RTSEG: REAL-TIME SEMANTIC SEGMENTATION COMPARATIVE STUDY

Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand

mennatul@ualberta.ca, senthil.yogamani@valeo.com
University of Alberta, Valeo Vision Systems, Cairo University

ABSTRACT

Semantic segmentation benefits robotics related applications, especially autonomous driving. Most of the research on semantic segmentation only focuses on increasing the accuracy of segmentation models with little attention to computationally efficient solutions. The few work conducted in this direction does not provide principled methods to evaluate the different design choices for segmentation. In this paper, we address this gap by presenting a real-time semantic segmentation benchmarking framework with a decoupled design for feature extraction and decoding methods. The framework is comprised of different network architectures for feature extraction such as VGG16, Resnet18, MobileNet, and ShuffleNet. It is also comprised of multiple meta-architectures for segmentation that define the decoding methodology. These include SkipNet, UNet, and Dilation Frontend. Experimental results are presented on the Cityscapes dataset for urban scenes. The modular design allows novel architectures to emerge, that lead to 143x GFLOPs reduction in comparison to SegNet. This benchmarking framework is publicly available at [1].

Index Terms**—** realtime; semantic segmentation; benchmarking framework

1. INTRODUCTION

Semantic segmentation has made progress in the recent years with deep learning. The first prominent work in this field was fully convolutional networks(FCNs) [1]. FCN was proposed as an end-to-end method to learn pixel-wise classification, where transposed convolution was used for upsampling. Skip architecture was used to refine the segmentation output, that utilized higher resolution feature maps. That method paved the road to subsequent advances in the segmentation accuracy. Multi-scale approaches [2, 3], structured models [4, 5], and spatio-temporal architectures [6] introduced different directions for improving accuracy. All of the above approaches focused on accuracy and robustness of segmentation. Well known benchmarks and datasets for semantic segmentation such as Pascal [7], NYU RGBD [8], Cityscapes [9], and Mapillary [10] boosted the competition toward improving accuracy.

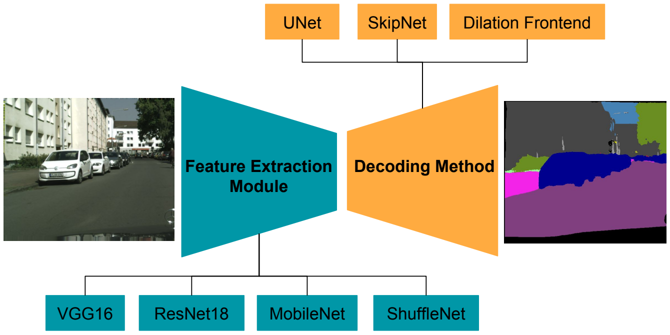
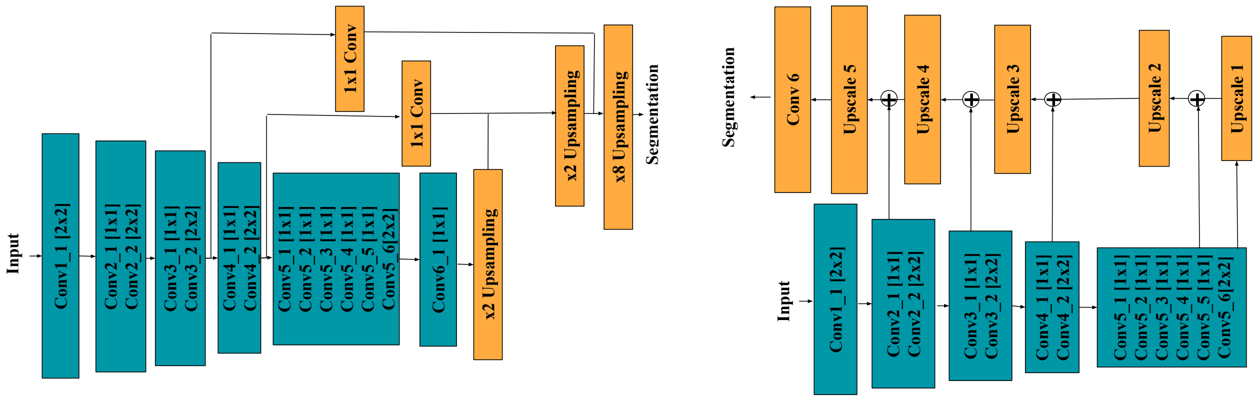


Fig. 1: Overview of the different components in the framework with the decoupling of feature extraction module and decoding method.

However, little attention is given to the computational efficiency of these networks. Although, when it comes to applications such as autonomous driving this would have tremendous impact. There exists few work that tries to address the segmentation networks efficiency such as [11, 12]. The survey on semantic segmentation [13] presented a comparative study between different segmentation architectures including ENet [12]. Yet, there is no principled comparison of different networks and meta-architectures. These previous studies compared different networks as a whole, without comparing the effect of different modules. That does not enable researchers and practitioners to pick the best suited design choices for the required task.

In this paper we propose the first framework toward benchmarking real-time architectures in segmentation. Our main contributions are: (1) we provide a modular decoupling of the segmentation architecture into feature extraction and decoding method which is termed as meta-architecture as shown in Figure 1. The separation helps in understanding the impact of different parts of the network on real-time performance. (2) A detailed ablation study with highlighting the trade-off between accuracy and computational efficiency is

[1] https://github.com/MSiam/TFSegmentation



(a) (b)

Fig. 2: Different Meta Architectures using MobileNet as the feature extraction network. a) SkipNet. b) UNet.

presented. (3) The modular design of our framework allowed the emergence of two novel segmentation architectures using MobileNet [14] and ShuffleNet [15] with multiple decoding methods. ShuffleNet lead to 143x GFLOPs reduction in comparison to SegNet. Our framework is built on top of Tensorflow and is publicly available.

2. BENCHMARKING FRAMEWORK

2.1. Meta-Architectures

Three meta-architectures are integrated in our benchmarking software: (1) SkipNet meta-architecture[1]. (2) U-Net meta-architecture[16]. (3) Dilation Frontend meta-architecture[3]. The meta-architectures for semantic segmentation identify the decoding method for in the network upsampling. All of the network architectures share the same down-sampling factor of 32. The downsampling is achieved either by utilizing pooling layers, or strides in the convolutional layers. This ensures that different meta architectures have a unified downsampling factor to assess the effect of the decoding method only.

SkipNet architecture denotes a similar architecture to FCN8s [1]. The main idea of the skip architecture is to benefit from feature maps from higher resolution to improve the output segmentation. SkipNet applies transposed convolution on heatmaps in the label space instead of performing it on feature space. This entails a more computationally efficient decoding method than others. Feature extraction networks have the same downsampling factor of 32, so they follow the 8 stride version of skip architecture. Higher resolution feature maps are followed by 1x1 convolution to map from feature space to label space that produces heatmaps corresponding to each class. The final heatmap with downsampling factor of 32 is followed by transposed convolution with stride 2. Elementwise addition between this upsampled heatmaps and the higher resolution heatmaps is performed. Finally, the output heat maps are followed by a transposed convolution for up-sampling with stride 8. Figure 2(a) shows the SkipNet architecture utilizing a MobileMet encoder.

U-Net architecture denotes the method of decoding that up-samples features using transposed convolution corresponding to each downsampling stage. The up-sampled features are fused with the corresponding features maps from the encoder with the same resolution. The stage-wise upsampling provides higher accuracy than one shot 8x upsampling. The current fusion method used in the framework is element-wise addition. Concatenation as a fusion method can provide better accuracy, as it enables the network to learn the weighted fusion of features. Nonetheless, it increases the computational cost, as it is directly affected by the number of channels. The upsampled features are then followed by 1x1 convolution to output the final pixel-wise classification. Figure 2(b) shows the UNet architecture using MobileNet as a feature extraction network.

Dilation Frontend architecture utilizes dilated convolution instead of downsampling the feature maps. Dilated convolution enables the network to maintain an adequate receptive field, but without degrading the resolution from pooling or strided convolution. However, a side-effect of this method is that computational cost increases, since the operations are performed on larger resolution feature maps. The encoder network is modified to incorporate a downsampling factor of 8 instead of 32. The decrease of the downsampling is performed by either removing pooling layers or converting stride 2 convolution to stride 1. The pooling or strided convolutions are then replaced with two dilated convolutions[3] with dilation factor 2 and 4 respectively.

2.2. Feature Extraction Architectures

In order to achieve real-time performance multiple network architectures are integrated in the benchmarking framework. The framework includes four state of the art real-time net-

Table 1: Comparison of different encoders and decoding methods in accuracy on cityscapes validation set. The modular decoupled design in RTSeg enabled such comparison. Coarse indicates whether the network was pre-trained on the coarse annotation or not.

Decoder	Encoder	Coarse	mIoU	Road	Sidewalk	Building	Sign	Sky	Person	Car	Bicycle	Truck
SkipNet	MobileNet	No	61.3	**95.9**	73.6	**86.9**	57.6	91.2	66.4	**89.0**	63.6	**45.9**
SkipNet	ShuffleNet	No	55.5	94.8	68.6	83.9	50.5	88.6	60.8	86.5	58.8	29.6
UNet	ResNet18	No	57.9	95.8	73.2	85.8	57.5	91.0	66.0	88.6	63.2	31.4
UNet	MobileNet	No	61.0	95.2	71.3	86.8	**60.9**	**92.8**	**68.1**	88.8	**65.0**	41.3
UNet	ShuffleNet	No	57.0	95.1	69.5	83.7	54.3	89.0	61.7	87.8	59.9	35.5
Dilation	MobileNet	No	57.8	95.6	72.3	85.9	57.0	91.4	64.9	87.8	62.8	26.3
Dilation	ShuffleNet	No	53.9	95.2	68.5	84.1	57.3	90.3	62.9	86.6	60.2	23.3
SkipNet	MobileNet	Yes	**62.4**	95.4	**73.9**	86.6	57.4	91.1	65.7	88.4	63.3	45.3
SkipNet	ShuffleNet	Yes	59.3	94.6	70.5	85.5	54.9	90.8	60.2	87.5	58.8	45.4

work architectures for feature extraction. These are: (1) VGG16[17]. (2) ResNet18[18]. (3) MobileNet[14]. (4) ShuffleNet [15]. The reason for using **VGG16** is to act as a baseline method to compare against as it was used in [1]. The other architectures have been used in real-time systems for detection and classification. **ResNet18** incorporates the usage of residual blocks that directs the network toward learning the residual representation on identity mapping.

MobileNet network architecture is based on depthwise separable convolution. It is considered the extreme case of the inception module, where separate spatial convolution for each channel is applied denoted as depthwise convolutions. Then 1x1 convolution with all the channels to merge the output denoted as pointwise convolutions is used. The separation in depthwise and pointwise convolution improve the computational efficiency on one hand. On the other hand it improves the accuracy as the cross channel and spatial correlations mapping are learned separately.

ShuffleNet encoder is based on grouped convolution that is a generalization of depthwise separable convolution. It uses channel shuffling to ensure the connectivity between input and output channels. This eliminates connectivity restrictions posed by the grouped convolutions.

3. EXPERIMENTS

In this section experimental setup, detailed ablation study and results in comparison to the state of the art are reported.

3.1. Experimental Setup

Through all of our experiments, weighted cross entropy loss from [12] is used, to overcome the class imbalance. Adam optimizer [19] learning rate is set to $1e^{-4}$. Batch normalization [20] is incorporated. L2 regularization with weight decay rate of $5e^{-4}$ is utilized to avoid over-fitting. The feature extractor part of the network is initialized with the pre-trained corresponding encoder trained on Imagenet. A width multiplier of 1 for MobileNet to include all the feature channels is performed through all the experiments. The number of groups used in ShuffleNet is 3. Based on previous [15] results on classification and detection three groups provided adequate accuracy.

Results are reported on Cityscapes dataset [9] which contains 5000 images with fine annotation, with 20 classes including the ignored class. Another section of the dataset contains coarse annotations with 20,000 labeled images. These are used in the case of Coarse pre-training that improves the results of the segmentation. Experiments are conducted on images with resolution of 512x1024.

Table 2: Comparison of the most promising models in our benchmarking framework in terms of GFLOPs and frames per second, this is computed on image resolution 512x1024.

Model	GFLOPs
SkipNet-MobileNet	13.8
UNet-MobileNet	55.9

3.2. Semantic Segmentation Results

Semantic segmentation is evaluated using mean intersection over union (mIoU), per-class IoU, and per-category IoU. Table1 shows the results for the ablation study on different encoders-decoders with mIoU and GFLOPs to demonstrate the accuracy and computations trade-off. The main insight gained from our experiments is that, UNet decoding method provides more accurate segmentation results than Dilation Frontend. This is mainly due to the transposed convolution by 8x in the end of the Dilation Frontend, unlike the UNet stage-wise upsampling method. The SkipNet architecture provides on par results with UNet decoding method. In some architectures such as SkipNet-ShuffleNet it is less accurate than UNet counter part by 1.5%.

The UNet method of incrementally upsampling with-in the network provides the best in terms of accuracy. However,

Table 3: Comparison of some of the models from our benchmarking framework with the state of the art segmentation networks on cityscapes test set. GFLOPs is computed on image resolution 360x640.

Model	GFLOPs	Class IoU	Class iIoU	Category IoU	Category iIoU
SegNet[21]	286.03	56.1	34.2	79.8	66.4
ENet[12]	3.83	58.3	24.4	80.4	64.0
DeepLab[2]	-	**70.4**	**42.6**	**86.4**	67.7
SkipNet-VGG16[1]	-	65.3	41.7	85.7	**70.1**
SkipNet-ShuffleNet	**2.0**	58.3	32.4	80.2	62.2
SkipNet-MobileNet	6.2	61.5	35.2	82.0	63.0

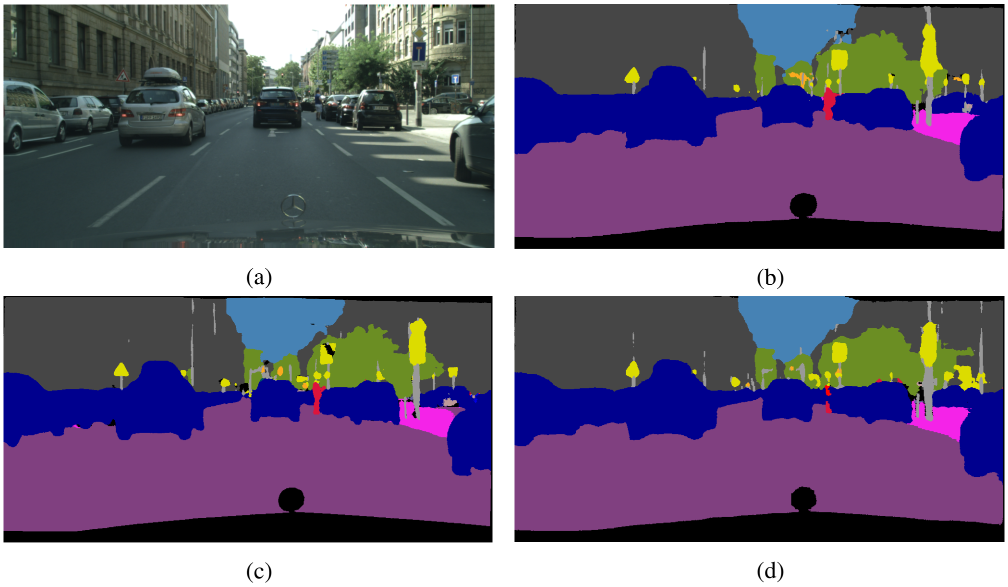


Fig. 3: Qualitative Results on CityScapes. (a) Original Image. (b) SkipNet-MobileNet pretrained with Coarse Annotations. (c) UNet-Resnet18. (d) SkipNet-ShuffleNet pretrained with Coarse Annotations.

Table 2 clearly shows that SkipNet architecture is more computationally efficient with 4x reduction in GFLOPs. This is explained by the fact that transposed convolutions in UNet are applied in the feature space unlike in SkipNet that are applied in label space. Table 1 shows that Coarse pre-training improves the overall mIoU with 1-4%. The underrepresented classes are the ones that often benefit from pre-training.

Experimental results on the cityscapes test set are shown in Table 3. Although, DeepLab provides best results in terms of accuracy, it is not computationally efficient. ENet [12] is compared to SkipNet-ShuffleNet and SkipNet-MobileNet in terms of accuracy and computational cost. SkipNet-ShuffleNet outperforms ENet in terms of GFLOPs, yet it maintains on par mIoU. Both SkipNet-ShuffleNet and SkipNet-MobileNet outperform SegNet [21] in terms of computational cost and accuracy with reduction up to 143x in GFLOPs. Figure 3 shows qualitative results for different encoders including MobileNet, ShuffleNet and ResNet18. It shows that MobileNet provides more accurate segmentation results than the later two. SkipNet-MobileNet is able to correctly segment the pedestrian and the signs on the right unlike the others.

4. CONCLUSION

In this paper we present the first principled approach for benchmarking real-time segmentation networks. The decoupled design of the framework separates 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. Different network architectures for feature extraction are included, which are ShuffleNet, MobileNet, VGG16, and ResNet-18. Our benchmarking framework provides researchers and practitioners with a mean to evaluate design choices for their tasks.

5. REFERENCES

[1] 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*, 2015, pp. 3431–3440.

[2] 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.

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

[4] 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.

[5] 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*, 2015, pp. 1529–1537.

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

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[8] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, "Indoor segmentation and support inference from rgbd images," *Computer Vision–ECCV 2012*, pp. 746–760, 2012.

[9] 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*, 2016, pp. 3213–3223.

[10] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy*, 2017, pp. 22–29.

[11] 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.

[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] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[14] 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.

[15] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *arXiv preprint arXiv:1707.01083*, 2017.

[16] 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*. Springer, 2015, pp. 234–241.

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

[18] 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*, 2016, pp. 770–778.

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

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

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