ABCoSeNet: An Atrous Bilateral Collapsed Semantic Segmentation Network

Lalit Lal
University of Toronto
lalit.lal@mail.utoronto.ca

Abstract— **There exists an accuracy and speed tradeoff regarding the performance of a real-time semantic segmentation task. Large networks (many millions of parameters) often achieve remarkable results in accuracy at the cost of very low inference speeds, measured in frames per second (FPS), while real-time networks can infer at remarkably fast speeds (>100 FPS) at the cost of accuracy. Unfortunately, this tradeoff is difficult since they are both crucial components to applications that require safe and time critical decisions. By learning detailed low level information and contextual, high level information separately, many works have been able to combine spatial and categorical information to achieve higher efficiency and accuracy for real-time semantic segmentation tasks. A strong example of this methodology is BiSeNetV2, which additionally uses a booster module to augment the training loss and improve network accuracy at no cost to the inference speed, along with some thoughtful bottleneck structures for its backbone. Additional works intuitively noticed that the detail and semantic branches in bilateral networks do not differ by much in their earlier layers, allowing for a possible reduction in network size by sharing information between these two paths early in the network. A promising example showcasing this work is Fast-SCNN, which is able to achieve remarkably fast inference speeds by using weight sharing as well as an extremely lightweight classifier module at the end of its network. Finally, modern realtime convolutional networks have improved inference speeds through parameter and size reduction by means of depthwise (DW) and depthwise separable (DS) convolutions, as well as increasing receptive fields for semantic tasks without large parameter size increases using atrous (or dilated) convolutions. To this end, a hybrid approach that leverages a bilateral network, a booster module, weight sharing, a lightweight classifier, and dilation is proposed in ABCoSeNet. Despite noticable regression in its accuracy, the network shows competitive inference speeds for realtime semantic segmentation tasks.**

1. Introduction

Semantic segmentation is widely useful for many applications. Specifically, it is a task that revolves around pixel-wise categorical classification. Example applications that would benefit from semantic segmentation include robotics, autonomous vehicles, human to machine interaction, scene understanding, and video surveillance. An interesting dimension to this task also involves the need for real-time processing of input. This requirement is raised from situations like the aforementioned examples, since delayed processing of semantic information can be detrimental in mission critical situations that require this information downstream in the technology pipeline. Due to this increasingly popular requirement, research in this topic has also gained popularity along due to improvements in speed and accuracy. Additionally, the growing popularity of embedded devices and the internet of things (IoT), smart, low-power devices can benefit from real-time performers that can infer at respectable speeds without a GPU. A clear example of this can be augmented reality in construction sites, or for wearable technology in industries such as medical devices and fitness.

Typically, in literature, the advent of deep convolutional neural networks (DCNNs) [7] have allowed for the development of competitive real-time semantic segmentation networks [20] [13] [11] [15] [23] [9]. The success of these networks for real-time applications are founded on an encoder-decoder backbone [16] [18] which are quite memory intensive, or a multi-branch framework such as [14] and [19], along with attempts to use dilation to improve the receptive field of convolution layers [10].

Specifically, the bilateral approach allows for fusing spatial and contextual information along with detailed information in order to preserve both object boundaries as well as determine characteristics of objects themselves. In detail, the semantic branch follows a pattern of wide and shallow layers, attempting to preserve resolution for learning shape information. On the other branch, detailed information is learned using narrow, deeper, layers in order to determine low level features. Upon fusing these two results, semantic segmentation can be achieved.

Moreover, other works [23] [14] [9] propose weight sharing among the first few layers in order to reduce the network size

In this work, the inspiration of a encoder-decoder based bilateral network, using early layer weight sharing, semantic branch dilated convolutions, and parameter reduction using DW/DS convolutions [17] [6] is hybridized in attempts to build from two previous inspirations BiSeNetv2 [19] and Fast-SCNN [14] to improve accuracy compared to Fast-SCNN and improve speed compared to BiSeNetv2. This work is proposed as ABCoSeNet, for an atrous, bilateral, collapsed semantic segmentation network.

Specifically, for accuracy boosting, this network adapts use of modified residual blocks, dubbed Gather and Expand (GA) layers as well as efficient fusion layers, dubbed context embedding (CE) block and bilateral guided aggregation (BGA) to combine semantic and detail information correctly. In order to leverage the training data, a booster training module is also adapted from BiSeNetv2. In order to improve speed performance, the network size is reduced by replacing the final layers with an efficient lightweight classifier as inspired from Fast-SCNN, along with a weight sharing module.

Applied on the Cityscapes dataset [4], ABCoSeNet achieves an accuracy measurement, mean intersection over union (mIoU) of 67.14 % at 110.24 FPS on a Tesla P100 GPU using the full input resolution, without any pretrained weights. In summary, the contributions of this work are:

1) ABCoSeNet - a lightweight architecture that boasts a competitive blend of accuracy (67.14% mIoU) and real-time semantic segmentation (110.24) for high resolution images (1024x2048)
2) We hybridize weight sharing, boosting, dilation, and parameter reduction in a single unified training framework.
3) We obtain competitive results on the Cityscapes dataset [4] without any pretrained weights.

2. Related Work

This section will discuss the frameworks that pave a path for real-time semantic segmentation, with a particular focus on bilateral encoder-decoder architectures and weight sharing.

A. Semantic Segmentation Foundations

The general semantic segmentation foundations consist of three main elements of the backbone. Namely, promising works have proposed what is known as a dilation backbone [22] [2] which build on previously mentioned DCNNs using a combination with dilation convolutions in order to increase the receptive field. While this is helpful for spatial information learning, it unfortunately is not helpful for inference speeds since dilations increase the network parameters and weights as though using a larger convolution kernel. Furthermore, many successful works have shown success using an encoder-decoder framework along with skip connections shown in UNET [16], and even larger kernels (for dilation purposes) such as GCN [12] to fuse features from earlier layers with later layers in attempts to maintain significant features that could otherwise be disregarded due to the vanishing gradient problem. Another showwcase is HRNet [18] which focuses on resolution by fusing multiple branches. While these are powerful and accurate networks in their times, they were unfortunately inadequate when applied in real-time applications, proving a need for further investigation in this subfield.

Real-time specific semantic segmentation networks employ a unique set of techniques, including overall architecture reduction [1], weight sharing in ICNet and Fast-SCNN [23] [14], factorized convolutions with skip connection in ERFNet [15], spatial pyramid dilated convolutions in ESPNet [10], guided upsampling in GUN [9], and early-layer feature reuse in DFANet [8].

Finally, overall architecture size and accuracy have shown to be maintained while also reducing the parameter list through examples such as Xception [3], and even using efficient residual bottleneck units that employ DW/DS convolutions as seen in MobileNets [6] and ShuffleNet [21]. The following two subsections will outline the primary works that have been inspirations to this proposed architecture.

B. Bilateral Networks

In the realm of bilateral networks for real-time semantic segmentation, many works have proven successful, including that of BiSeNetV2 [19], which employs a narrow, deep-channeled branch to capture low level information (named the detail branch), along with a wide, shallow-channeled branch to maintain high resolution spatial information (named as the semantic branch). Expanding on this, in the semantic branch, the authors for this model propose a novel Stem block (semantic branch downsampling), Context Embedding block (for semantic feature fusion), and a Gather-and-Expansion (GE) block which is their modification of a traditional mobile inverted bottleneck as suggested in MobileNetv2 [6]. Furthermore, for their detail branch, the model suggests a novel Bilateral Guided Aggregation block (BGA) to efficiently and accurately fuse the information from the detail and semantic branch. Along with this bilateral approach, this model boasts benefiting from the losses at multiple stages in the semantic branch by using a booster module, causing no loss to inference speed at evaluation time. In summary, the novel suggestions of BiSeNetv2's bilateral approach, along with these elements of a Stem, Context, GE, and BGA block will be elements of inspiration for ABCoSeNet.

It is important to note that the second inspiration of this work, Fast-SCNN, also employs a bilateral encoder-decoder network, which also affirmed the proposed approach. However, the inspiration from Fast-SCNN is actually a different topic, mentioned in the next section: weight sharing.

C. Weight Sharing

Although several approaches have shown the that the bilateral frameworks work well, Fast-SCNN [14] is able to show the validity of weight (or feature) sharing at early stages. The authors propose, in a block called ”Learning to Downsample”, that the equivalent detail and semantic branches can validly use the same feature extractor for the first three layers to ensure low-level sharing. The authors chose three layers to keep this feature sharing shallow, since the intuition is that these branches extract similar low level features early on, but not in deeper layers.

The following section will aim to follow these inspirations and hybridize their innovations, with hopes to increase accuracy against Fast-SCNN, and improve inference speed against BiSeNetv2 [19].

3. Methods

The proposed network, ABCoSeNet is shown below in figure 1. The detail branch from BiSeNetv2 was completely removed and replaced with a subset set of efficient DW convolutions (plus one dilation convolution) in hopes to reduce the network and parameter size. Also, figures labelled with Conv2D are the traditional 2D convolutions, while DSConv and DWConv are depthwise separable and depthwise convolutions respectively. The major blocks (BGA module omitted) of ABCoSeNet are also summarized in table 1, starting with an input of 512x1024x3, where t represents the expansion factor (only applicable to GE layers), c represents output channels of that layer, n represents number of repetitions of that block, s represents the stride of the first block of that sequence (if multiple).

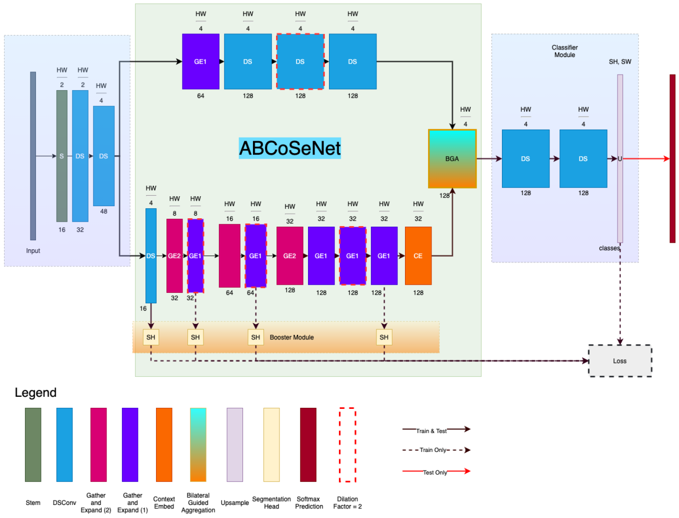


Fig. 1: Proposed Architecture of ABCoSeNet: the blue regions at the ends are inspired from Fast-SCNN, while the green layer in the middle is inspired from BiSeNetv2.

A. Early Stage Collapsing and Parameter Reduction

Inspired from what was addressed as the Stem Block in BiSeNetv2's semantic branch, as well as Fast-SCNN's Learning to Downsample block, ABCoSeNet's modification allows for lightweight and efficient weight sharing in its initial collapsed stage, labelled the ”Shared Stem”, followed by 2 DS convolutions. The intention of the stem block is to maintain as much as possible, globally relevant features using skip connections and max pooling. The output resolution of the first stage (shared stem plus 2 DS convolutions) is (H/4, W/4, 48). From here, the detail and semantic branches each work on separate copies of this output.

Stage	Input	Block	c	n	s
Weight Share	512x1024	Shared Stem	16	1	2
	256x512	DSConv	32	1	1
	256x512	DSConv	48	1	2
Detail Branch	128x256	GE	64	1	1
	128x256	DSConv	128	1	1
	128x256	DSConv	128	1	1
	128x256	DSConv	128	1	1
Semantic Branch	128x256	DSConv	16	1	1
	128x256	GE	32	2	2
	64x128	GE	64	2	2
	32x64	GE	128	4	2
	32x64	CE	128	1	1
Classifier	128x256	DSConv	128	2	1
	128x256	Upsample	N	1	1

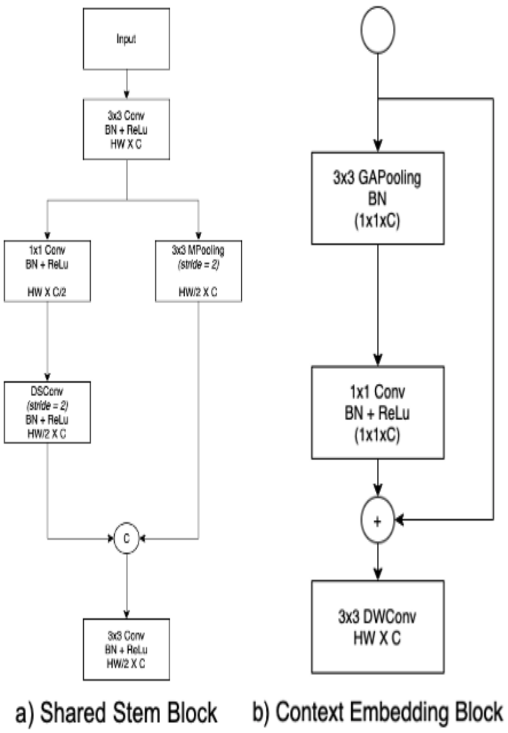
TABLE 1: Tabular summary of major blocks of ABCoSeNet, BGA omitted

As mentioned before, the typical detail branch proposed in many bilateral networks was largely stripped and replaced with a very mild set of DW convolutions. Dilation is applied to the second-last DW convolution in the detail branch to maintain a wide receptive field in its deeper layers, since it is possible that context may be lost at this stage.

Coupled with this change is the mildly modified Context Embedding block, which appears at the end of the semantic branch; the only change is modifying the final fully connected convolution to a DW convolution with intention to reduce the parameter size. The expected output is of this block (H/32, W/32, 128) is unchanged. Both the shared stem and the context block are shown below in figure 2

B. Atrous Convolutions

This section will disclose the remaining specific regions in which atrous convolutions were used, in hopes to improve the receptive field of the proposed network at key layers. Specifically, as mentioned above, the gather-and-expansion (GE) layer is BiSeNetv2's [19] modification of the inverted residual block as seen from MobileNetv2 [6]. An explanation of this GE layer is seen from BiSeNetv2's paper, and enclosed below in figure 3. Intuitively, dilation was applied to the second, fourth, and seventh GE blocks, since this would provide an offset set of locations at which dilation can occur. Dilation is only applied to single stride GE blocks (picture b in figure 3) , and only on the middle 3x3



a) Shared Stem Block b) Context Embedding Block

Fig. 2: a) Shared Stem block, b) Context Embedding Block

DW convolution block. The expected dimensions of the GE outputs are unchanged since the appropriate padding was provided to accommodate these wider receptive fields.

C. Boosting and Classification

Inspired from Fast-SCNN [14], a similar lightweight classifier replaces the Segment Head module from BiSeNetv2 (these modules are now only used for the booster module to augment the loss during training only). The lightweight classifier is simply two DS convolutions followed by an upsample module, which is applied as the *pixelshuffle* module in pyTorch. Finally, this classifier module expects to receive output from the unchanged BGA layer (H/4, W/4, 128), and the expected output of the classifier is (1024, 2048, N), where N is the number of classes in the cityscapes dataset.

To importantly note, the segmentation head (SE) and bilateral guided aggregation modules (BGA) in figure 1 are unchanged adaptations from BiSeNetV2 [19].

During training, softmax is used as the final activation function (after the output of the classification module) to leverage gradient descent, and this is further explained in the next section.

4. Experiments and Results

This section summarizes the training and implementation details along with dataset information. Specifically, methods

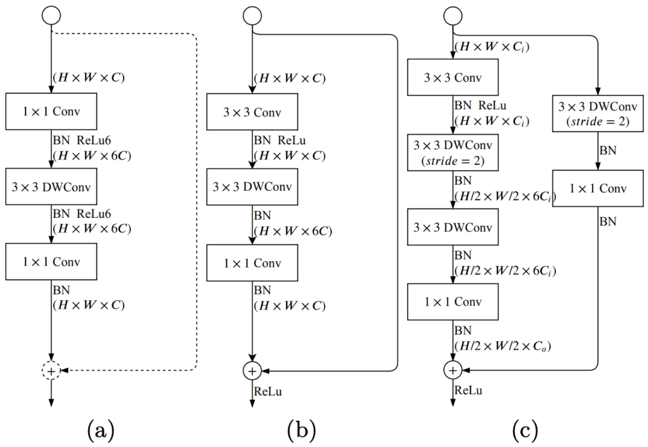


Fig. 3: a) Inverted Residual proposed in MobileNetv2. (b)(c) are the proposed GE layer for a stride = 1, and stride = 2, respectively. Adapted from [19]

on how accuracy and inference speed are measured are detailed along with results against the proposal's inspirations.

A. Dataset

This model is entirely trained from scratch on the Cityscapes [4] dataset, which is split into training, validation, and test sets of 2975, 500, and 1525 images respectively. Only the finely annoted images are used for training and evaluation, which includes 19 classes for the semantic segmentation task. This dataset is an applicable application to validate the proposed model since it incorporates a wide distribution of urban scene landscapes which are the likely scenes of operation for the aforementioned applications (such as autonomous driving or video surveillance). Interesting to note, Fast-SCNN [14] tests the benefit of using the 20k coursely annoted additional images and made note that the impact was insignificant and due to random initlaizations of DCNNs, and training the model for longer yielded similar, if not better results.

B. Implementation Details

The model training pipeline is largely adapted form BiSeNetv2 [19], which includes the following: From scratch, the model is training using a kaiming normal initialization [5]. For training, this model utilises stochastic gradient descent with 0.9 momentum with a batch size of 16. On the convolution layers, a weight decay regularization factor of 0.0005 is employed. The ”poly” learning rate is adapted with $5e^{-2}$ as the initial rate, and in each iteration is multiplied by $(1-iter/(iters_{max}))^{0.9}$ where $iters_{max}$ is 150k. For training loss, since softmax activation was used for classification, a weighted cross-entropy with threshold 0.7 was used, also known as Online Hard Example Mining (OHEM). For data augmentation, the input it randomly flipped in the horizontal axis, then randomly scaled within [0.2, 2], and then randomly cropped to a fixed size for the training pipeline. The fixed

Method	**cls**	**cls**	**cat**	**FPS**
	$mIoU_{val}$	$mIoU_{test}$	$mIoU$	
Fast-SCNN	68.62	68.0	84.7	94.93
BiSeNetv2	73.4	**72.6**	–	**128.558**
ABCoSeNet	70.1	67.14	**86.86**	110.24

TABLE 2: class (test and validation) and category (test) mIoU(%) on cityscapes, and FPS

SS	**SSC**	**MSF**	**MSFC**
70.10	70.27	72.02	71.32

TABLE 3: mIoU(%) on cityscapes validation dataset with different data augmentation: single scale (SS), single scale cropped (SSC), multi-scale flipped (MSF) and multi-scale flopped and cropped (MSFC).

size crop is (1024x2048), which is the same as the original resolution, and then downsampled by a factor of 2 for the training pipeline (512x1024).

For interested readers, the development environment consisted of ubuntu 18.04, NVIDIA Tesla P100 GPU, cuda 10.2, cudNN version 7.x, and pyTorch 1.6.x.

C. Inference

During inference, the booster module is not utilised. In other words, the forward pass of the evaluation stage of the network does not use segmentation heads at each output of the semantic branch. The input of resolution 1024x2048 is first downsampled by 2, run through inference, and then upsampled to the original input size with pixel-wise class labels. The inference time is where this paper varies strongly from the original Fast-SCNN [14] and BiSeNetv2 [19] measurements. The major areas of difference are that this network is not optimized using TensorRT, as the others are, and at the time of testing, the appropriate GPU was not available for testing (NVIDIA GeForce GTX 1080Ti or NVIDIA Titan X). Instead, a Tesla P100 GPU was used, and per frame inference time was calculated by measuring the elapsed time between full resolution input and softmax classification (i.e., the network's output). Furthermore, the inference speed was averaged over all images on the cityscapes test set, which is 1525 images to yield an average FPS measurement for each method. This differs from the 5k measurements that BiSeNetv2 uses for FPS calculation to avoid error fluctuation

Similar to both methods, the accuracy on the dataset is measured using the mean intersection over union metric (mIoU), and is compared between ABCoSeNet's inspirations as well as other models.

D. Results

This section will outline the qualitative and quantitative results of ABCoSeNet alongside its inspirations in the test and validation splits of Cityscapes [4] dataset. Qualitatively, the results of the proposed method are compared alongside its inspirations and the groundtruth in figure 4. The proposed method shows relatively appreciable results.

Furthermore, table 2 includes the final results (mIoU % and FPS) for the networks in question, and table 4 showcase the test set results for a variety of related works mentioned earlier. The generalizability of the network was evaluated using multiple levels of data augmentation (scaling, flipping, cropping) on the validation set in table 3.

It is once again important to note here that FPS calculations, though conducted on different hardware (Tesla P100 instead of NVIDIA Titan X or otherwise) are still measured in the same method as other publications, despite not using TensorRT. From these results, it can be seen that ABCoSeNet has performed strongly in its categorical loss as well as its inference speeds. However, from tables 4 and 2 it also shows that it was also not able to generalize well (performing competitively in the validation data, whilst seeing significant regression in the test data). A plausible reasoning behind this is the significant changes made in the detail branch of ABCoSeNet, causing loss in detail information that would help during the fusing of semantic and detail information later on.

5. Concluding Remarks and Future Works

In this work, the hybrid of a bilateral yet early weight sharing network was proposed with a lightweight classifier in attempts to efficiently fuse semantic (high level) and detail (low level) features for real-time semantic segmentation tasks. The goal was to improve accuracy compared to Fast-SCNN [14], while improving the inference speed against BiSeNetv2 [19]. Unfortunately, while the validation results looked very promising to show an accuracy improvement (first goal), it showed that ABCoSeNet did not generalize well and performed worse than Fast-SCNN in the test results. Additionally, ironically, the local FPS calculations showed that ABCoSeNet outperformed Fast-SCNN in FPS, but not BiSeNetv2, which was unexpected, but likely due to different hardware and optimizations not applying. Moving forward, some improvements involve utilizing and evaluating on more competitive hardware (NVIDIA Titan X, for

Method	**Backbone**	$mIoU_{test}$
Fast-SCNN	none	68.0
BiSeNetv2	none	72.6
ESPNet	ESPNet	60.3
ERFNet	none	68.0
ICNet	PSPNet50	69.5
GUN	DRD-D-22	70.4
PSPNet	ResNet101	**78.4**
ABCoSeNet	none	67.14

TABLE 4: mIoU(%) on cityscapes test set

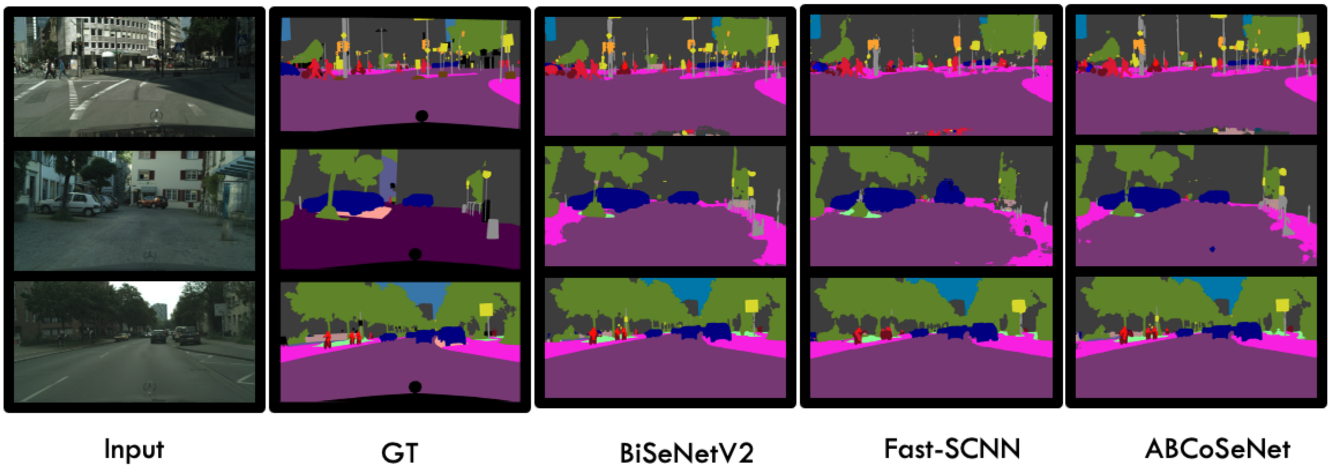


Fig. 4: Qualitative results comparing three images from cityscapes validation against ground truth, proposed method, and inspirations.

example) as well as TensorRT, and conducting an ablation study on each hybridized method (weight sharing, dilation, lightweight classifier), in order to understand how to improve the generalizability of ABCoSeNet.

REFERENCES

[1] Badrinarayanan, V., Kendall, A., Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017, 39(12):2481–2495

[2] Chen, LC., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, AL., Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv, 2016

[3] Chollet, F. Xception: Deep learning with depthwise separable convolutions. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

[4] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016

[5] He, K., Zhang, X., Ren, S., Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In IEEE International Conference on Computer Vision (ICCV), 2015, pp 1026–1034

[6] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs], 2017

[7] Krizhevsky, A., Sutskever, I., Hinton, GE., Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012

[8] Li, H., Xiong, P., Fan, H., Sun, J. Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019b

[9] Mazzini, D., Guided Upsampling Network for Real-Time Semantic Segmentation. In *BMVC*, 2018

[10] Mehta, S., Rastegari, M., Shapiro, LG., Hajishirzi, H. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019

[11] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. arXiv:1606.02147 [cs], 2016

[12] Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J., Large kernel matters–improve semantic segmentation by global convolutional network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[13] Poudel, R., Bonde, U., Liwicki, S., Zach, S. Contextnet: Exploring context and detail for semantic segmentation in real-time. In *BMVC*, 2018

[14] Poudel, R., Liwicki, S., Cipolla, R., Fast-scnn: Fast semantic segmentation network. arXiv preprint arXiv:1902.04502, 2019

[15] Romera, E., Alvarez, JM., Bergasa, LM., Arroyo, R., ERFNet: Efficient Residual Factorized ConvNet for RealTime Semantic Segmentation. in *IEEE* Transactions on Intelligent Transportation Systems, 2018

[16] Ronneberger, O., Fischer, P., Brox, T., U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015

[17] Sifre, L., Rigid-motion scattering for image classification. PhD thesis, 2014

[18] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B., Deep high-resolution representation learning for visual recognition. In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019

[19] Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. arXiv preprint arXiv:2004.02147, 2020

[20] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N., Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018

[21] Zhang, X., Zhou, X., Lin, M., Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp 6848–6856

[22] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., Pyramid scene parsing network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[23] Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J., ICNet for RealTime Semantic Segmentation on High-Resolution Images. In *ECCV*, 2018