AttaNet: Attention-Augmented Network for Fast and Accurate Scene Parsing

Qi Song,[1,3] **Kangfu Mei,**[1,2] **Rui Huang,**[1,2]*
[1]Shenzhen Institute of Artificial Intelligence and Robotics for Society
[2]The Chinese University of Hong Kong, Shenzhen
[3]Jilin University
{songqi, ruihuang}@cuhk.edu.cn, kangfumei@link.cuhk.edu.cn

Abstract

Two factors have proven to be very important to the performance of semantic segmentation models: global context and multi-level semantics. However, generating features that capture both factors always leads to high computational complexity, which is problematic in real-time scenarios. In this paper, we propose a new model, called Attention-Augmented Network (AttaNet), to capture both global context and multi-level semantics while keeping the efficiency high. AttaNet consists of two primary modules: Strip Attention Module (SAM) and Attention Fusion Module (AFM). Viewing that in challenging images with low segmentation accuracy, there are a significantly larger amount of vertical strip areas than horizontal ones, SAM utilizes a striping operation to reduce the complexity of encoding global context in the vertical direction drastically while keeping most of contextual information, compared to the non-local approaches. Moreover, AFM follows a cross-level aggregation strategy to limit the computation, and adopts an attention strategy to weight the importance of different levels of features at each pixel when fusing them, obtaining an efficient multi-level representation. We have conducted extensive experiments on two semantic segmentation benchmarks, and our network achieves different levels of speed/accuracy trade-offs on Cityscapes, e.g., 71 FPS/79.9% mIoU, 130 FPS/78.5% mIoU, and 180 FPS/70.1% mIoU, and leading performance on ADE20K as well.

Introduction

Scene parsing, also known as semantic segmentation, predicts dense labels for all pixels in an image. As one of the fundamental tasks in computer vision, it has various applications in the fields of autonomous driving, video surveillance, robot sensing, and so on, many of which have a high demand for both segmentation accuracy and inference speed.

To achieve high accuracy, segmentation models need to generate features with global context information and multi-level semantics, both of which are known to be important factors in scene parsing. Global scene clues are typically captured via heavy networks with sizable receptive fields, e.g., PSPNet (Zhao et al. 2017), DANet (Fu et al. 2019a), and AlignSeg (Huang et al. 2020) use ResNet101 (He et al. 2016) as the backbone network. Besides, multi-level representations rely on both semantic information in high-level features and spatial details in low-level features. Nevertheless, both of them almost always require huge computation which is problematic to real-time applications.

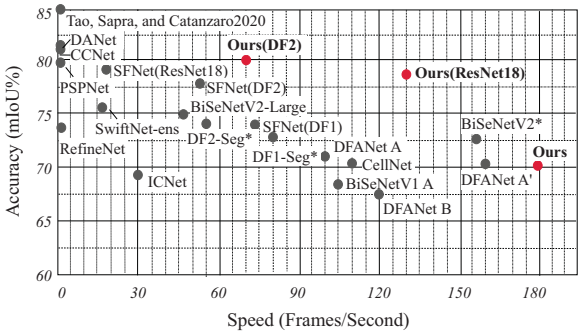

Figure 1: Inference speed and mIoU performance on the Cityscapes test set. Our method is marked as red points, while grey dots represent other methods. $*$ indicates that the model uses TensorRT for acceleration.

On the other hand, in order to accelerate models for real-time scenarios, many state-of-the-art methods adopt light-weight backbone networks or restrict the size of input images. Though greatly boosting the inference speed, their accuracy is still unsatisfying due to the compromise on the aforementioned two factors. To achieve a better speed/accuracy trade-off, we propose Attention-Augmented Network (AttaNet) which can capture both global context and multi-level representations while keeping high computational efficiency.

In order to capture non-local contextual information with limited computation, we started by investigating one of the most commonly used approaches in scene parsing, the self-attention mechanism (Cheng, Dong, and Lapata 2016; Vaswani et al. 2017), which is capable of capturing long-range dependencies. Yet we can observe that these self-attention based models need to generate huge attention maps which are computationally expensive, e.g., the non-local block in Non-local Networks (Wang et al. 2018) and the position attention module in DANet (Fu et al. 2019a) both have a computational complexity of $O((H \times W) \times (H \times W))$,

*Rui Huang is the corresponding author.

where H and W donate the spatial dimensions of the input feature map. Therefore, many recent studies were proposed to achieve the same goal in a more efficient way, e.g., CCNet (Huang et al. 2019) reduces the computational complexity to $O((H \times W) \times (H + W - 1))$. However, the computational overhead is still too high to meet the real-time requirement.

In this work, we address this challenge by proposing an efficient self-attention based module called Strip Attention Module (SAM). SAM is inspired by the segmentation results in previous works, from which we find that various networks all achieved the lowest accuracies in classes such as *fence*, *pole*, and *train*, which are contextually consistent and robust in a specific direction. The usage of large square pooling kernels would corrupt the structural details of these classes and incorporate contaminating information from irrelevant regions. This motivates us to introduce a striping operation into the traditional self-attention method, which can reduce the size of the attention map and also strengthen the directional consistency. Specifically, viewing that for classes with low accuracy there are a significantly larger amount of vertical strip areas than horizontal ones, SAM utilizes a striping operation to encode the global context in the vertical direction and then harvests long-range relations along the horizontal axis. By applying SAM, each position in the feature map is connected with pixels in different column spaces, and the computational complexity is reduced to $O((H \times W) \times W)$. Besides, SAM can be trivially modified to perform horizontal striping for different purposes.

Moreover, we investigate how to generate multi-level representations for each pixel with negligible computational overhead. In mainstream semantic segmentation architectures, the feature fusion method is used to incorporate multi-level semantics into encoded features. Here we choose the cross-level aggregation architecture for its high efficiency. However, we find that multi-level features have different properties, e.g., high-level features encode stronger semantics while low-level features capture more spatial details. Simply combining those features would limit the effectiveness of information propagation. To mitigate this issue, we propose an Attention Fusion Module (AFM) which adopts an attention strategy that learns to weight multi-level features at each pixel location with minimal computation. Besides, we only apply AFM between the last two stages of the backbone network to further improve the efficiency.

We conducted extensive experiments on the two most competitive semantic segmentation datasets, i.e., Cityscapes (Cordts et al. 2016) and ADE20K (Zhou et al. 2017). Our approach achieves top performance on both of them. To illustrate the performance comparisons, we show the accuracy and inference time of different networks on the Cityscapes dataset in Figure 1.

To summarize, our main contributions are three-fold:

- We introduce a Strip Attention Module which is able to capture long-range dependencies with only slightly increased computational cost.
- We propose a novel Attention Fusion Module to weight the importance of multi-level features during fusion, which attains a multi-level representation effectively and efficiently.
- Not only did our network achieve the leading performance on Cityscapes and ADE20K, the individual modules can also be combined with different backbone networks to achieve different levels of speed/accuracy tradeoffs. Specifically, our approach obtains 79.9%, 78.5%, and 70.1% mIoU scores on the Cityscapes test set while keeping a real-time speed of 71 FPS, 130 FPS, and 180 FPS respectively on GTX 1080Ti.

Related Work

Self-attention Model. Self-attention models can capture long-range dependencies and have been widely used in many tasks. Mou et al. (Mou, Hua, and Zhu 2019) introduced two network units to model spatial and channel relationships respectively. OCNet (Yuan and Wang 2018) and DANet (Fu et al. 2019a) use the self-attention mechanism to capture long-range dependencies from all pixels. However, these methods need to generate huge attention maps, which adds much computational overhead. To reduce the complexity of the self-attention mechanism, CCNet (Huang et al. 2019) leverages two criss-cross attention modules to generate sparse connections $(H + W + 1)$ for each position. ACFNet (Zhang et al. 2019) directly exploits class-level context to reduce the computation along channel dimensions. To capture long-range relations more effectively and efficiently, we introduce a striping operation in the Strip Attention Module. Different from the strip pooling used to enlarge the receptive field in work (Hou et al. 2020), ours is designed to strengthen the contextual consistency in a specific direction while reduce the size of the affinity map.

Multi-level Feature Fusion. Feature fusion is frequently employed in semantic segmentation to combine multi-level representations (Long, Shelhamer, and Darrell 2015; Ronneberger, Fischer, and Brox 2015; Lin et al. 2017; Badrinarayanan, Kendall, and Cipolla 2017; Fu et al. 2019b). For multi-level feature fusion, one solution is to adopt the multi-branch framework, e.g., ICNet (Zhao et al. 2018) and BiSeNet series (Yu et al. 2018a, 2020) add an extra branch to remedy the lost spatial details in high-level features. To further boosts the inference speed, another type of methods (Li et al. 2019a, 2020) implement a cross-level feature aggregation architecture with less computation. Nevertheless, all these methods ignore the representation gap among multi-level features, which limits the effectiveness of information propagation. Recently, GFF (Li et al. 2019b) uses gates to control information propagation, but ignores to limit the computation while maintaining effectiveness. In this regard, we propose the Attention Fusion Module which adopts a lightweight attention strategy to bridge the gap among multi-level features with high adaptability and efficiency.

Real-time Segmentation. The goal of real-time semantic segmentation algorithms is to generate high-quality predictions while keeping high inference speed. ICNet proposes an image cascade network using multi-resolution images as input to raise efficiency. BiSeNetV2 introduces a detail branch and a semantic branch to reduce calculation. Both of them

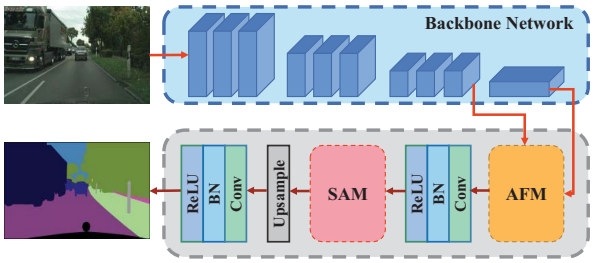

Figure 2: Illustration of the overall architecture. In the figure, ResNet18 is used as the backbone for exemplar illustration.

adopt shallow layers on the high-resolution image to speed up, while other branches have deep layers to obtain high-level semantics on low-resolution images. Besides, DFANet and LiteSeg (Emara, Abd El Munim, and Abbas 2019) adopt a lightweight backbone to speed up the inference. Different from these approaches, our model can work with large backbone networks while reducing computational complexity and reserving both semantic and spatial information.

Method

The overall network architecture of the proposed AttaNet is shown in Figure 2. As we can see, our AttaNet is a convolutional network that uses a cross-level aggregation architecture, which will be explained in the next subsection. Two key modules are then introduced respectively. To capture long-range relations efficiently, we propose the Strip Attention Module (SAM). And we introduce the Attention Fusion Module (AFM) where efficient feature aggregation is performed. Without loss of generality, we choose pre-trained ResNet (He et al. 2016) from ImageNet (Russakovsky et al. 2015) as our backbone by removing the last fully-connected layer, and other CNNs can also be chosen as the backbone.

Network Architecture

As shown in Figure 2, taking an image $I \in R^{3\times H\times W}$ as input, where $3, H, W$ indicate the RGB channels, height, width of I respectively, we first feed the image into the backbone network to generate convolutional features F_1, F_2 from the last layer of the res4 and res5 blocks respectively. Then F_1, F_2 are sent into Attention Fusion Module followed by a 3×3 convolution layer for feature smoothness. After that, we send the feature into Strip Attention Module, then resize it to 1/8 of the original image and pass it through a 3×3 convolutional layer to form the final features which are used to predict the pixel-wise segmentation results.

In SAM, we add a Striping layer before the Affinity operation to get the strongest consistency along anisotropy or banded context. Then we utilize the Affinity operation to find out the long-range relations in the horizontal direction to further enhance the consistency. Furthermore, in AFM, we use an attention strategy to make the model focus on the most relevant features as needed, which bridges the representation gap between multi-level features and enables effective information propagation.

For explicit feature refinement, we use deep supervision to get better performance and make the network easier to optimize. We use the principal loss function to supervise the output of the whole network. Moreover, we add two specific auxiliary loss functions to supervise the output of the res3 block and AFM. Finally, we use a parameter λ to balance the principal loss and the auxiliary loss:

$$L = l_p + \lambda \sum_{i=1}^{K} l_i \,,$$

where l_p is the principal loss of the final output. l_i is the auxiliary loss for the output of the res3 block and AFM. L is the joint loss. Particularly, all the loss functions are cross-entropy losses. K and λ are equal to 2 and 1 respectively in our implementation.

Strip Attention Module

In order to capture non-local contextual relations and also reduce the computational complexity in time and space, we introduce a module called Strip Attention Module. In particular, motivated by segmentation results in precious works (Fu et al. 2019a; Zhang et al. 2019), we apply a Striping operation to maintain contextual consistency along the vertical direction and further gather global affinity information between each pixel and banded structures along the horizontal axis. Figure 3 gives the detailed settings of Strip Attention Module.

More precisely, given an input feature map $F \in R^{C\times H\times W}$, where C is the number of channels, H and W are the spatial dimensions of the input tensor. We first feed F into two convolution layers with 1×1 filters to generate two new feature maps Q and K respectively, where $\{Q, K\} \in R^{C'\times H\times W}$. C' is less than C due to dimension reduction. We then apply a Striping operation on feature K to encode the global context representation in the vertical direction. Since the number of vertical strip areas is significantly larger than that of the horizontal ones in the natural images we are dealing with, the Striping operation represents average pooling with a pooling window of size $H \times 1$ in our work, and it can be extended to other directions for different purposes.

Then we reshape local features Q and K to $R^{C'\times N}$ and $R^{C'\times W}$ respectively, where $N = H \times W$ is the number of pixels. After that, we perform an Affinity operation between Q^T and K to further calculate the attention map $A \in R^{N\times W}$ along the horizontal direction. The Affinity operation is defined as follows:

$$A_{j,i} = \frac{exp(Q_i \cdot K_j)}{\sum_{i=1}^{N} exp(Q_i \cdot K_j)} \,,$$

where $A_{j,i} \in A$ denotes the degree of correlation between Q_i and K_j.

Meanwhile, we feed feature F into another convolutional layer with a kernel size of 1×1 to generate feature map $V \in R^{C\times H\times W}$. Similar to the above operation, for local feature V we can obtain a representation map in the vertical dimension whose spatial dimension is $C \times 1 \times W$

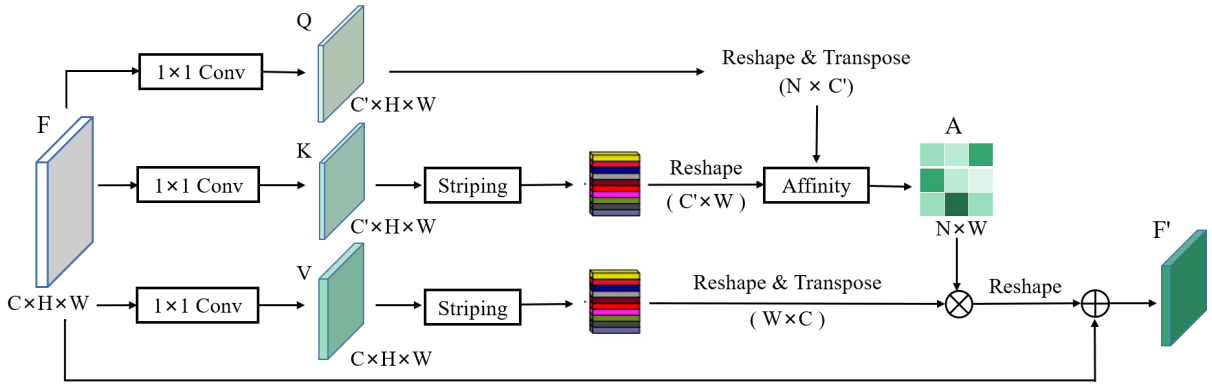

Figure 3: The details of Strip Attention Module.

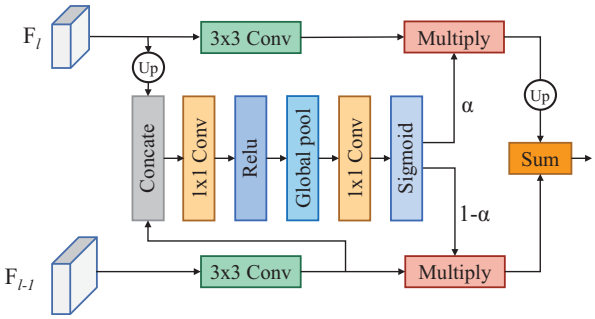

Figure 4: The details of Attention Fusion Module.

and reshape it to $V \in R^{C\times W}$. Then we perform a matrix multiplication between A and V^T, and reshape the result to $R^{C\times H\times W}$. Finally, we perform an element-wise sum operation with the input feature map F to obtain the final output $F' \in R^{C\times H\times W}$ as follows:

$$F'_j = \sum_{i=1}^{N} A_{j,i} \cdot V_i + F_j \,,$$

where F'_j is a feature vector in the output feature map F' at position j. The contextual information is added to the input feature map F to augment the pixel-wise representation ability especially for banded structures.

The benefits of our SAM are three-fold. First, since the striped feature map is the combination of all pixels along the same spatial dimension, this gives strong supervision in capturing anisotropy or banded context. Second, we first ensure that the relationships between each pixel and all columns are considered, and then estimate the attention map along the horizontal axis, thus our network can generate dense contextual dependencies. Moreover, this module adds only a few parameters to the backbone network, and therefore takes up very little GPU memory.

Attention Fusion Module

As stated before, feature fusion is widely used for incorporating multi-level representations. The most commonly used approaches for aggregation are like works (Yu et al. 2018c; Chen et al. 2018), i.e., first upsampling F_l via bilinear interpolation and then concatenating or adding the upsampled F_l and F_{l-1} together. However, low-level features contain excessive spatial details while high-level features are rich in semantics. Simply aggregating multi-level information would weaken the effectiveness of information propagation. To address this issue, we introduce Attention Fusion Module which enables each pixel to choose individual contextual information from multi-level features in the aggregation phase.

The detailed structure of our Attention Fusion Module is illustrated in Figure 4. Given two adjacent feature maps F_l and F_{l-1}, we first upsample F_l to the same size as F_{l-1} by the standard bilinear interpolation. Meanwhile, we feed F_{l-1} into a 3×3 convolutional layer (with BN and ReLU). Then the upsampled F_l is concatenated with the local feature F_{l-1} , and the concatenated features are fed to a 1×1 convolutional layer. After that, we leverage a global average pooling operation followed by a convolutional layer with kernel size of 1×1 to predict the relative attention mask α. After obtaining two attention maps, we further perform pixel-wise product between masks and the predictions followed by pixel-wise summation among them to generate the final results, i.e.,

$$Output = Sum(Upsample(F_l) \cdot \alpha, F_{l-1} \cdot (1 - \alpha)) \,.$$

This module employs the relative attention mask of adjacent features to guide the response of both features. This way, it bridges the semantic and resolution gap between multi-level features compared to the simple combination.

Experiments

To evaluate the proposed approach, we conducted extensive experiments on the Cityscapes dataset (Cordts et al. 2016) and the ADE20K dataset (Zhou et al. 2017). Experimental results demonstrate that AttaNet obtains leading performance on both Cityscapes and ADE20K. In the following subsections, we first introduce the datasets and implementation details, and then we carry out a series of comparisons and ablation experiments on the Cityscapes dataset. Finally, we report our results on the ADE20K dataset.

Datasets

Cityscapes. Cityscapes is a dataset for urban scene segmentation, which contains 5000 images with fine pixel-level annotations and 20000 images with coarse annotations. Each image has a resolution of 1024×2048 and contains 19 classes of semantic labels. The 5000 images with fine annotations are further divided into 3 subsets of 2975, 500, and 1525 images for training, validation, and testing, respectively.

ADE20K. ADE20K is a challenging scene parsing benchmark. The dataset contains 20K/2K images for training and validation which are densely labeled as 150 stuff/object categories. Images in this dataset are from different scenes with more scale variations.

Implementation Details

Our model utilizes ImageNet pre-trained ResNet18 (He et al. 2016) as the backbone. The last fully-connected layer is removed and the feature fusion method is applied between the output of the res4 block and res5 blocks.

Training Settings. We train the network using standard SGD (Krizhevsky, Sutskever, and Hinton 2012). The mini-batch size is set to 16 and 32 for Cityscapes and ADE20K respectively. And we use the momentum of 0.9 and a weight decay of $5e^{(-4)}$. Similar to other works(Chen et al. 2017; Yu et al. 2018b), we apply the 'poly' learning rate policy in which the initial learning rate is set to $1e^{(-2)}$ and decayed by $(1 - \frac{iter}{max_iter})^{power}$ with power=0.9. The training images are augmented by employing random color jittering, random horizontal flipping, random cropping, and random scaling with 5 scales $\{0.75, 1.0, 1.5, 1.75, 2.0\}$. For Cityscapes, images are cropped into size of 1024×1024, and the network is trained with 200k iterations. For ADE20K, crop size of 512×512 and 250K training iterations are used for training.

Inference. During the inference phase, we use a full image as input and follow the resizing method used in BiSeNetV2 (Yu et al. 2020). For quantitative evaluation, the standard metric of mean pixel intersection-over-union (mIOU) is employed for accurate comparison and performance measuring, frames per second (FPS), number of float-point operations (FLOPs), and the number of model parameters are adopted for speed comparison.

Experiments on Cityscapes

Comparisons with state of the art. In Table 1, we provide the comparisons between our AttaNet and the state-of-the-art real-time models. Our method is tested on a single GTX 1080Ti GPU with a full image of 1024×2048 as input which is resized into 512×1024 in the model. Then we resize the prediction to the original size and the time of resizing is included in the inference time measurement, which means the practical input size is 1024×2048. The speed is tested without any accelerating strategy and we only use train-fine data for training. As reported in Table 1, we get 70.1% mIoU with 180 FPS by stacking only eight convolution layers as the backbone network. As can be observed, the inference speed is significantly faster than that of the other models and the accuracy is comparable, which proves that even without heavy backbones our approach still achieves better performance than other approaches. Besides, our ResNet18 and DF2 based model achieves 130 FPS with 78.5% mIoU and 71 FPS with 79.9% mIoU respectively, which set the new state of the art on accuracy/speed trade-offs on the Cityscapes benchmark. It is worth mentioning that, with ResNet18 and DF2, the accuracy of our method even approaches the performance of the models that mainly focus on accuracy.

Approach	Backbone	mIoU / FPS
ICNet (Zhao et al. 2018)	ResNet50	69.5 / 34
BiSeNetV1 A (Yu et al. 2018a)	Xception39	68.4 / 105.8
SwiftNet (Orsic et al. 2019)	ResNet18	75.5 / 39.9
SwiftNet-ens (Orsic et al. 2019)	ResNet18	76.5 / 18.4
DFANet A (Li et al. 2019a)	Xception A	71.3 / 100
DFANet A' (Li et al. 2019a)	Xception B	70.3 / 160
DFANet B (Li et al. 2019a)	Xception B	67.1 / 120
DF1-Seg† (Li et al. 2019)	DF1	73.0 / 80.8
DF2-Seg† (Li et al. 2019)	DF2	74.8 / 55
CellNet (Zhang et al. 2019)	-	70.5 / 108
BiSeNetV2† (Yu et al. 2020)	-	72.6 / 156
BiSeNetV2† (Yu et al. 2020)	-	75.8 / 47.3
SFNet (Li et al. 2020)	DF1	74.5 / 74
SFNet (Li et al. 2020)	DF2	77.8 / 53
SFNet (Li et al. 2020)	ResNet18	78.9 / 18
SFNet‡ (Li et al. 2020)	ResNet18	**80.4** / 18
FANet-18 (Hu et al. 2020)	ResNet18	74.4 / 72
FANet-34 (Hu et al. 2020)	ResNet18	75.5 / 58
AttaNet	-	70.1 / **180**
AttaNet (ResNet18)	ResNet18	78.5 / 130
AttaNet (DF2)	DF2	79.9 / 71

Table 1: Comparison on the Cityscapes test set with state-of-the-art real-time models. † indicates that the model is tested using TensorRT for acceleration. ‡ indicates the model uses Mapillary dataset for pretraining.

SAM	AFM	mIoU (%)	Δ a	GFLOPs (Δ)
		72.8	-	-
√		77.1	4.3↑	0.185
	√	77.3	4.5↑	0.336
√	√	78.5	5.7↑	0.521

Table 2: Ablation study for the proposed modules on the Cityscapes validation set, where ResNet18 with feature aggregation architecture serves as the strong baseline.

To demonstrate the advantages of AttaNet, we provide the qualitative comparisons between AttaNet and the baseline in Figure 5. We use the red squares to mark the challenging regions. One can observe that the baseline network easily mislabels those regions but our proposed network is able to correct them, which clearly shows the effectiveness of AttaNet.

Ablation study on proposed modules. To verify the effectiveness of the proposed modules, we first conduct ablation experiments on individual components, namely Strip

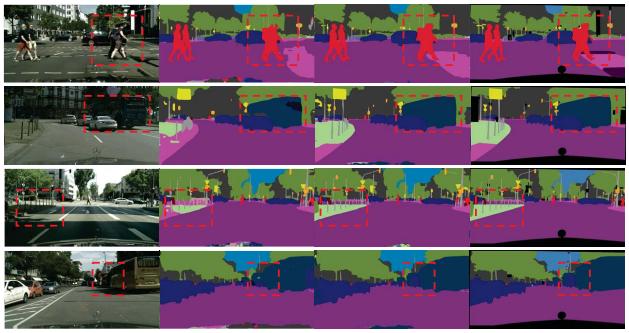

Figure 5: Visualization results of AttaNet on the Cityscapes validation set.

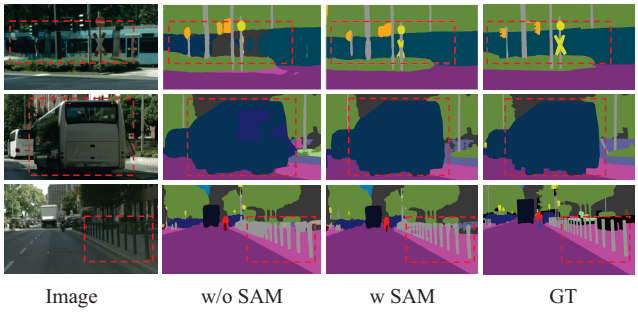

Figure 6: Qualitative comparison between our approach w/o and w/ SAM on the Cityscapes validation set.

Attention Module (SAM) and Attention Fusion Module (AFM). Without loss of generality, all results are obtained by training on the training set and evaluating on the validation set of Cityscapes. As shown in Table 2, the baseline network achieves 72.8% mIoU. By adding SAM, we get 77.1% mIoU by an improvement of 4.3%. Meanwhile, adding AFM brings 4.5% mIoU gain. Finally, we append SAM and AFM together, which further improves mIoU to 78.5%. All these improvements show that our modules bring great benefit to scene parsing. We also cropped some patches from some images in the Cityscapes val set, and show the comparison results in Figure 6 and Figure 7. We superimposed red squares to mark those challenging regions. While other methods easily mislabel those areas, the proposed modules are able to rectify misclassification results. Also, we can observe that SAM generates more consistent segmentation inside large objects or along the banded areas, while AFM can exploit more discriminative context for each class, and that is probably why AFM achieves slightly higher performance than SAM does.

We further conducted a series of comparison experiments on other algorithmic aspects. Specifically, we adopt the amount of computation, Memory usage, and mIoU score for comparison. As shown in Table 3, the top part compares the attention methods, and the bottom part compares the feature fusion methods. When given an input feature with a fixed size, SAM significantly reduces FLOPs by about 94.5%,

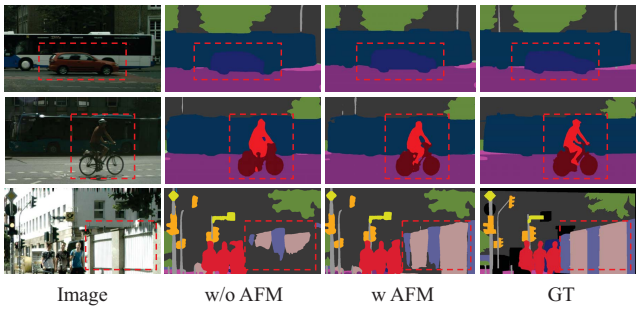

Figure 7: Qualitative comparison between our approach w/o and w/ AFM on the Cityscapes validation set.

Approach	mIOU	GFLOPs	Memory
Baseline	72.8	-	-
NL (Wang et al. 2018)	78.1	3.357	334M
RCCA (Huang et al. 2019)	77.7	0.472	26M
EMA (Li et al. 2019)	75.0	0.335	12M
SAM-horizontal (Ours)	76.9	0.185	8M
SAM-vertical (Ours)	77.1	0.185	8M
Addition (Baseline)	72.8	-	-
Concat	73.7	0.336	10M
AFM (Ours)	77.3	0.336	12M

Table 3: Comparison with other methods on the Cityscapes validation set, where ResNet18 with aggregation architecture is used as the baseline. GFLOPs (Δ) and Memory usage (Δ) are calculated for an input of $1 \times 3 \times 1024 \times 1024$.

60.8%, and 44.8% over NL, RCCA module in CCNet, and EMA unit in EMANet respectively. Compared with previous attention modules, our SAM achieves comparable segmentation performance while requiring significantly less GPU memory usage with both vertical striping (SAM-vertical) and horizontal striping (SAM-horizontal). Figure 8 shows several qualitative comparisons, where SAM generates more consistent segmentation inside the banded objects. Moreover, we visualize the learned attention maps of SAM in Figure 9. For each input image, we select two columns (marked as yellow and green dots) and show their corresponding attention maps in columns 2 and 3 respectively. The last two columns are results from our AttaNet and the ground truth. We can find that SAM is able to capture long-range dependencies. Moreover, from the bottom part of Table 3, we can observe that AFM achieves the best performance among the three methods with only slightly increased computation.

Robustness on different backbones. To show the generalization ability of AttaNet, we further carry out a set of comparison experiments on adopting different backbone networks including both heavy and light-weight ones. Note that AttaNet can be easily inserted into various backbone networks. For light-weight backbones, we select ShuffleNetV2 (Ma et al. 2018), DF1, and DF2 (Li et al. 2019) as the representatives. For really deep networks, ResNet50 and ResNet101 (He et al. 2016) are experimented on. Note that only ShuffleNetV2 and ResNet are pretrained on ImageNet. The comparison results are reported in Table 4, which

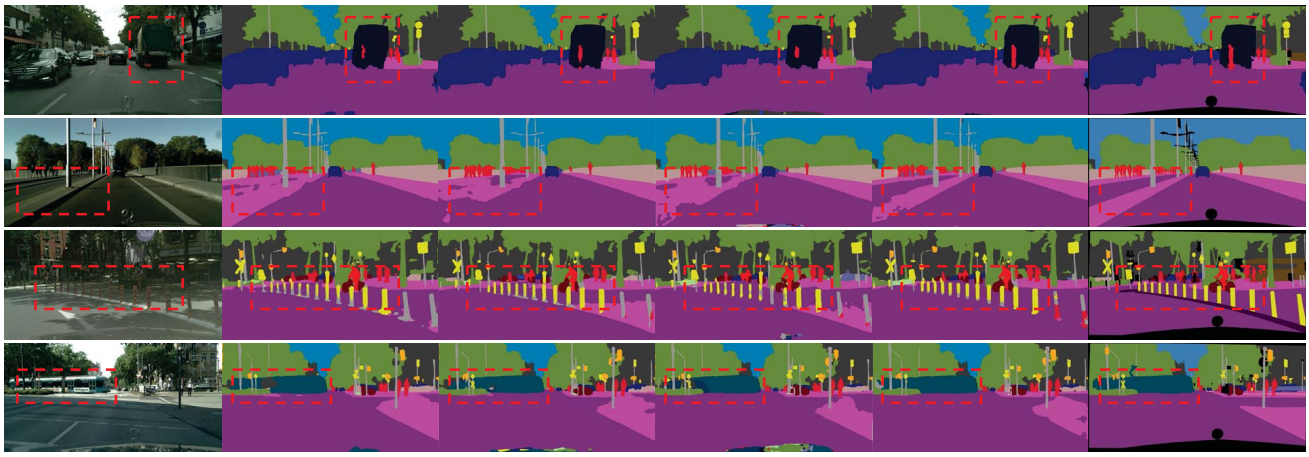

Figure 8: Qualitative comparison against different attention modules on the Cityscapes validation set.

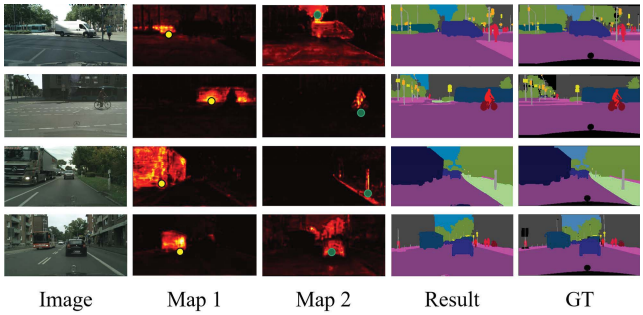

Figure 9: Visualization results of SAM on Cityscapes val set.

proves that our model can achieve considerably better mIoU on either heavy or light-weight backbones with only slightly increased computational cost.

Experiments on ADE20K

Table 5 reports the performance comparisons between AttaNet and the state-of-the-art models on the ADE20K validation set. Our approach achieves 41.79% mIoU and 43.71% mIoU respectively with much less computation.

Conclusions

In this paper, we focus on achieving a better speed/accuracy trade-off on the semantic segmentation task, and present an Attention-Augmented Network (AttaNet) for real-time scene parsing. First, we introduce Strip Attention Module to exploit long-range dependencies among all pixels. Particularly, by using the Striping operation, our network dramatically reduces the computation cost of the self-attention mechanism. Moreover, to attain a high-level and high-resolution feature map efficiently, we propose Attention Fusion Module which enables each pixel to choose private contextual information from multi-level features by utilizing the attention strategy. Experimental results show that AttaNet achieves outstanding speed/accuracy trade-offs on Cityscapes and ADE20K.

Backbone	mIoU	Params	GFLOPs
ResNet50 (He et al. 2016)	73.4	41.51M	171.36
+AttaNet	81.2	53.62M	176.91
ResNet101 (He et al. 2016)	74.5	65.80M	324.36
+AttaNet	81.0	78.96M	329.91
ShuffleNetV2 (Ma et al. 2018)	67.7	2.55M	12.10
+AttaNet	76.0	3.31M	12.59
DF1 (Li et al. 2019)	70.3	9.34M	25.92
+AttaNet	78.0	10.43M	26.44
DF2 (Li et al. 2019)	72.5	18.96M	48.74
+AttaNet	80.0	20.08M	49.23

Table 4: Ablation study on different backbones, where cross-level aggregation architecture is used as the baseline.

Approach	Backbone	mIoU/ GFLOPs
PSPNet (Zhao et al. 2017)	ResNet50	42.78 / 335.0
SFNet (Li et al. 2020)	ResNet50	42.81 / 151.1
AttaNet	ResNet50	41.79 / **116.3**
UperNet (Xiao et al. 2018)	ResNet101	42.66 / -
PSPNet (Zhao et al. 2017)	ResNet101	43.29 / 476.3
PSANet (Zhao et al. 2018)	ResNet101	43.77 / 529.3
SAC (Zhang et al. 2017)	ResNet101	44.30 / -
EncNet (Zhang et al. 2018)	ResNet101	44.65 / -
SFNet (Li et al. 2020)	ResNet101	44.67 / 187.5
CFNet (Zhang et al. 2019)	ResNet101	44.82 / -
CCNet (Huang et al. 2019)	ResNet101	45.22 / -
ACNet (Fu et al. 2019b)	ResNet101	45.90 / -
AlignSeg (Huang et al. 2020)	ResNet101	45.95 / -
AttaNet	ResNet101	43.71 / **150.5**

Table 5: Comparison on the ADE20K validation set with the state-of-the-art models.

Acknowledgments

This work is supported in part by funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society, and Shenzhen NSF JCYJ20190813170601651.

References

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

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

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733* .

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Emara, T.; Abd El Munim, H. E.; and Abbas, H. M. 2019. LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, 1–7. IEEE.

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

Fu, J.; Liu, J.; Wang, Y.; Li, Y.; Bao, Y.; Tang, J.; and Lu, H. 2019b. Adaptive context network for scene parsing. In *Proceedings of the IEEE international conference on computer vision*, 6748–6757.

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

Hou, Q.; Zhang, L.; Cheng, M.-M.; and Feng, J. 2020. Strip Pooling: Rethinking Spatial Pooling for Scene Parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4003–4012.

Hu, P.; Perazzi, F.; Heilbron, F. C.; Wang, O.; Lin, Z.; Saenko, K.; and Sclaroff, S. 2020. Real-time Semantic Segmentation with Fast Attention. *ArXiv* abs/2007.03815.

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

Huang, Z.; Wei, Y.; Wang, X.; Shi, H.; Liu, W.; and Huang, T. S. 2020. Alignseg: Feature-aligned segmentation networks. *arXiv preprint arXiv:2003.00872* .

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

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

Li, X.; You, A.; Zhu, Z.; Zhao, H.; Yang, M.; Yang, K.; and Tong, Y. 2020. Semantic Flow for Fast and Accurate Scene Parsing. *arXiv preprint arXiv:2002.10120* .

Li, X.; Zhao, H.; Han, L.; Tong, Y.; and Yang, K. 2019b. GFF: Gated Fully Fusion for Semantic Segmentation. *ArXiv* abs/1904.01803.

Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; and Liu, H. 2019. Expectation-Maximization Attention Networks for Semantic Segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9167–9176.

Li, X.; Zhou, Y.; Pan, Z.; and Feng, J. 2019. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, 9145–9153.

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

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, 116–131.

Mou, L.; Hua, Y.; and Zhu, X. X. 2019. A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 12416–12425.

Orsic, M.; Kreso, I.; Bevandic, P.; and Segvic, S. 2019. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 12607–12616.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115(3): 211–252.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified Perceptual Parsing for Scene Understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 432–448.

Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; and Sang, N. 2020. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation. *arXiv preprint arXiv:2004.02147* .

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

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

Yu, F.; Wang, D.; Shelhamer, E.; and Darrell, T. 2018c. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2403–2412.

Yuan, Y.; and Wang, J. 2018. OCNet: Object Context Network for Scene Parsing. *ArXiv* abs/1809.00916.

Zhang, F.; Chen, Y.; Li, Z.; Hong, Z.; Liu, J.; Ma, F.; Han, J.; and Ding, E. 2019. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 6798–6807.

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018. Context Encoding for Semantic Segmentation. *arXiv preprint arXiv:1803.08904* .

Zhang, H.; Zhang, H.; Wang, C.; and Xie, J. 2019. Co-Occurrent Features in Semantic Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 548–557.

Zhang, R.; Tang, S.; Zhang, Y.; Li, J.; and Yan, S. 2017. Scale-Adaptive Convolutions for Scene Parsing. 2050–2058. doi:10.1109/ICCV.2017.224.

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

Zhao, H.; Qi, X.; Shen, X.; Shi, J.; and Jia, J. 2018. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 405–420.

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

Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C. C.; Lin, D.; and Jia, J. 2018. PSANet: Point-wise Spatial Attention Network for Scene Parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 270–286.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene Parsing Through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.