A Comparative Study of Real-time Semantic Segmentation for Autonomous Driving

Mennatullah Siam
University of Alberta
mennatul@ualberta.ca

Mostafa Gamal, Moemen Abdel-Razek
Cairo University
mostafa.gamal95@eng-st.cu.edu.eg

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

Martin Jagersand, Hong Zhang
University of Alberta
jag, hzhang@cs.ualberta.ca

Abstract

Semantic segmentation is a critical module in robotics related applications, especially autonomous driving. Most of the research on semantic segmentation is focused on improving the accuracy with less attention paid to computationally efficient solutions. Majority of the efficient semantic segmentation algorithms have customized optimizations without scalability and there is no systematic way to compare them. In this paper, we present a real-time segmentation benchmarking framework and study various segmentation algorithms for autonomous driving. We implemented a generic meta-architecture via a decoupled design where different types of encoders and decoders can be plugged in independently. We provide several example encoders including VGG16, Resnet18, MobileNet, and ShuffleNet and decoders including SkipNet, UNet and Dilation Frontend. The framework is scalable for addition of new encoders and decoders developed in the community for other vision tasks. We performed detailed experimental analysis on cityscapes dataset for various combinations of encoder and decoder. The modular framework enabled rapid prototyping of a custom efficient architecture which provides ∼x143 GFLOPs reduction compared to SegNet and runs real-time at ∼15 fps on NVIDIA Jetson TX2. The source code of the framework is publicly available [1].

1. Introduction

Semantic segmentation has witnessed tremendous progress with deep learning. The main goal is to perform pixel-wise classification of the image, that serves the purpose of scene understanding. Scene understanding has various benefits in robotics applications [55, 3, 56, 30], the most prominent benefit is in autonomous driving [61, 42, 4, 12]. Segmentation has also been used in medical applications [11, 66], and augmented reality [36]. The first prominent work in deep semantic segmentation was fully convolutional networks(FCNs) [35], that proposed an end-to-end method to learn pixel-wise classification. That method paved the road to subsequent advances in the segmentation accuracy. Multi-scale approaches [7][60], context aware models [33][65], and temporal models [44] introduced different directions for improving accuracy. All of the above approaches focused on accuracy and robustness of segmentation.

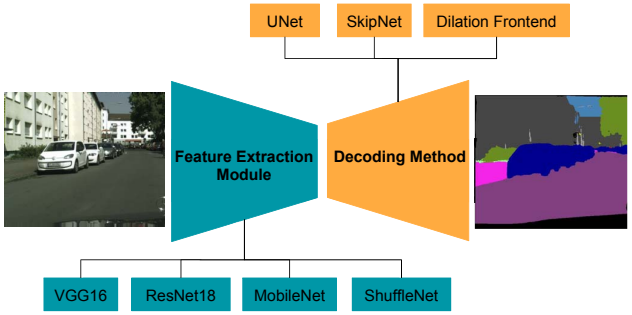


Figure 1: Overview of the different components in the framework with the decoupling of feature extraction module and decoding method.

However, some aspects for semantic segmentation such as computational efficiency has not been thoroughly studied in the literature. Although, when it comes to applications such as autonomous driving this would have tremendous impact. There is little work which address the seg-

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

2160-7516/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPRW.2018.00101

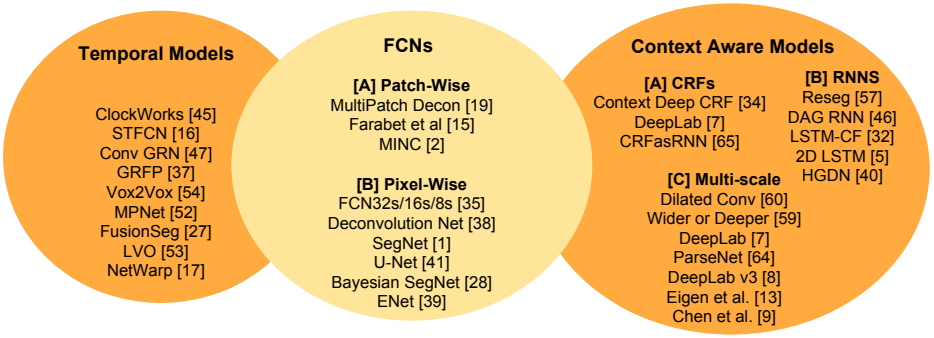


Figure 2: Taxonomy of semantic segmentation approaches.

mentation networks' efficiency such as [63][39]. The survey on semantic segmentation [18] presented a comparative study between different segmentation architectures including ENet [39]. 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 highlighting the trade-off between accuracy and computational efficiency is presented. (3) The modular design of our framework allowed the emergence of two novel segmentation architectures using MobileNet [24] and ShuffleNet [62] with multiple decoding methods. ShuffleNet lead to x143 GFLOPs reduction in comparison to SegNet. It was able to run real-time at 15 fps on a Jetson TX2. Our framework is built on top of Tensorflow and is publicly available.

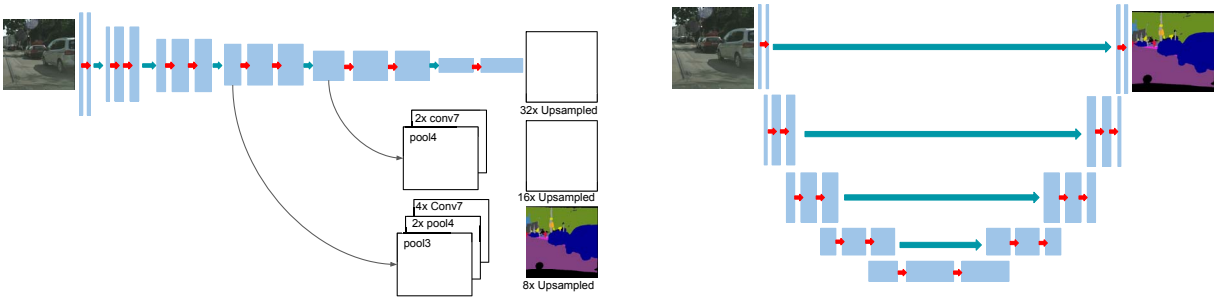
2. Semantic Segmentation

In this section a taxonomy of deep semantic segmentation is presented. The literature work in semantic segmentation is categorized into three main subcategories: **(1)** Fully Convolutional Networks. **(2)** Context Aware Models. **(3)** Temporal Models. The first category is about the main body of work on semantic segmentation using deep learning. The other two categories include the work exploiting context knowledge and temporal information. Note that both temporal and context aware models are considered under fully convolutional networks category. However they are considered as further refinement and are excluded in their own categories due to the large body of work under them. Figure 2 summarizes the general taxonomy and literature in semantic segmentation.

2.1. Fully Convolutional Networks(FCN)

The initial direction in semantic segmentation using convolutional neural networks was towards patch-wise training [14, 19, 2] to yield the final segmentation. Grangier et al. [19] proposed a multi-patch training strategy for convolutional neural networks to perform segmentation. Farabet et al. [14, 15] proposed a multi-scale dense feature extractor. The method used a Laplacian pyramid of the image, where each scale is forwarded through a 3-stage network to extract hierarchical features. For each pixel the features are encoded from a contextual patch around the pixel. The scene is then over-segmented into super pixels and conditional random fields over the super pixels are used. Bell et al. [2] proposed a method to utilize convolutional neural networks to classify each patch in a sliding window fashion.

The dominant direction in deep semantic segmentation is to learn pixel-wise classification in an end-to-end manner [35, 38, 1]. Long et al. [35] started with proposing fully convolutional networks(FCN). The network learned heatmaps that were then upsampled with-in the network using transposed convolution to get dense predictions. Unlike patch-wise training methods this method uses the full image to infer dense predictions. The SkipNet architecture was utilized to refine the segmentation using higher resolution feature maps. Noh et al. [38] proposed a deeper decoder network, in which stacked transposed convolution and unpooling layers are used. Badrinarayanan et al. [1] proposed SegNet which is an encoder-decoder architecture. The decoder network upsampled the feature maps by keeping the maxpooling indices from the corresponding encoder layer. Kendall et al. [28] followed that work by proposing Bayesian SegNet, which incorporates uncertainties in the predictions using dropout during inference. Ronneberger et al. [41] proposed a u-shaped architecture network where feature maps from different encoding layers are concate-



(a) SkipNet [35]

(b) U-Net [41]

Figure 3: Different Decoding methods for fully convolutional networks. Figure reproduced from [35, 41]

nated with the upsampled feature maps from the corresponding decoding layers. Paszke et al. [39] proposed the use of bottleneck modules for a computationally efficient solution that is denoted as ENet. Figure 3 shows the architecture for FCN8s [35] and U-Net [41].

2.2. Context Aware Models

Refinements on fully convolutional networks was introduced to improve the segmentation accuracy by incorporating context. In this section we consider only the spatial context that does not include any temporal information. The methods to enforce models to become context aware are mainly categorized into multi-scale support, utilizing conditional random fields, or recurrent neural networks. Farabet et al. [14] handled the scale by introducing multiple rescaled versions of the image to the network. However with the emergence of end-to-end pixel-wise training, Long et al. [35] proposed the skip architecture to merge heatmaps from different resolutions. Since these architectures include pooling layers to increase the receptive field, this leads to the downsampling of the image with a loss in the resolution.

Yu et al. [60] introduced dilated or atrous convolutions, which expanded the receptive field without losing resolution based on the dilation factor. Thus it provided a better solution for handling multiple scales. Wu et al. [59] proposed a shallower network using residual connections that included dilated convolution and outperformed deeper models. Chen et al. [7] proposed DeepLab that uses atrous spatial pyramid pooling (ASPP) for multi-scale support. This idea builds on utilizing the dilated convolutions. Figure 4 shows dilated convolutions and spatial pyramid pooling as separate methods that can be used to incorporate multi-scale support. Zhao et al. [64] proposed to incorporate global context features from previous layers into the next layers. Chen et al. [8] refined further the DeepLab method by incorporating global context features. Chen et al. [9] provided a way for handling scale by using attention models that provides a mean to focus on the most relevant features. This attention model is able to learn a weight map, that weighs feature maps pixel-by-pixel from different scales. Eigen et al. [13] proposed a method to sequentially utilize multiple scales to refine the prediction of depth, surface normals, and semantic segmentation.

One of the commonly used models to incorporate context is conditional random field (CRF). Chen et al. [7] utilized the fully connected conditional random fields as a post processing. The unary potentials of the CRF are set to the probabilities from their convolutional network, while pairwise potentials are gaussian kernels based on the spatial and color features. Lin et al [34] proposed a method to use pairwise potentials based on convolutional neural networks feature maps. In contrast to the previous work that uses conditional random fields as post processing refinement step, this work went further in integrating CNNs and CRFs. Zheng et al. [65] formulated the mean field CRF inference algorithm as a recurrent network. Thus, the proposed method enabled the end-to-end training of the model.

Another way to incorporate context is using recurrent neural networks (RNN) to capture the long range dependencies of various regions. Visin et al. [57] used a recurrent layer to sweep the image horizontally and vertically, which ensures the usage of contextual information for a better segmentation. One of the main bottlenecks in vanilla RNN is the vanishing gradients problem, gated recurrent architectures such as LSTMs [23] and GRUs [10] alleviate this problem. Byeon et al. [5] proposed a segmentation method that splits the image into non overlapping regions, then incorporates context using four separate LSTM blocks. Li et al. [32] proposed a method for context fusion using LSTMs. In their work both RGB and depth information were utilized, and the global context was modeled vertically on both, followed by the horizontal fusion. Another bottleneck in vanilla recurrent networks is that it could lead to the loss in spatial relationships. Shuai et al. [46] utilized directed acyclic graph RNN to incorporate long range dependencies. This directed acyclic graph maintains spatial relations-ships unlike using chained RNNs. Finally, Qi et

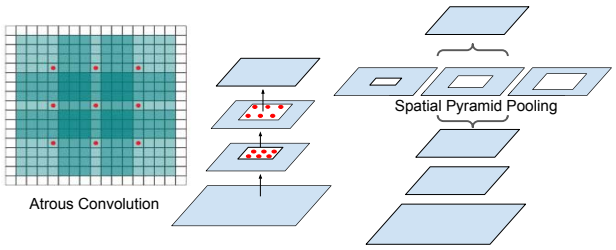


Figure 4: Atrous Convolution and Spatial Pyramid Pooling for Multi-scale support. Figure reproduced from [8, 60].

al [40] proposed hierarchically gated deep network, which is a multi-scale deep network that incorporates context at various scales. Multiple LSTM memory cells are used in the network between convolutional layers, to learn whether to incorporate spatial context from the lower layer into the higher one.

2.3. Temporal Models

All the discussed work was focused on still image segmentation. Recently some approaches emerged for video semantic segmentation that utilized temporal information [45][16] [47][37]. Shelhamer et al. [45] introduced clockwork networks which are clock signals that control the learning of different layers with different rates. Tran et al. [54] proposed a 3D convolutional network trained end-to-end for video semantic segmentation. An issue with 3D convolutional is its small extent on the temporal axis that would not capture long termporal dependencies. Recurrent neural networks can alleviate such a bottleneck. Fayyaz et al. [16] incorporated spatio temporal features by using a layer grid of Long Short term memory models (LSTMs).

However, conventional LSTMs as mentioned earlier do not utilize the spatial coherence and would end up with more parameters to learn. Siam et al. [47] proposed a convolutional gated recurrent network to learn temporal information to leverage the semantic segmentation of videos. The gated recurrent unit used in the work was convolutional, this enabled it to learn both spatial and temporal information with less number of parameters. Nilsson et al. [37] combined the power of both convolutional gated architectures and spatial transformers for leveraging video semantic segmentation. However in an action recognition comparative study [6], two-stream architectures that utilize optical flow information has shown to perform better than Conv-LSTM models. That motivated more research in the direction of incorporating motion and appearance for video segmentation. Tokmakov et al. [52] proposed a U-Net architecture that takes as input optical flow information to perform video segmentation. Jain et al. [27] proposed a model that fuses both RGB and optical flow information for the final video segmentation prediction. Tokmakov et al. [53] proposed a further improvement by utilizing optical flow information in a two-stream architecture that utilizes convolutional gated recurrent units. Gadde et al. [17] proposed a method for applying feature warping through an intermediate module termed as NetWarp in order to incorporate temporal information from videos.

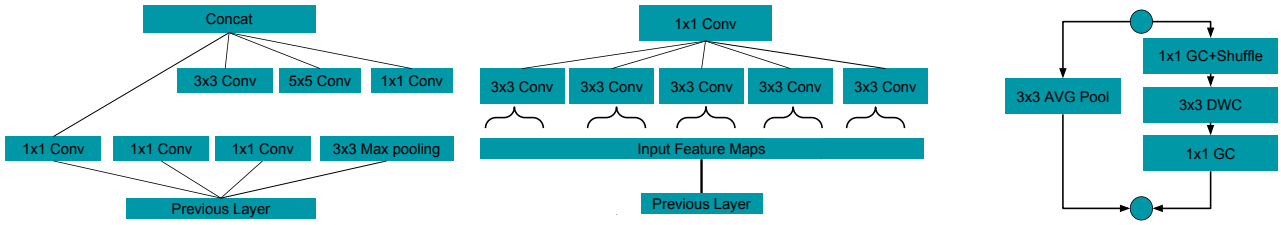
3. Real-time CNNs

In recent years there has been an increasing need for running deep neural networks real-time on embedded platforms, in various applications. Two main categories in the work of efficient CNNs are discussed: (1) Efficient CNN models that introduce different layers and modules to improve its computational efficiency. (2) Model compression and pruning. Other approaches such as model quantization and hardware acceleration are out of the scope of this paper.

3.1. Efficient CNN Models

Convolutional layers are required to learn cross channel and spatial correlations. This process can be performed in an efficient manner by separating both. Szegedy et al. [50, 51, 49] introduced the inception module and utilized it in Inception V1, V2 and further refined it in Inception V3 [51] and Inception-ResNet [49]. The main purpose of the inception module is to decouple the cross channel and spatial convolution. This separation is performed using 1x1 for the cross channel convolution that maps to 3 or 4 separate spaces. This is followed by 3x3 and/or 5x5 convolution for the spatial correlations. The extreme case of the inception module with one spatial convolution per channel is what is termed as depthwise separable convolution. Figure 5 shows the inception module [50], and depthwise separable convolution which is kind of an extreme case of inception. Howard et al. presented depth-wise separable convolutions as a mean to improve efficiency [24] in what is known as MobileNets. Zhang et al. developed a generalized form of separable convolution denoted as grouped convolution, while utilizing channel shuffle to ensure the input-output connectivity between different groups [62]. Figure 5 shows the shufflenet unit utilized in their model.

Huang et al. [25] proposed training a densely connected network with sparsified connections denoted as CondenseNet. The connectivity pattern is implemented efficiently using grouped convolutions. This method is considered also as a network pruning method. Most of the research conducted in efficient convolutional networks is directed towards classification and detection. Little attention is given to the computational efficiency of deep neural networks for semantic segmentation. When it comes to applications such as autonomous driving this consideration is extremely important. Some studies such as the work by Paszke et al. [39] tried to address the issue of segmentation efficiency.



(a) Inception Module [50] (b) Depthwise separable Convolution [24] (c) ShuffleNet Unit [62]

Figure 5: Differences between Computationally Efficient Modules for Convolution. GC: Grouped Convolution. DWC: Depth-Wise Convolution.

Sandler et al. [43] proposed inverted residual module with linear bottleneck. This takes low dimensional representation as input then expands it to a higher dimensional space applies convolution then maps it back. The convolution operation is performed using the efficient depthwise separable convolutions. This work proposed an efficient segmentation method as well.

3.2. Model Compression and Pruning

There are two main pruning techniques for model compression namely weight pruning and filter pruning. Han et al. proposed DeepCompression framework [21] which learns both weights and connections in a three steps process. They make use of a regularization loss which pushes parameters towards zero and thus reducing the number of parameters of AlexNet by a factor of 9. Sparsity can lead to inefficient parallelism. To alleviate sparsity constraint, Han et al. [20] presented an efficient inference engine relying on sparse matrix-vector multiplication with weight sharing. The resulting computation speed achieves x189 and x13 gain when compared to CPU and GPU implementations of the same DNN without compression. Model compression also enables networks to fit in the on-chip SRAM which reduces energy consumption per memory read by a factor x120 compared from fetching weights from DRAM.

Filter pruning is a similar approach like weight pruning. While weight pruning results in sparse connectivity pattern, removing the entire filter and their associated feature maps preserve dense connectivity. Consequently computational cost reduction does not rely on sparse convolution libraries or dedicated hardware and existing efficient BLAS libraries for dense matrix multiplication can be further used. Wen et al. [58] proposed filter pruning using model structure learning and group lasso which is an efficient regularization to learn sparse structures. Their method is even more general than filter regularization since the Structured Sparsity Learning (SSL) method can regularize any structure (filters, channels, filter shapes, and layer depth) of CNNs. This learning technique acts like a compression method to learn a smaller model from a larger one reducing the computational cost. Li et al. [31] presented another pruning approach which is not based on filter magnitude. The method relied on reinforcement Learning to train a pruning agent which made a set of binary actions to decide to remove or not each filter. It maximized a reward function which combined two terms, the accuracy term and the efficiency term. The accuracy term ensured the performance drop is bounded, and the efficiency term encouraged to prune more filters away.

4. Segmentation Benchmarking Framework

In this section a detailed description of the benchmarking framework is presented. We implemented a generic framework through the decoupled encoder-decoder design. This allows the extensibility for more encoding and decoding methods. It also allows principled comparison between different design choices that can aid practitioners.

4.1. Meta-Architectures

Three meta-architectures are integrated in our benchmarking software: (1) SkipNet meta-architecture [35]. (2) U-Net meta-architecture [41]. (3) Dilation Frontend meta-architecture [60]. 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 down-sampling factor to assess the effect of the decoding method only.

SkipNet architecture denotes a similar architecture to FCN8s [35]. 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 the 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

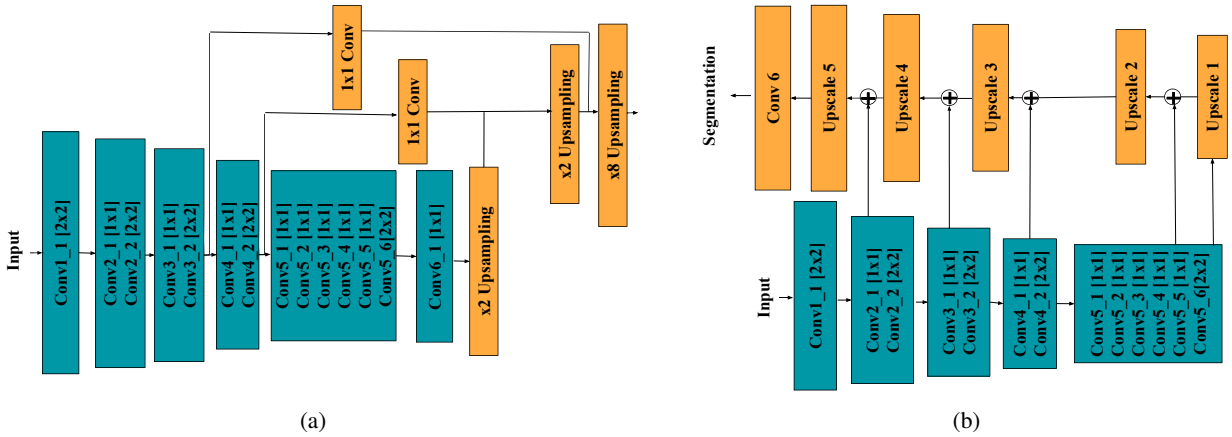


Figure 6: Different Meta Architectures using MobileNet as the feature extraction network. a) SkipNet architecture. b) UNet.

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 final output heat maps are followed by a transposed convolution for up-sampling with stride 8. Figure 6(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 [41]. The upsampled 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 6(b) shows the UNet architecture using MobileNet as a feature extraction network.

Dilation Frontend architecture utilizes dilated convolution [60] 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 [60] with dilation factor 2 and 4 respectively.

4.2. 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 network architectures for feature extraction. These are: (1) VGG16 [48]. (2) ResNet18 [22]. (3) MobileNet [24]. (4) ShuffleNet [62]. The reason for using **VGG16** is to act as a baseline method to compare against as it was used in [35]. 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 [24]. 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 is used and denoted as pointwise convolutions. 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 [62]. It uses channel shuffling to ensure the connectivity between input and output channels. This eliminates connectivity restrictions posed by the grouped convolutions.

5. Experiments

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

Table 1: Comparison of different encoders and decoders on **Cityscapes validation set**. GFLOPs are measured on image size **1024x512**.

Encoder	Decoder	GFLOPs	mIoU	Road	Sidewalk	Building	Sign	Sky	Person	Car
SkipNet	MobileNet	13.8	61.3	**95.9**	73.6	**86.9**	57.6	91.2	66.4	**89.0**
SkipNet	ShuffleNet	**4.63**	55.5	94.8	68.6	83.9	50.5	88.6	60.8	86.5
UNet	ResNet18	43.9	57.9	95.8	73.2	85.8	57.5	91.0	66.0	88.6
UNet	MobileNet	55.9	61.0	95.2	71.3	86.8	**60.9**	**92.8**	**68.1**	88.8
UNet	ShuffleNet	17.9	57.0	95.1	69.5	83.7	54.3	89.0	61.7	87.8
Dilation	MobileNet	150	57.8	95.6	72.3	85.9	57.0	91.4	64.9	87.8
Dilation	ShuffleNet	71.6	53.9	95.2	68.5	84.1	57.3	90.3	62.9	86.6

Table 2: Comparison of different encoders and decoders on **Cityscapes validation set** with Coarse annotations pre-training then using fine annotations.

Encoder	Decoder	mIoU	Road	Sidewalk	Building	Sign	Sky	Person	Car
SkipNet	MobileNet	**62.4**	**95.4**	**73.9**	**86.6**	**57.4**	**91.1**	**65.7**	**88.4**
SkipNet	ShuffleNet	59.3	94.6	70.5	85.5	54.9	90.8	60.2	87.5

5.1. Experimental Setup

Through all of our experiments, weighted cross entropy loss from [39] is used, to overcome the class imbalance. The class weight is computed as $w_{class} = \frac{1}{ln(c+p_{class})}$. Adam optimizer [29] learning rate is set to $1e^{-4}$. Batch normalization [26] after all convolutional or transposed convolution layers 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 experiments. The number of groups used in ShuffleNet is 3. Based on previous [62] results on classification and detection three groups provided adequate accuracy.

Results are reported on Cityscapes dataset [12] 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 proved to improve the results of the segmentation. Experiments are conducted on images with resolution of 512x1024.

5.2. Ablation Study

Semantic segmentation is evaluated using mean intersection over union (mIoU), per-class IoU, and per-category IoU. Table 1 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 x8 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%.

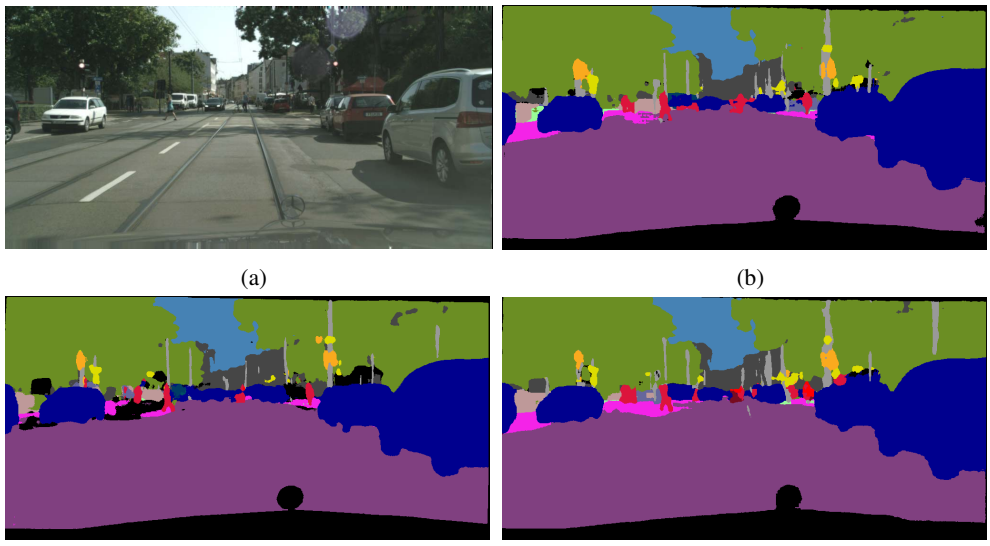
The UNet method of incrementally upsampling with-in the network provide the best in terms of accuracy. However, SkipNet architecture is more computationally efficient with x4 reduction in GFLOPs. This is explained by the fact that transposed convolutions in UNet are applied in the feature space unlike in SkipNet that are applied in label space. Table 2 shows that pre-training with cityscapes coarse annotation, then finetuning on the fine annotation improves the segmentation in terms of mIoU with 1-4%. The underrepresented classes are the ones that often benefit from pre-training.

5.3. Embedded Vision Experiments

Experimental results on the cityscapes test set are shown in Table 3. ENet [39] 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 [1] in terms of computational cost and accuracy with reduction up to x143 in GFLOPs. SkipNet-ShuffleNet was deployed on a Jetson TX2 that delivered real-time performance in 15 frames per second on image resolution 640x360. Figure 8 shows the comparison between different image resolution versus frame-rate and running time in milliseconds. These were measured on the Jetson TX2 for the SkipNet-ShuffleNet architecture. Figure 7 shows qualitative results for different encoders including MobileNet, ShuffleNet and ResNet18. It shows that MobileNet provides more accurate

Table 3: Comparison to the state of the art segmentation networks on **Cityscapes test set**. GFLOPs is computed on image resolution **640x360**.

Model	GFLOPs	Class IoU	Class iIoU	Category IoU	Category iIoU
SegNet[1]	286.03	56.1	34.2	79.8	66.4
ENet[39]	3.83	58.3	24.4	80.4	64.0
SkipNet-VGG16[35]	445.9	**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



(a) (b) (c) (d)

Figure 7: Qualitative Results on CityScapes. (a) Original Image. (b) SkipNet-MobileNet pretrained with Coarse Annotations. (c) UNet-Resnet18. (d) SkipNet-ShuffleNet pretrained with Coarse Annotations.

segmentation results than the later two.

Figure 8: Running Time in milliseconds and Frames per second versus the different image resolution. Measured on Jetson TX2.

6. Conclusion

In this paper, we present the first principled approach for benchmarking real-time segmentation networks. The decoupled design of the framework separates encoder and decoder modules and allows for systematic comparison. The first module is comprised of the feature extraction network architecture and the second module is the meta-architecture that provides the decoding method. This generic meta-architecture allows for extensibility further on to other encoders and decoding methods. Detailed analysis of different image resolutions versus frame-rate on Jetson TX2 is presented. Our benchmarking framework provides researchers and practitioners a mechanism to systematically evaluate new encoders and decoders. New computationally efficient models for segmentation emerged that outperform the state of the art in terms of GFLOPs, while maintaining on par accuracy. It enabled one of the models to run real-time at $\sim$16 fps on a Jetson TX2. Future work is to mathematically formalize the meta-architecture to enable automated topology exploration using meta-learning.

References

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

[2] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[3] T. M. Bonanni, A. Pennisi, D. Bloisi, L. Iocchi, and D. Nardi. Human-robot collaboration for semantic labeling of the environment. In *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration*, 2013.

[4] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[5] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3547–3555, 2015.

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[8] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[9] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *arXiv preprint arXiv:1511.03339*, 2015.

[10] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[11] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. 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.

[13] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[15] C. Farabet, N. EDU, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers.

[16] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, and R. Klette. STFCN: spatio-temporal FCN for semantic video segmentation. *CoRR*, abs/1608.05971, 2016.

[17] R. Gadde, V. Jampani, and P. V. Gehler. Semantic video cnns through representation warping. *CoRR, abs/1708.03088*, 2017.

[18] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[19] D. Grangier, L. Bottou, and R. Collobert. Deep convolutional networks for scene parsing. In *ICML 2009 Deep Learning Workshop*, volume 3. Citeseer, 2009.

[20] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally. Eie: efficient inference engine on compressed deep neural network. In *Proceedings of the 43rd International Symposium on Computer Architecture*, pages 243–254. IEEE Press, 2016.

[21] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1135–1143, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks

for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] G. Huang, S. Liu, L. van der Maaten, and K. Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. *arXiv preprint arXiv:1711.09224*, 2017.

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

[27] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *arXiv preprint arXiv:1701.05384*, 2(3):6, 2017.

[28] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[29] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision*, pages 703–718. Springer, 2014.

[31] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[32] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *European Conference on Computer Vision*, pages 541–557. Springer, 2016.

[33] G. Lin, C. Shen, A. v. d. Hengel, and I. Reid. Exploring context with deep structured models for semantic segmentation. *arXiv preprint arXiv:1603.03183*, 2016.

[34] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015.

[35] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[36] O. Miksik, V. Vineet, M. Lidegaard, R. Prasaath, M. Nießner, S. Golodetz, S. L. Hicks, P. Pérez, S. Izadi, and P. H. Torr. The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3317–3326. ACM, 2015.

[37] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *arXiv preprint arXiv:1612.08871*, 2016.

[38] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[39] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[40] G.-J. Qi. Hierarchically gated deep networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] O. Ronneberger, P. Fischer, and T. 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.

[42] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *arXiv preprint arXiv:1801.04381*, 2018.

[44] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. In *Computer Vision–ECCV 2016 Workshops*, pages 852–868. Springer, 2016.

[45] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. *CoRR*, abs/1608.03609, 2016.

[46] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Dag-recurrent neural networks for scene labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3620–3629, 2016.

[47] M. Siam, S. Valipour, M. Jagersand, and N. Ray. Convolutional gated recurrent networks for video segmentation. *arXiv preprint arXiv:1611.05435*, 2016.

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

[51] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[52] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 531–539. IEEE, 2017.

[53] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. *arXiv preprint arXiv:1704.05737*, 2017.

[54] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Deep end2end voxel2voxel prediction. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 402–409. IEEE, 2016.

[55] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *The 2016 International Symposium on Experimental Robotics (ISER 2016)*, 2016.

[56] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Perez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[57] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2016.

[58] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[59] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[60] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[61] H. Zhang, A. Geiger, and R. Urtasun. Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3056–3063, 2013.

[62] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv preprint arXiv:1707.01083*, 2017.

[63] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. *arXiv preprint arXiv:1704.08545*, 2017.

[64] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[65] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[66] W. Zhu and X. Xie. Adversarial deep structural networks for mammographic mass segmentation. *arXiv preprint arXiv:1612.05970*, 2016.