AASeg: Attention Aware Network for Real Time Semantic Segmentation

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we present a new network named Attention Aware Network (AASeg) for real time semantic image segmentation. Our network incorporates spatial and channel information using Spatial Attention (SA) and Channel Attention (CA) modules respectively. It also uses dense local multi-scale context information using Multi Scale Context (MSC) module. The feature maps are concatenated individually to produce the final segmentation map. We demonstrate the effectiveness of our method using a comprehensive analysis, quantitative experimental results and ablation study using Cityscapes, ADE20K and Camvid datasets. Our network performs better than most previous architectures with a 74.4% Mean IOU on Cityscapes test dataset while running at 202.7 FPS.

1 Introduction

Semantic Segmentation is a fundamental problem in computer vision where the goal is to label each and every pixel in the image to its appropriate class. Since it is required to be deployed in real world settings like robots and autonomous vehicles, hence there is a need to balance the speed vs performance tradeoff. Fully Convolutional Network (FCN) (Long et al., 2015) composed of convolutional layers was one of the first works to get strong semantic representation.

However, this method was not able to capture boundary information accurately. Atrous convolutions (Yu and Koltun, 2015) at the last several stages of their network was used to give feature maps with strong semantic representation, thus solving the problem with FCN based architectures. However, this comes at at the cost of increased computational complexity. For real world navigation tasks like autonomous driving, there is a need to improve the Frame Per Second(FPS).

Chen et al. (2017) and (Yu and Koltun, 2015) used dilated convolutions to increase the recpetive field while maintaining the number of parameters. SegNet (Badrinarayanan et al., 2017) utilizes a small network structure and the skip-connected method to achieve improved FPS. (Mehta et al., 2018), (Paszke et al., 2016) and (Poudel et al., 2019) proposed unique approaches to tackle real time semantic segmentation problem.

2 Related Work

(Fu et al., 2019) introduces spatial-wise and channelwise attention modules to enhance the recpetive field (Paszke et al., 2016) trims a a lot of convolution filters to reduce computation. ICNet (Zhao et al., 2018a) proposed an image cascade network using multiresolution branches. (Iandola et al., 2016) allows the neural network to find the critical channels of the feature map and select the most suitable channels by itself.

ESPNet (Mehta et al., 2018) introduces an efficient spatial pyramid (ESP), which brings great improvement in both speed and performance. Bilateral Segmentation Network (BiSeNet) (Yu et al., 2018a) used two parts: Spatial Path (SP) is used to get with the loss of spatial information and Context Path (CP) for compressing the receptive field. (Yu et al., 2020) used multi-path framework to combine the low-level details and high-level semantics.

(Li et al., 2019b) utilizes a light-weight backbone to speed up its network and a multi scale feature aggregation to improve accuracy. SwiftNet (Orsic et al., 2019) used lateral connections to restore the prediction resolution while maintaining the speed. (Lin et al., 2017) uses a multipath refinement network to refine the feature but ignores the global context feature.

The speed-Accuracy performance comparison of state of the art methods on the Cityscapes test set is shown in Figure 1:

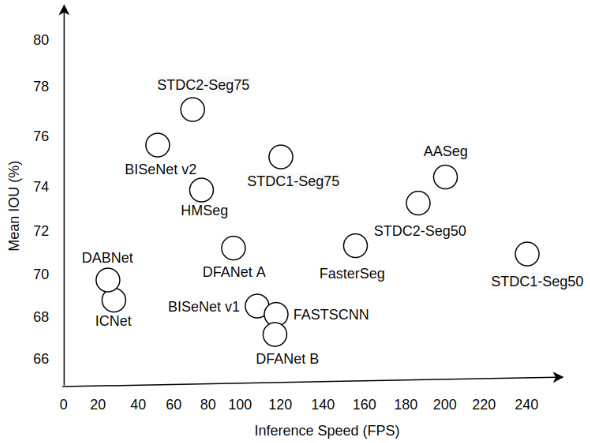

Figure 1: Speed-Accuracy performance comparison on the Cityscapes test set. Our approach achieves higher mean IOU while still being faster than most existing methods.

.

We summarize our main contributions as follows:

• We propose a novel Attention Aware Network (AASeg) for real time semantic segmentation.

• Our network is comprised of three parts: Spatial Attention(SA) module to capture the spatial dependencies of the feature maps, Channel Attention(CA) module to extract high level semantic information and Multi Scale Context(MSC) module to learn the information flow between feature maps of consecutive levels.

• Detailed experiments and analysis indicate the efficacy of our proposed network in not only improving the performance but also FPS. We achieve results on par with previous state of the art using Cityscapes, Camvid and on ADE20K datasets.

3 Background

3.1 Spatial Information

The spatial information of the image is important to predict the detailed output for semantic segmentation. Modern existing approaches uses encoding of spatial information. DUC (Wang et al., 2018), PSPNet (Zhao et al., 2017), DeepLab v2 (Chen et al., 2017) use the dilated convolution to preserve the spatial size of the feature map.

3.2 Context Information

Semantic segmentation requires context information to generate a high-quality result. Most of the used methods enlarge the receptive field or fuse different contextual information. (Chen et al., 2017), (Wang et al., 2018) and (Yu and Koltun, 2015) use different dilation rates in convolution layers to capture different scale contextual information. In (Chen et al., 2017), an “ASPP” module is used to capture context information of different receptive field. PSPNet (Zhao et al., 2017) applies a “PSP” module which contains several different scales of average pooling layers.

3.3 Attention Mechanism

(Hu et al., 2018) applied channel attention for image recognition and achieve the state-of-the-art. (Yu et al., 2018b) proposed a network that learns the global context as attention and revise the features. Multi scale network along with a custom attention module was proposed by (Sagar and Soundrapandiyan, 2020) for semantic segmentation.

3.4 Feature Fusion

The spatial information captured by the Spatial Path consists of rich detailed information. The output feature of the Context Path is made up of contextual information. Feature fusion was used to achieve state of the art results on image classification, object detection and instance segmentation using DMSANet (Sagar, 2021b).

4 Proposed Method

4.1 Dataset

The following datasets have been used to benchmark our results:

1. Cityscapes It is used for urban street segmentation. The 5000 annotated images are used in our experiments which are divided into 2975, 500 and 1525 images for training, validation, and testing respectively.

2. ADE20K This dataset contains labels of 150 object categories. The dataset includes 20k,2k and 3k images for training, validation and testing respectively.

3. CamVid This dataset is used for semantic segmentation for autonomous driving scenarios. It is composed of 701 densely annotated images.

4.2 Network Architecture

The fundamental goal in semantic segmentation is to map an RGB image $X \in R^{H \times W \times 3}$ to a semantic map $Y \in R^{H \times W \times C}$ with the same spatial resolution $H \times W$, where C is the number of classes of objects present in image. The input image X is converted to a set of feature maps F_l where l=1,...,3 from each network stage, where $F_l \in R^{H_l W_l C_l}$ is a C_l-dimensional feature map.

Our network dosen't use any backbone to extract features from the input image unlike many previous architectures. The input image is first passed through a block comprising of convolutional, batch normalization and ReLU activation function.

We express a convolution layer $W^n(x)$ as follows:

$$\mathrm{W}^n(x) = \mathbf{W}^{n \times n} \odot x + \mathbf{b} \tag{1}$$

where $\odot$ represents the convolution operator, $W^{n \times n}$ represents the $n \times n$ convolutional kernel, x represents the input data and b represents the bias vector.

4.3 Multi Scale Context Module

The feature map after passing through convolutional block is split to three different parts with 1×1 convolution, 3×3 convolution and 5×5 convolution respectively. The individual feature maps are then fused together. The output fused feature map is first convolved with a 1×1 convolution to reduce the number of channels from 2048 to 256. The feature map produced is of size $H \times W \times n_c$, where H, W are the height and width of the feature map, and n_c denotes the number of channels.

The input feature map is convolved with dilated convolution layers with increasing dilation rates of 3, 6 and 12. The dilated convolution layer input at every stage is formed by concatenating the input feature map with the outputs from previous convolutions. At the final step, the outputs from the three dilated convolutions are concatenated with the input feature map.

4.4 Spatial Attention Module

The spatial attention module is used for capturing the spatial dependencies of the feature maps. The spatial attention (SA) module in our network is defined below:

$$f_{SA}(x) = f_{sigmoid}\left(W_2\left(f_{ReLU}\left(W_1(x)\right)\right)\right) \tag{2}$$

where W_1 and W_2 denotes the first and second 1×1 convolution layer respectively, x denotes the input data, $f_{Sigmoid}$ denotes the sigmoid function, f_{ReLU} denotes the ReLu activation function.

The spatial attention module used in this work is shown in Figure 2:

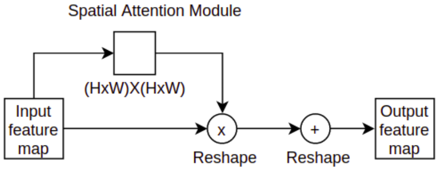

Figure 2: Details of our spatial attention module

.

4.5 Channel Attention Module

The channel attention module is used for extracting high level multi-scale semantic information. The channel attention (CA) module in our network is defined below:

$$f_{CA}(x) = f_{sigmoid}(W_2(f_{ReLU}(W_1 f^1_{AvgPool}(x)))) \tag{3}$$

where W_1 and W_2 denotes the first and second 1×1 convolution layer, x denotes the input data. $f^1_{AvgPool}$ denotes the global average pooling function, $f_{Sigmoid}$ denotes the Sigmoid function, f_{ReLU} denotes ReLU activation function.

The channel attention module used in this work is shown in Figure 3:

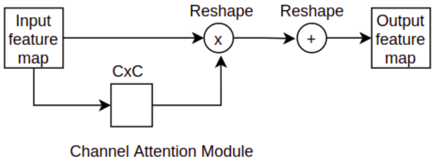

Figure 3: Details of our channel attention module

.

4.6 Aggregation

We denote the concatenation operation as follows:

$$x_{concat} = x_1 \oplus x_2 \oplus x_3 \tag{4}$$

where $\oplus$ represents the concatenation operator and x_1, x_2 and x_3 represents the features of the two branches. The AASeg module can be denoted as follow:

$$x_{AASeg} = \left(\left(f_{SA}\left(x_{concat}\right) \otimes x_{concat}\right) \oplus \left(f_{CA}\left(x_{concat}\right) \otimes x_{concat}\right) \oplus \left(f_{MSC}\left(x_{concat}\right) \otimes x_{concat}\right)\right) \tag{5}$$

where $\oplus$ represents the concatenation operator, f_{CA} represents the channel attention module mentioned in Equation 2, f_{SA} represents the spatial attention module, f_{MSC} represents the multi scale attention and x_{concat} represents the combined feature.

we use $ConvX_i$ to denote the operations of i^{th} block. Therefore, the output of i^{th} block is calculated as follows:

$$x_i = ConvX_i\left(x_{i-1}, k_i\right) \tag{6}$$

where $ConvX$ includes one of each convolutional layer, batch normalization layer and ReLU activation layer, k_i is the kernel size of convolutional layer, x_{i-1} and x_i are the input and output of i^{th} block. Fusion operation is used to combine high-level features with low-level features using Equation 7:

$$x_{output} = F\left(x_1, x_2, \ldots, x_n\right) \tag{7}$$

The overall structure of our proposed AASeg network is shown in Figure 4.

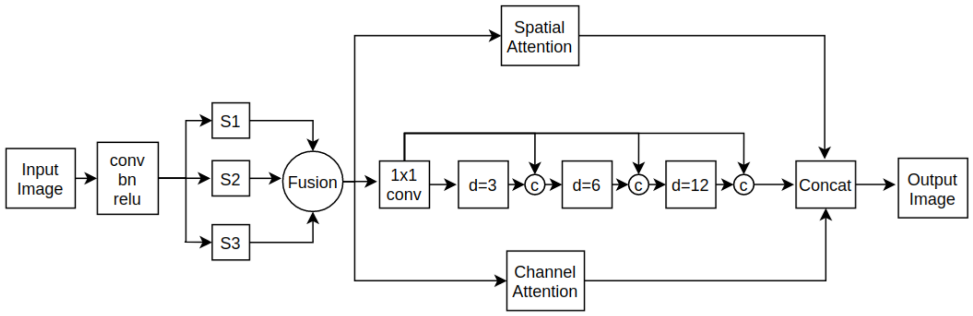

Figure 4: Architecture of proposed AASeg network architecture. “c” denotes concatenation.

.

4.7 Loss Functions

We use cross-entropy loss function to weigh the difference between the forward propagation result of the network and the ground truth of the samples. The cross-entropy loss is calculated as defined in Equation 8:

$$L_{ce} = \frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + \left(1 - y_n\right)\log\left(1 - \hat{y}_n\right)\right] \tag{8}$$

where N denotes the total number of samples, y_n denotes the probability that the forward propagation result is true, and 1-y_n denotes the probability that the forward propagation result is false.

We also use the auxiliary supervision loss L_{aux} to improve the model performance thus making it easier to optimize. The auxiliary loss can be defined as:

$$\mathcal{L}_{aux} = -\frac{1}{BN}\sum_{i=1}^{B}\sum_{j=1}^{N}\sum_{k=1}^{K}\mathbb{I}\left(g_j^i = k\right)\log\left(\frac{\exp\left(p_{j,k}^i\right)}{\sum_{m=1}^{K}\exp\left(p_{j,m}^i\right)}\right) \tag{9}$$

$$I\left(g_j^i = k\right) = \left\{\begin{array}{ll} 1, & g_j^i = k \\ 0, & otherwise \end{array}\right. \tag{10}$$

where B is the mini batch size, N is the number of pixels in every batch; K is the number of categories; p_{ijk} is the prediction of the j^{th} pixel in the i^{th} sample for the k^{th} class, $I(g_{ij} = k)$ is a function which is defined in Equation 10.

The class attention loss L_{cls} from channel attention module is also used. The class attention loss is defined as follows:

$$\mathcal{L}_{cls} = -\frac{1}{BN} \sum_{i=1}^{B} \sum_{j=1}^{N} \sum_{k=1}^{K} \mathbb{I}\left(g_j^i = k\right) \log\left(\frac{\exp\left(a_{j,k}^i\right)}{\sum_{m=1}^{K} \exp\left(a_{j,m}^i\right)}\right) \tag{11}$$

where a_{ijk} is the value generated of the class attention map of the j^{th} pixel in the i^{th} sample for the k^{th} class. We combine the three terms to balance final loss term as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{ce} + \lambda_2 \mathcal{L}_{cls} + \lambda_3 \mathcal{L}_{aux} \tag{12}$$

where $\lambda 1$, $\lambda 2$ and $\lambda 3$ are set as 1, 0.5 and 0.5 to balance the loss.

4.8 Implementation Details

PyTorch deep learning framework is used to carry out our experiments. Our neural networks is trained with stochastic gradient descent (SGD) as optimizer, batch size of 16, momentum of 0.9 and weight decay of $5e^{-4}$. The network is trained for 20K iterations with an initial learning rate value of 0.01. The, “poly” learning rate policy is used to decay the initial learning rate while training the model to reduce over-fitting.

Data augmentation operations like random horizontal flip, random resizing with scale range of [1.0, 2.0], and random cropping with crop size of 1024×1024 was done.

4.9 Evaluation Metrics

For quantitative evaluation of the performance of our network, mean of class-wise intersection-over-union (mIoU) is used for bench-marking the performance, number of float-point operations (FLOPs) and frames per second (FPS) are used for bench-marking the speed.

5 Results

We present the segmentation accuracy and inference speed of our proposed method on Cityscapes validation and test set in Table 1. We use the training set and validation set to train our models before submitting to Cityscapes online server.

The qualitative results on Cityscapes validation set are presented in Figure 5:

Table 1: Comparisons with other state-of-the-art methods using Cityscapes dataset. no indicates the method do not have any backbone for training and testing. Best results are highlighted in bold.

Model	Resolution	Backbone	mIoU val(%)	mIoU test(%)	FPS
DFANet B	1024 × 1024	Xception B	-	67.1	120
DFANet A	1024 × 1024	Xception A	-	71.3	100
BiSeNetV1	768 × 1536	Xception39	69.0	68.4	105.8
BiSeNetV1	768 × 1536	ResNet18	74.8	74.7	65.5
SFNet	1024 × 2048	DF1	-	74.5	121
BiSeNetV2	512 × 1024	no	73.4	72.6	156
BiSeNetV2-L	512 × 1024	no	75.8	75.3	47.3
FasterSeg	1024 × 2048	no	73.1	71.5	163.9
STDC1-Seg50	512 × 1024	STDC1	72.2	71.9	**250.4**
STDC2-Seg50	512 × 1024	STDC2	74.2	73.4	188.6
STDC1-Seg75	768 × 1536	STDC1	74.5	75.3	126.7
STDC2-Seg75	768 × 1536	STDC2	**77.0**	**76.8**	97.0
AASeg	512 × 1024	no	74.8	74.4	202.7

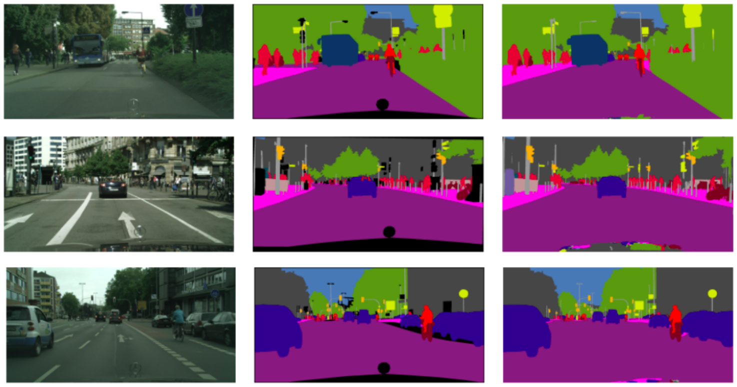

Figure 5: Visualized segmentation results on Cityscapes validation set. The three columns left-to-right refer to input image, ground truth, prediction from our network.

.

We evaluate our method on CamVid dataset. The comparison results with other methods is shown in Table 2:

Table 2: Comparisons with other state-of-the-art methods on CamVid. no indicates the method do not have a backbone for both training and testing. Best results are highlighted in bold.

Model	Resolution	Backbone	mIoU(%)	FPS
DFANet A	720 × 960	no	64.7	120
DFANet B	720 × 960	no	59.3	160
BiSeNetV1	720 × 960	Xception39	65.6	175
BiSeNetV1	720 × 960	ResNet18	68.7	116.3
BiSeNetV2	720 × 960	no	72.4	124.5
BiSeNetV2-L	720 × 960	no	73.2	32.7
SFNet	720 × 960	DF2	70.4	134.1
SFNet	720 × 960	ResNet-18	73.8	35.5
SFNet	720 × 960	DF2	70.4	134.1
SFNet	720 × 960	ResNet-18	73.8	35.5
STDC1-Seg	720 × 960	STDC1	73.0	**197.6**
STDC2-Seg	720 × 960	STDC2	**73.9**	152.2
AASeg	720 × 960	no	73.5	188.7

The comparison of our network with previous state of the art methods for semantic segmentation using ADE20K validation set is shown in Table 3:

Table 3: Performance comparison with results reported on ADE20K validation set. Best results are highlighted in bold.

Method	Mean IoU(%)	GFLOPs
PSPNet50	42.78	167.6
PSPNet101	43.29	238.4
SFNet	42.81	**75.7**
SFNet	44.67	94.0
DCANet	45.49	-
AASeg	**46.29**	80.26

6 Ablation Study

We explore the effect of 3 different upsamling operations: bilinear upsampling, deconvolution and nearest neighbor upsampling as shown in Table 4. The best results are obtained using bilinear upsampling but there isn't considerable difference.

Table 4: Ablation study on Upsampling operation in our network using Cityscapes validation set. Best results are highlighted in bold.

Method	mIoU (%)
bilinear upsampling	**79.2**
deconvolution	78.5
nearest neighbor	78.3

We also try the various kernel size in Table 5. Larger kernel size of 7 × 7 is also tried which results in a similar performance but increases computational complexity.

We use increasing sequence of dilation rates in our network, which is 0, 1, 2, 3. It seems large dilation rates increases the performance but comes at the cost of increased number of parameters, hence decrease in FPS. This effect is shown in Table 6:

Table 5: Ablation study on kernel size k in our network using Cityscapes validation set. Best results are highlighted in bold.

Method	mIoU (%)	GFlops
k = 1	79.2	**118.2**
k = 3	79.4	120.8
k = 5	79.5	127.5
k = 7	**79.5**	136.1

Table 6: Ablation study results on Cityscapes validation set. FPS are estimated for an input image of resolution of 512×1024. Best results are highlighted in bold.

Model	mIoU (%)	FPS	Parameters (M)
AASeg-baseline	79.2	118.2	**0.76**
AASeg-w/o dilation	77.4	**121.3**	0.80
AASeg-(r=1)	79.8	115.5	0.79
AASeg-(r=2)	80.1	102.7	0.84
AASeg-(r=3)	**80.2**	88.6	0.90

7 Conclusions

In this paper, we present a new network named Attention Aware Network (AASeg) for real time semantic segmentation. We use Spatial Attention(SA) and Channel Attention(CA) modules to enhance features of objects without adding extra computational cost. The fused spatial-channel attention along with Multi Scale Context(MSC) module enables network to extract discriminative and robust features of targets or background. We test our network using Cityscapes, ADE20K and Camvid datasets. Our network balances speed-performance tradeoff at par with previous state of the art network architectures. In the future, we would like to use multi scale attention module for instance segmentation problem.

Acknowledgments

We would like to thank Nvidia for providing the GPUs.

References

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

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

W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang. Fasterseg: Searching for faster real-time semantic segmentation. *arXiv preprint arXiv:1912.10917*, 2019.

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.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9725, 2021.

J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020.

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, 2016a.

K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016b.

J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

P. Hu, F. Caba, O. Wang, Z. Lin, S. Sclaroff, and F. Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8827, 2020.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

L. Jing, Y. Chen, and Y. Tian. Coarse-to-fine semantic segmentation from image-level labels. *IEEE Transactions on Image Processing*, 29:225–236, 2019.

G. Li, I. Yun, J. Kim, and J. Kim. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv preprint arXiv:1907.11357*, 2019a.

H. Li, P. Xiong, H. Fan, and J. Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9522–9531, 2019b.

P. Li, X. Dong, X. Yu, and Y. Yang. When humans meet machines: Towards efficient segmentation networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020a.

X. Li, A. You, Z. Zhu, H. Zhao, M. Yang, K. Yang, S. Tan, and Y. Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision*, pages 775–793. Springer, 2020b.

C.-Y. Lin, Y.-C. Chiu, H.-F. Ng, T. K. Shih, and K.-H. Lin. Global-and-local context network for semantic segmentation of street view images. *Sensors*, 20(10):2907, 2020.

G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

Y. Liu, C. Xu, and X. Jin. Dcanet: Dense context-aware network for semantic segmentation. *arXiv preprint arXiv:2104.02533*, 2021.

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.

S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 552–568, 2018.

R. Niu, X. Sun, Y. Tian, W. Diao, K. Chen, and K. Fu. Hybrid multiple attention network for semantic segmentation in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

M. Orsic, I. Kreso, P. Bevandic, and S. Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12607–12616, 2019.

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.

R. P. Poudel, S. Liwicki, and R. Cipolla. Fast-scnn: fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019.

A. Sagar. Aa3dnet: Attention augmented real time 3d object detection. *arXiv preprint arXiv:2107.12137*, 2021a.

A. Sagar. Dmsanet: Dual multi scale attention network. *arXiv preprint arXiv:2106.08382*, 2021b.

A. Sagar and R. Soundrapandiyan. Semantic segmentation with multi scale spatial attention for self driving cars. *arXiv preprint arXiv:2007.12685*, 2020.

M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

Q. Tang, F. Liu, T. Zhang, J. Jiang, and Y. Zhang. Attention-guided chained context aggregation for semantic segmentation. *arXiv preprint arXiv:2002.12041*, 2020.

P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1451–1460. IEEE, 2018.

X. Yang, S. Li, Z. Chen, J. Chanussot, X. Jia, B. Zhang, B. Li, and P. Chen. An attention-fused network for semantic segmentation of very-high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 177:238–262, 2021.

C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018a.

C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018b.

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

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

H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

Y. Zhang, Z. Qiu, J. Liu, T. Yao, D. Liu, and T. Mei. Customizable architecture search for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11641–11650, 2019.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018a.

H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018b.