it difficult to gather evidence about their true performance under time and memory constraints. Instead we chose to perform a controlled benchmarking where we used batch normalization to enable end-to-end training with the same solver (SGD). However, we note that this approach cannot entirely disentangle the effects of model versus solver (optimization) in achieving a particular result. This is mainly due to the fact that training these networks involves gradient back-propagation which is imperfect and the optimization is a non-convex problem in extremely large dimensions. Acknowledging these shortcomings, our hope is that this controlled analysis complements other benchmarks and reveals the practical trade-offs involved in different well known architectures.

다양한 매개 변수화의 다양한 심층 아키텍처를 벤치마킹할 때 우리가 내려야 했던 중요한 선택은 그러한 아키텍처를 훈련하는 방식이었다. 이러한 아키텍처의 대부분은 데이터셋에 대한 높은 정확도에 도달하기 위해 수많은 지원 기술 및 다단계 교육 레시피를 사용했지만, 이는 시간과 메모리 제약 조건 하에서 실제 성능에 대한 증거를 수집하기 어렵게 합니다. 대신 일괄 정규화를 사용하여 동일한 솔버(SGD)를 사용한 엔드 투 엔드 트레이닝을 가능하게 하는 제어 벤치마킹을 선택했습니다. 그러나, 우리는 이 접근방식이 특정 결과를 달성할 때 모델 대 솔버(최적화)의 효과를 완전히 분리할 수 없다는 점에 주목한다. 이것은 주로 이러한 네트워크의 트레이닝이 불완전한 구배 역전파를 수반하며 최적화는 매우 큰 차원에서 볼록하지 않은 문제이기 때문입니다. 이러한 단점을 인정하면서, 우리의 희망은 이 통제된 분석이 다른 벤치마크를 보완하고 서로 잘 알려진 아키텍처에 관련된 실질적인 트레이드오프를 드러내는 것이다.

For the future, we would like to exploit our understanding of segmentation architectures gathered from our analysis to design more efficient architectures for real-time applications. We are also interested in estimating the model uncertainty for predictions from deep segmentation architectures.

향후에는 분석을 통해 수집된 세그멘테이션 아키텍처에 대한 이해를 활용하여 실시간 애플리케이션을 위한 보다 효율적인 아키텍처를 설계하고 싶습니다. 또한 심층 세분화 아키텍처의 예측에 대한 모델 불확실성을 추정하는 데에도 관심이 있습니다.

1. **Conclusion**

We presented SegNet, a deep convolutional network architecture for semantic segmentation. The main motivation behind SegNet was the need to design an efficient architecture for road and indoor scene understanding which is efficient both in terms of memory and computational time. We analysed SegNet and compared it with other important variants to reveal the practical trade-offs involved in designing architectures for segmentation, particularly training time, memory versus accuracy. Those architectures which store the encoder network feature maps in full perform best but consume more memory during inference time. SegNet on the other hand is more efficient since it only stores the max-pooling indices of the feature maps and uses them in its decoder network to achieve good performance. On large and well known datasets SegNet performs competitively, achieving high scores for road scene understanding. End-to-end learning of deep segmentation architectures is a harder challenge and we hope to see more attention paid to this important problem.

우리는 의미 분할을 위한 심층 컨볼루션 네트워크 아키텍처인 SegNet을 제시했다. SegNet의 배후에 있는 주된 동기는 도로 및 실내 장면 이해를 위한 효율적인 아키텍처를 설계하는 것이었습니다.이 아키텍처는 메모리와 계산 시간 면에서 모두 효율적입니다. 세그먼트화를 위한 아키텍처 설계, 특히 훈련 시간, 메모리 대 정확도에 관련된 실질적인 트레이드오프를 밝히기 위해 SegNet을 분석하고 다른 중요한 변형과 비교했다. 인코더 네트워크피처 맵을 완전하게 격납하는 아키텍처는, 퍼포먼스가 최적이지만, 추론시에 메모리를 많이 소비합니다. 반면 SegNet은 기능 맵의 최대 풀링 인덱스만 저장하고 디코더 네트워크에서 이를 사용하여 성능을 향상시키므로 더 효율적입니다. 잘 알려진 대규모 데이터셋에서 SegNet은 경쟁적으로 성능을 발휘하여 현장 이해에 높은 점수를 획득합니다. 심층 세분화 아키텍처에 대한 엔드 투 엔드 학습은 더 어려운 과제이며, 이 중요한 문제에 더 많은 관심이 집중되기를 바랍니다.

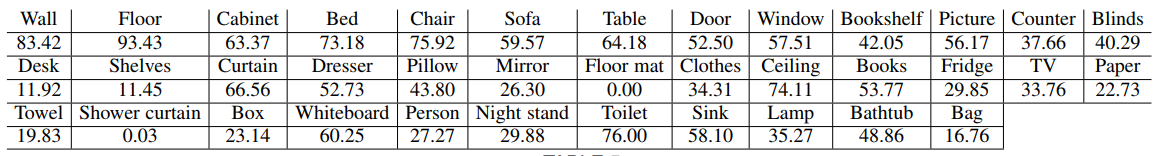


TABLE 5 Class average accuracies of SegNet predictions for the 37 indoor scene classes in the SUN RGB-D benchmark dataset. The performance correlates well with size of the classes in indoor scenes. Note that class average accuracy has a strong correlation with mIoU metric.

표 5 SUN RGB-D 벤치마크 데이터 세트의 37개 실내 장면 클래스에 대한 세그넷 예측의 클래스 평균 정확도. 실내기능은 실내 장면에서 수업의 크기와 잘 연관된다. 클래스 평균 정확도는 mIoU 메트릭과 강한 상관관계가 있습니다.

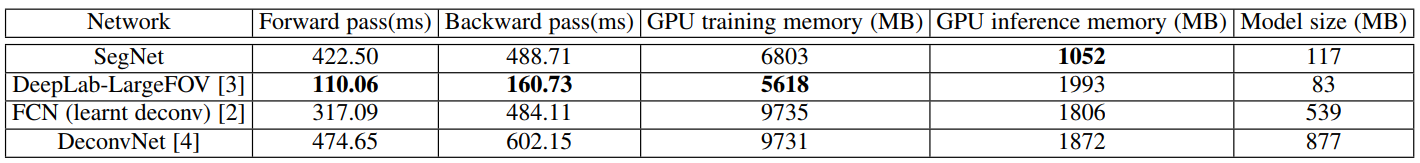


TABLE 6 A comparison of computational time and hardware resources required for various deep architectures. The caffe time command was used to compute time requirement averaged over 10 iterations with mini batch size 1 and an image of 360 × 480 resolution We used nvidia-smi unix command to compute memory consumption. For training memory computation we used a mini-batch of size 4 and for inference memory the batch size was 1. Model size was the size of the caffe models on disk. SegNet is most memory efficient during inference model.

표 6 다양한 심층 아키텍처에 필요한 계산 시간과 하드웨어 리소스 비교 caffe time 명령어는 미니 배치 크기 1과 360 × 480 해상도의 이미지를 사용하여 평균 10회 이상의 반복 시간을 계산하기 위해 사용되었습니다.메모리 소비량을 계산하기 위해 nvidia-smi unix 명령을 사용했습니다. 트레이닝 메모리 연산의 경우 크기가 4인 미니 배치를 사용했으며, 추론 메모리의 경우 배치 크기는 1이었다. 모델 크기는 디스크에 있는 카페 모델 크기입니다. SegNet은 추론 모델 중에 가장 메모리 효율적입니다.