Real-Time Semantic Segmentation Benchmarking Framework

Mennatullah Siam
University of Alberta
mennatul@ualberta.ca

Mostafa Gamal *
Cairo University
mostafa.gamal95@eng-st.cu.edu.eg

Moemen Abdel-Razek *
Cairo University
moemen.abdelrazek96@eng-st.cu.edu.eg

Senthil Yogamani
Valeo Vision Systems
senthil.yogamani@valeo.com

Abstract

Semantic segmentation has major benefits in autonomous driving and robotics related applications, where scene understanding is a necessity. Most of the research on semantic segmentation is focused on increasing the accuracy of segmentation models with few research on real-time performance. The few work conducted in this direction does not also provide principled methods to evaluate the different design choices for segmentation. In this paper, we address this gap by presenting the first real-time semantic segmentation benchmarking framework [2]. The framework is comprised of different network architectures for feature extraction such as VGG16, MobileNet, and ResNet-18. It is also comprised of multiple meta-architectures for segmentation that define the decoding methodology. These include Skip architecture, UNet, and Dilation Frontend. Experimental results on cityscapes with a case study using MobileNet architecture and two meta-architectures are presented.

1 Introduction

Semantic segmentation has widely progressed through the recent years with deep learning approaches. The first prominent work in this field was fully convolutional networks(FCNs)[11]. It proposed an end-to-end method to learn pixel-wise classification, where it used transposed convolution for upsampling. It also used skip architecture to refine the segmentation output. That method paved the road to subsequent advances in the segmentation accuracy. Multi-scale approaches[3][17], structured models[10][19], and spatio-temporal architectures[14][15] introduced different directions for improving the accuracy. Yu et. al.[17] presented the idea of dilated or Atrous convolution that can increase the receptive field without down-sampling. That was inspired from the important observation that segmentation unlike classification or detection tasks is greatly affected by the input resolution. Chen et. al.[3] later improved on the idea and introduced the DeepLab architecture that builds Atrous spatial pyramid pooling (ASPP). That can be used to segment objects at multiple scales, then followed it by conditional random fields as post processing. Zheng et. al.[19] formulated mean field approximation of conditional random fields as a recurrent network. Thus, he was able to train end-to-end for the segmentation, instead of using it as post processing. Siam et. al. incorporated convolutional gated recurrent units with FCNs to utilize temporal information. Convolutional gated recurrent units can work with feature maps instead of conventional recurrent gated units that work with flattened input.

*equally contributing

[2] `https://github.com/MSiam/TFSegmentation`

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

All of the above approaches focused on accuracy and robustness of segmentation. However, little attention is given to the efficiency of these networks. Although, when it comes to applications such as autonomous driving this would have tremendous impact. There exists some work that tries to address the segmentation networks efficiency such as [2][18][12]. Yet, there is no principled comparison of different architectures and meta-architectures that would enable researchers to pick the best suited network for the job. Chaurasia et. al.[2] presented the LinkNet architecture that is based on residual connections. Their method proved to be computationally efficient than other state of the art segmentation networks.

Huang et. al.[7] demonstrated principled comparison between accuracy and speed trade-offs for object detection. That inspired us to present the first benchmarking framework for multiple segmentation architectures. Instead of comparing standalone architectures, we compare the design choices in feature extraction part and decoders. This provides researchers with a tool to benchmark, analyze and to pick the best design per application. Our contribution lies in presenting the first benchmarking framework for segmentation architectures. On another perspective, we present a novel real-time segmentation network that is based on MobileNets[6]. It is able to beat the state of the art in computational performance, while maintaining relatively good accuracy. Our library is built on Tensorflow, and the code will be made publicly available. The paper is organized as follows, section2 details the benchmarking framework. Section3 discusses a case study on one of the design choices and its results, then section4 presents the concluding remarks.

2 Semantic Segmentation Framework

In this section first an overview of the framework is presented. Then different feature extraction architectures and meta-architectures are detailed.

2.1 Framework Overview

In order to create a benchmarking framework for segmentation, the main design choices to compare against have to be determined. Each model in our framework is represented by two main design decisions. The first is the network architecture that is used to extract features from the input. The other one is the meta-architecture, which denotes the decoding style of the segmentation framework. The decoding style is considered as a meta-information of the network. In order not to cause confusion, all models are trained end-to-end. This separation is only in the framework design, for the sake of extensibility. Thus, with new feature extractors or decoding styles it will provide an easier method to compare with different combinations from them.

2.2 Meta-Architectures

The meta-architectures for segmentation identify the decoding method to output the pixel-wise labels. All of the network architectures share the same down-sampling factor of 32. This is achieved either by utilizing 5 pooling layers, or by using strides in convolutional layers. This ensures that different meta architectures have a unified down-sampling factor to assess the effect of the decoding method alone. Three meta-architectures are integrated in our benchmarking software: (1) SkipNet meta-architecture[11]. (2) U-Net meta-architecture[13]. (3) Dilation frontend meta-architecture[17].
SkipNet architecture denotes a similar architecture to FCN8s [11]. The main idea of the skip architecture is to use the feature maps from earlier layers before pooling to increase the output resolution. Since all architecture have the same downsampling factor, it is possible that all of them follow the 8 stride version of skip architecture. In our design all feature extractors define three variables that are used for the skip architecture which are *feed1*, *feed2*, and *score* layers. *Feed1* is the layer before pool4, *feed2* is the layer before pool3, while *score* layer is the output from pool5. *Feed1* and *feed2* are followed by 1x1 convolution to produce heatmaps for each class. The *score* layer is followed by transposed convolution with stride 2, to sum with the higher resolution feature maps. Finally the output feature maps are followed by a transposed convolution for up-sampling with stride 8.

U-Net architecture denotes the method of decoding that up-samples features after each pooling layer using transposed convolution. This is then followed by concatenating the up-sampled features with the corresponding features maps from the encoder with the same resolution. The last upsampled

features are then followed by 1x1 convolution to output the final pixel-wise classification. The method in [2] uses ResNet-18 with a UNet meta architecture named LinkNet . Finally the **dilation frontend** architecture, removes the last two pooling layers. It then replaces them with two dilated convolutions[17] with dilation factor 2 and 4 respectively. In order to have equivalent receptive field to the original network with all pooling layers. Yet, dilated convolution does not hurt the resolution as pooling does. Figure 1 shows the different meta-architectures applied on MobileNet feature extractor.

2.3 Feature Extraction Architectures

In order to achieve real-time performance multiple network architectures is integrated in our framework. The framework includes three state of the art network architectures for feature extraction. These are: (1) VGG-16[16]. (2) ResNet-18[5]. (3) MobileNet[6]. The reason for picking VGG-16 is to act as a baseline, since FCN[11] is based on VGG-16. The other two architectures have been used in real-time systems. Thus, they would act as a starting point for benchmarking real-time segmentation. **VGG-16** is constructed of 5 pooling layers, and 16 convolution layers. **ResNet-18** has 1 convolutional layer, 8 residual blocks then one final convolutional layer. **MobileNet** network architecture is based on the idea of depthwise separable convolution. It is considered the extreme case of the inception module, where separate spatial convolutions for each channel is applied. This is followed by 1x1 convolution with all the channels to merge the output again.

3 Experiments

In this section detailed experimental results are discussed on one case study. The MobileNet feature extractor with two different meta-architectures are compared against the state of the art.

3.1 Experimental Setup

Experiments are conducted on images with size 512x1024, with 20 classes including the last class for the ignored class. A weighted cross entropy loss is used from [12], to overcome the imbalance in the data between different classes. The class weight is computed as $w_{class} = \frac{1}{ln(c+p_{class})}$, where c is a constant hyper-parameter with value 1.02. L2 regularization is used to avoid over-fitting with weight decay rate of $5e^{-4}$. Adam optimizer[9] is used with learning rate $1e^{-4}$. Batch normalization[8] is used after all convolutional or transposed convolution layers, to ensure faster convergence. The feature extractor part of the network is initialized with the pre-trained MobileNet[6] on Imagenet. It is worth noting that through all of the experiments we use width multiplier of 1 for MobileNet to use all the feature channels. In order to perform the benchmarking needed Cityscapes dataset[4] is used. It contains 5000 images with fine annotation, with 20 classes including the ignored class. The dataset is split into 2975 images for training, 500 for validation and 1525 for testing.

3.2 Semantic Segmentation Results

Semantic segmentation is evaluated using mean intersection over union (mIoU) and perclass IoU. Table2 and Table1 shows the results of Unet MobileNet and FCN8s MobileNet. It is shown that there is minimal differences between the two decoding methods in terms of overall mean IoU. Nonetheless, when looking into perclass IoU the best performing among them is UNet MobileNet. It provides better accuracy especially with smaller objects such as person or traffic sign to be segmented. Although, LinkNet[2] is beating UNet and FCN8s MobileNet in terms of accuracy, FCN8sMobileNet is beating it in terms of computational performance as shown in the next section. The results of the state of the art is reported from [2] for the sake of comparison on the validation set. Figure2 shows the qualitative results of the UNet version on Cityscapes.

Table 1: Quantitative comparison with perclass IoU on Cityscapes for Three meta-architecture UNet and FCN8s with MobileNet feature extraction part.

Architecture	Road	Sidewalk	Building	Traffic Sign	Vegetation	Person	Car
UNet MobileNet	**92.1**	**66.5**	**83.6**	**54.1**	**88.4**	**64.1**	**88.1**
FCN8s MobileNet	91.3	64.8	83.1	49.6	87.4	60	86.2

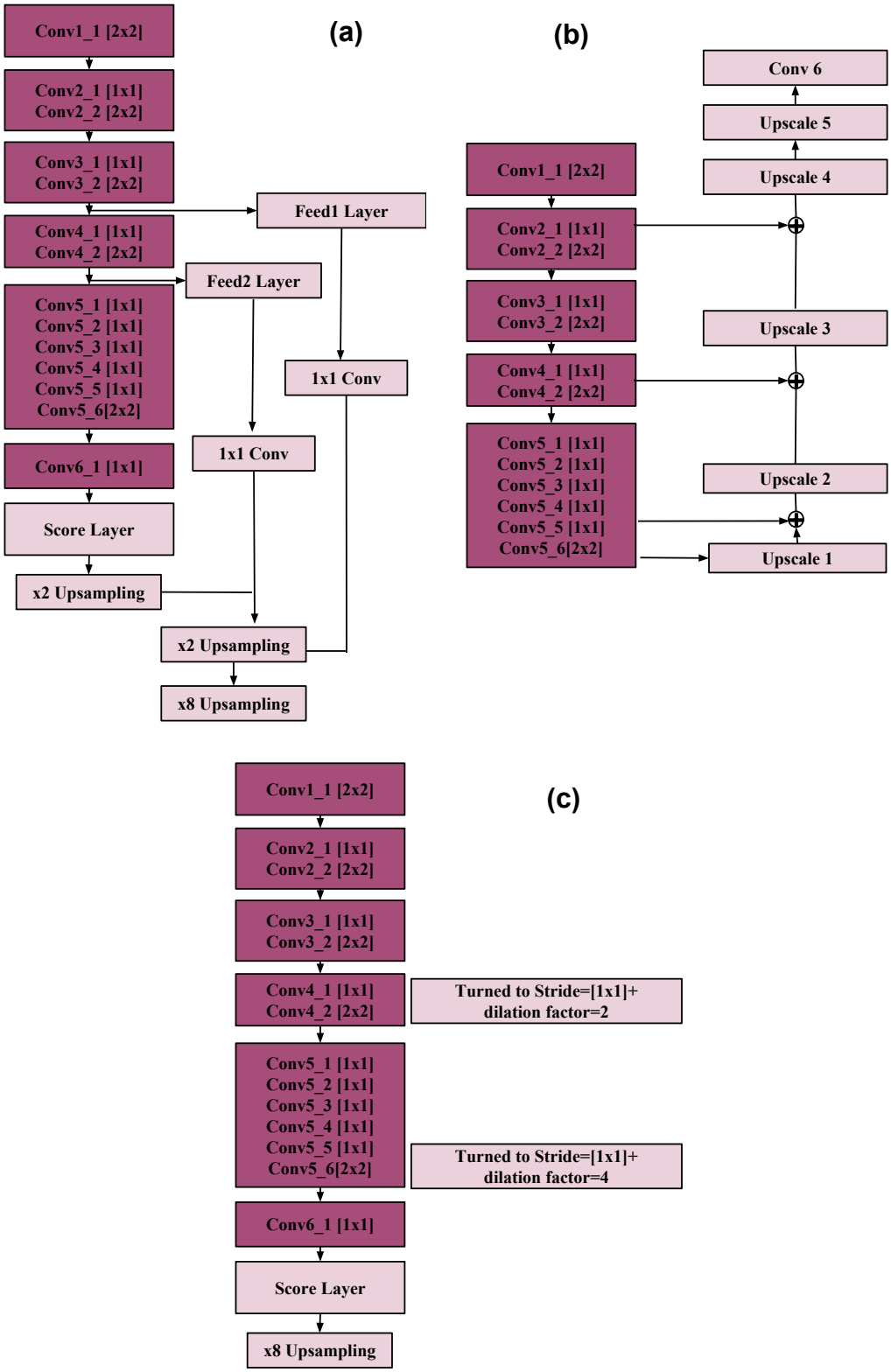

Figure 1: Different Meta Architectures using MobileNet as the feature extraction network. a) Skip Architecture termed as FCN8s. b) UNet. c) Dilation Frontend.

Table 2: Quantitative comparison on Cityscapes for Three meta-architecture UNet and FCN8s with MobileNet feature extraction part. Comparison against state of the art segmentation networks.

	UNetMobile	FCN8sMobile	Dilation10	DeepLab	LinkNet
mIoU	58.4	57	68.7	65.9	**76.4**

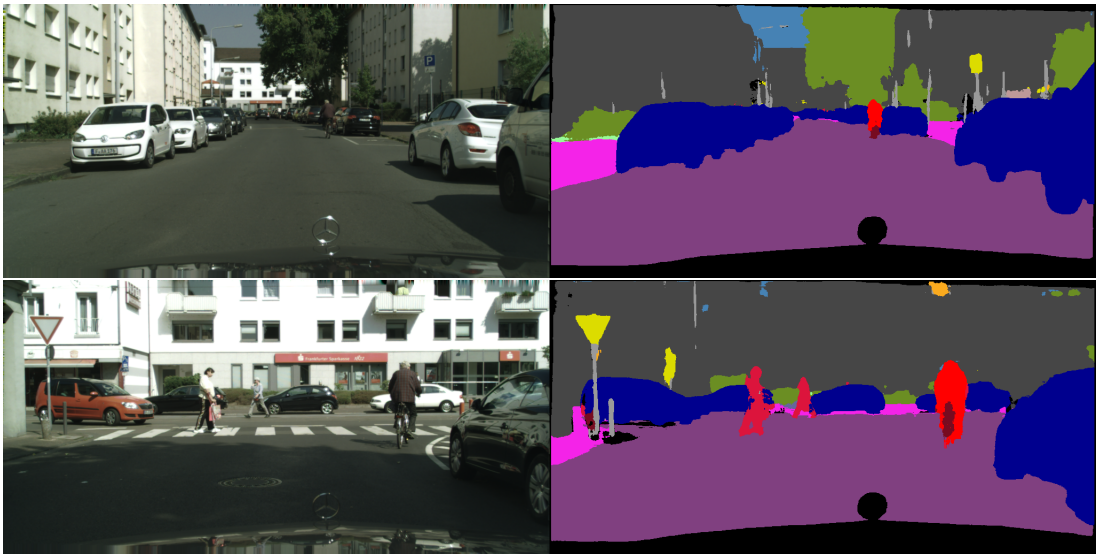

Figure 2: Qualitative evaluation on Cityscapes, Input image on the right and output labels from UNetMobileNet on the left.

3.3 Performance Analysis

In order to evaluate the computational performance of these networks as it is the main focus of the benchmarking framework. The metric used is GFLOPs that counts the floating points operation used. Table3 shows that FCN8sMobileNet that is the best performing model in MobileNet variants, is less by half than LinkNet the state of the art method. It is substantially better than SegNet[1] performance by a large margin. This shows that MobileNet might have potential benefits for real-time segmentation. Most importantly, through our benchmarking framework we provide a principled method to compare different design choices. This will help researchers better analyze and select the best network designs toward real-time semantic segmentation.

Table 3: Quantitative comparison between the best performing model FCN8sMobileNet and state of the art real-time segmentation networks in terms of GFLOPs.

	FCN8sMobile	SegNet[1]	LinkNet[2]
GFLOPs	**6.2**	286	21.2

4 Conclusion

In this paper we present the first principled approach for benchmarking real-time segmentation networks. It is based on dividing the design choices to separate modules for better quantitative comparison. The first module is comprised of the feature extraction network architecture, the second is the meta-architecture that provides the decoding method. Three different meta-architectures are included in our framework, including skip architecture, unet, and dilation frontend. At the same time, three different network architectures for feature extraction are included, which are MobileNet, VGG16, and ResNet-18. Results on the different versions of MobileNet with the three meta-architectures are presented on CityScapes benchmark. Accuracy and performance analysis on these versions are presented. This benchmarking framework provides researchers with a method to benchmark and choose the best design choices for real-time segmentation network.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[2] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. *arXiv preprint arXiv:1707.03718*, 2017.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[7] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[9] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian Reid. Exploring context with deep structured models for semantic segmentation. *arXiv preprint arXiv:1603.03183*, 2016.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[12] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[14] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *Computer Vision–ECCV 2016 Workshops*, pages 852–868. Springer, 2016.

[15] Mennatullah Siam, Sepehr Valipour, Martin Jagersand, and Nilanjan Ray. Convolutional gated recurrent networks for video segmentation. *arXiv preprint arXiv:1611.05435*, 2016.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[18] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. *arXiv preprint arXiv:1704.08545*, 2017.

[19] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.