AttNAS: Searching Attentions for Lightweight Semantic Segmentation

Yingying Jiang
yy.jiang@samsung.com
Zhuoxin Gan
zhuoxin1.gan@samsung.com
Ke Lin
ke17.lin@samsung.com
Yong A
yong.a@samsung.com

Samsung Research China-Beijing (SRC-B)
Beijing, China

Abstract

To use multiple attentions in semantic segmentation task, attentions are generally arranged in parallel and then combined by concatenation or summation. In this work, we use Neural Architecture Search (NAS) to explore attentions and their combination patterns (e.g., parallel, hybrid parallel+sequential) in semantic segmentation. We propose AttNAS, which searches both backbone and attention for a lightweight semantic segmentation model. In particular, we define a new attention search space with a two-layer structure and propose a unified differentiable formulation to support both attention type search and attention combination search. Our experimental results show that the model searched with AttNAS achieves state-of-the-art compared to existing lightweight methods on Cityscapes, CamVid and Pascal VOC 2012. Moreover, the attention patterns obtained by AttNAS have robust generalization capability and can be used in combination with existing backbones, such as MobileNetV2 and ResNet18, to improve segmentation performance.

1 Introduction

In semantic segmentation, there are mainly two modules. One is the backbone for feature extraction, and the other is the attention module, which incorporates the contextual information to obtain the semantic segmentation result. Currently, researchers of semantic segmentation focus more on the attention module [1, 3, 5, 7, 9, 23, 24, 25, 28, 30, 31]. Early works generally use local regional attentions to get the contextual information, such as the PSP module in PSPNet [28] and the ASPP module in Deeplab [1]. Although these local attentions contribute to the improvement of the segmentation performance, they are inferior in segmenting large objects due to the lack of global context information. With the rapid development of self-attention and graph neural network, long-range global attentions have been used in semantic segmentation, such as GCN [23], Strip Pooling [5], EMA [9] and Dual attention [3]. They are good at capturing long-range context to get a global understanding of the image, but they may overlook local detailed contextual information. Some researchers try to combine both local and global attentions [3, 5]. Their methods employ different attentions in parallel after the feature extraction backbone, which may not be the ideal way to combine them. In this paper, we try to find a better attention module for semantic segmentation with the assistance of Neural Architecture Search (NAS).

Gradient based NAS [14] has been used to discover high performance network architectures in semantic segmentation task [2, 11, 12, 13, 21, 26, 27]. These methods search different type of modules of the network. For example, Auto-Deeplab [13] searches both network level and cell level structure. SparseMask [21] learns the connectivity structure for the decoder. GAS [12], DF-Seg [11] and FasterSeg [2] search in a lightweight search space for real-time semantic segmentation. However, none of them explores the specific attention module for semantic segmentation.

In this work, we propose a NAS-based semantic segmentation method AttNAS (Figure 1), of which the search space consists of both backbone and attention modules. Instead of manually designing a new attention module, AttNAS searches for a combination of existing carefully designed attention modules. To obtain a lightweight module, we use hardware constraints (e.g., FLOPs) to guide the gradient based search process. Experimental results show that our AttNAS is effective in discovering lightweight semantic segmentation architectures and that our searched model achieves state-of-the-art performance on benchmarks.

Our contributions are as follows:

- We are the first to explore attention architecture search specifically for semantic segmentation.
- We propose AttNAS, in which a new search space is defined to explore combination patterns of existing attention modules and a unified differentiable formulation way is proposed to support attention architecture derivation.
- AttNAS attains state-of-the-art performance compared with existing methods that have similar FLOPs on Cityscapes, CamVid and Pascal VOC 2012. Moreover, the attention designs searched by AttNAS have superior generalization ability.

2 Related Work

2.1 Attentions in Semantic Segmentation

Attention mechanisms in semantic segmentation have been studied by many researchers. Local attentions are studied in early work[1, 28]. PSPNet [28] captures context information by pooling at different sizes. Deeplab[1] designs an ASPP module by doing atrous convolution at different dilation rates. Although these attentions have fused multi-scale features, they can only get neighborhood local context information. Recently, researchers focus more on studying global attentions[3, 5, 7, 9, 23, 25, 30, 31], methods such as self-attention and graph neural network are utilized. While PAN[7], PSANet[30], EncNet[25], DANet[3] and OCRNet [24] are methods for extracting global context information, EMA[9], SPNet[5] and representative graph neural network[23] focus on lightweight global attentions. With so many attentions, some researchers try to combine different types of attentions to get even better context information, such as DANet[3] and SPNet[5].

Although these attentions are useful in modeling the context information for semantic segmentation, current methods simply place attentions in parallel when combining multiple attentions, and there's no research on how these attentions should work together better.

2.2 NAS in Semantic Segmentation

NAS has been used in semantic segmentation task to discover good network architectures[2, 11, 12, 13, 21, 27]. Researchers use gradient based method [14] to search different modules of the network. As semantic segmentation is sensitive to resolution, some researchers design their resolution-related search space[2, 13, 21, 26]. Auto-Deeplab [13] searches both network level and cell level structure and tries to find the positions of down-sample and up-sample at network level search. SparseMask [21] learns the connectivity structure for the decoder. DCNAS[26] searches the optimal network structures from the densely connected search space without proxy.

Hardware-aware NAS has been utilized to find an architecture that is both accurate and fast. In semantic segmentation, GAS [12], DF-Seg[11] and FasterSeg [2] adopt hardware-aware approaches. They search with lightweight search space and use latency constraints in their search process for real-time semantic segmentation.

Although existing NAS-based semantic segmentation methods have achieved good performance, none of them explores the attention module in semantic segmentation. AttentionNAS [20] first attempts to extend NAS to discover attention cell in video classification, but it focuses on spatiotemporal attention which cannot be used in semantic segmentation. Our idea is to use NAS to discover good attentions and find whether there's better attention combination patterns than parallel attentions.

3 Methodology

3.1 Motivation and Overview

Currently, various kinds of attention are developed for semantic segmentation. The different attentions have different properties: global attentions is superior in handling global long-range contexts; local attentions focus more on detailed contexts. To account for both local and global context, multiple attention modules are often simply placed in parallel. It is not certain which attentions are important and whether this parallel style is the best way to organize these attentions.

NAS can pick up good architectures from a huge set of candidate architectures. It can be used to explore attentions in semantic segmentation. Thus, we adopt a NAS based approach to find good attention designs. Figure 1 shows the architecture of our method AttNAS. It aims to search both the backbone and attentions for a lightweight semantic segmentation architecture. Specifically, we design a dedicated search space and a unified differentiable formulation method to explore the combination patterns of attention.

3.2 Search Space

As shown in Figure 1, the search space of AttNAS consists of two parts: backbone search space and attention search space.

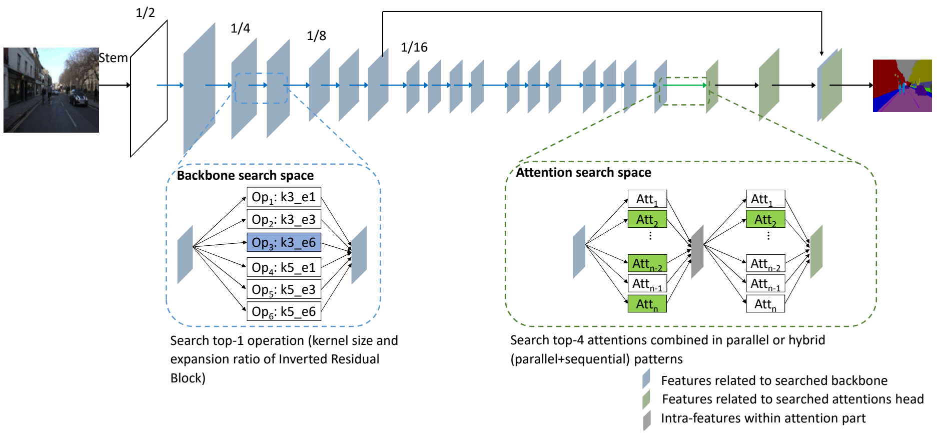

1/2
1/4
1/8
1/16
Stem
Backbone search space
Op$_1$: k3_e1
Op$_2$: k3_e3
Op$_3$: k3_e6
Op$_4$: k5_e1
Op$_5$: k5_e3
Op$_6$: k5_e6
Search top-1 operation (kernel size and expansion ratio of Inverted Residual Block)
Attention search space
Att$_1$
Att$_2$
Att$_{n-2}$
Att$_{n-1}$
Att$_n$
Search top-4 attentions combined in parallel or hybrid (parallel+sequential) patterns
Features related to searched backbone
Features related to searched attentions head
Intra-features within attention part

Figure 1: **The network architecture of AttNAS.** We search for both backbone and attention. The backbone search space is composed of kernel sizes and expansion ratios in MobileNetV2 search space. The attention search space has a two-layer structure and each layer consists of local spatial attention, global spatial attention and channel wise attention. AttNAS will find top-4 attentions from two layers, thus their combination pattern is also found.

Backbone search space. As our goal is to get a lightweight model, our backbone search space is based on MobileNetV2. In our implementation, we search the operations in 17 inverted residual blocks in MobileNetV2. Within each block, we search the kernel size (3,5) and expansion ratios (1,3,6). Thus, there are six candidate operations in each block as shown in Figure 1.

Attention search space. AttNAS searches attentions and their combination patterns. The candidate attentions consist of n human designed attentions including local regional attentions, global long-range attentions and channel wise attentions.

In our implementation, n is 13 and there are totally 13 candidate attentions in the attention search space. Table 1 shows the details of the attentions. Att1-Att9 are local regional attentions. Att1-Att5 are attention modules inspired by the design of PSP module in PSPNet[28], which is composed of pool, conv-bn and upsample. Att1-Att5 have different pooling sizes. Att6-Att9 are atrous convolutions with different dilation rates and their idea comes from ASPP module in Deeplab [1]. Att10-Att12 are global long-range attentions. Att10 is strip pooling designed by [5]. Att11-Att12 are self-attentions [19] with a pool operation in front and an upsample operation behind. Att11 and Att12 are different as they have different input sizes for self-attention. Att13 is the channel-wise attention. It is inspired by the Squeeze and Excitation block [6]. The FLOPs in table 1 is obtained when the input image size is 1024×2048. The reason why we do not use ASPP module and PSP module directly in the search space is that we want to explore more attention combination patterns at more basic attention modules. The original ASPP module and PSP module consist of several similar parallel attention sub-modules.

As shown in Figure 1, we stack two layers of attention candidates as the total attention search space. This search space includes various combination patterns of the attentions. For example, if we select four attentions, it has the following possibilities: a) four attentions in layer1; b) three attentions in layer1, one attention in layer2; c) two attentions in layer1, two

Type	From	Operations	FLOPs	Attentions
Local attention (spatial)	PSP[28]	Pool1_conv1bn_upsample	2.7M	Att1~Att5
		Pool2_conv1bn_upsample	3.0M	
		Pool3_conv1bn_upsample	3.4M	
		Pool6_conv1bn_upsample	6.1M	
		Pool9_conv1bn_upsample	10.9M	
	ASPP[1]	AtrousConv1	841M	Att6~Att9
		AtrousConv6	7552M	
		AtrousConv12	7552M	
		AtrousConv18	7552M	
Global attention (spatial)	SP [5]	HoriPool_VertPool_conv3bn	7619M	Att10
	SA[19]	Pool6_SelfAttention_upsample	2.4M	Att11~Att12
		Pool9_SelfAttention_upsample	2.5M	
Channel wise attention	SE[6]	Pool1_fc_multiply	2.6M	Att13

Table 1: Attention Candidates in Attention Search Space. Existing PSP[28] and ASPP[1] modules are decoupled into several attention candidates.

attentions in layer2; d) one attention in layer1, three attentions in layer2; e) four attentions in layer2. It includes both the parallel combination pattern and hybrid parallel+sequential combination pattern. In total, there are about 2.5×10^{17} candidate network architectures in the whole search space.

3.3 Search Algorithm

Differential architecture search treats architecture search as a bi-level optimization problem and searches an architecture by gradient descent in a short time[14]. AttNAS adopts gradient-based search and it uses a new unified differentiable formulation way for attention search.

3.3.1 Differentiable formulation of search space

We use (α,β) to represent architecture parameters. α is related to the possibility of choosing operations in the backbone search space and β is related to the possibility of choosing attention operations in the attention search space.

For backbone search, there are 17 searchable blocks in MobileNetV2 (Figure 1). For each backbone search block i, all the operations' probabilities in this block sum to one. Let x_{i-1} be the input feature map for block i and x_i be the output feature map after block i. x_i is obtained by weighted combining all possible operations over x_{i-1}.

$$x_i = \sum_{j=1...6} \frac{exp(\alpha_{ij})}{\sum_{k=1...6} \exp(\alpha_{ik})} Op_j(x_{i-1}) \tag{1}$$

where α_{ij} corresponds to the possibility of choosing the jth operation in block i and Op_j is the jth operation in block i.

For attention search space, it includes the attention candidates from two layers $AttSet_1$ and $AttSet_2$. In our implementation, both $AttSet_1$ and $AttSet_2$ have 13 attention candidates. Let x_{in} be the input feature map to the attention module and x_{out} be the output feature map from the attention module. Att_j is the candidate attention operation defined in Table 1. To make the attention search space continuous, we relax the categorical choice of a particular attention to a softmax over all possible attentions in $AttSet_1$ and $AttSet_2$. It implies that the probability of all attention candidates in $AttSet_1$ and $AttSet_2$ sum to one.

$$\begin{aligned} x' &= \sum_{j=1...13} \frac{exp(\beta_{1j})}{\sum_{i=1,2,k=1...13} \exp(\beta_{ik})} Att_j(x_{in}) \\ x_{out} &= \sum_{j=1...13} \frac{exp(\beta_{2j})}{\sum_{i=1,2,k=1...13} \exp(\beta_{ik})} Att_j(x') \end{aligned} \tag{2}$$

where β_{ij} is related to the probability of choosing jth attention in attention set $AttSet_i$, and x' is the output after performing attentions in $AttSet_1$ on x_{in}.

This unified differentiable formulation design makes searching attention combination patterns possible, because all the architecture parameters for attentions in $AttSet_1$ and $AttSet_2$ are normalized in a unified way. After finishing the search process, we may get important attentions only from $AttSet_1$, only from $AttSet_2$, or from both of them.

3.3.2 Architecture parameter learning

We optimize the architecture parameters with a bi-level optimization process [14]. We alternately optimize network weights w and architecture parameters (α, β) .

As our goal is to find a lightweight architecture with high performance, we should consider both the semantic segmentation performance and its computation cost. In our implementation, we use floating point operations (FLOPs) to measure the computation cost of a network as it is not influenced by hardware and software environment. If a model has small FLOPs, its latency is often small.

The total loss function is designed as follows:

$$L(\alpha, \beta, w) = L_{ce}(\alpha, \beta, w_{\alpha,\beta}) + \lambda L_{flops}(\alpha, \beta) \tag{3}$$

where $L_{ce}(\alpha, \beta, w_{\alpha,\beta})$ denotes the cross entropy loss of architecture (α, β) with parameter $w_{\alpha,\beta}$, L_{flops} denotes the overall floating point operations of architecture (α, β), and λ controls the balance between accuracy and computation cost.

L_{flops} is defined as follows:

$$L_{flops} = \frac{(FLOPs_{pred}(\alpha, \beta) - FLOPs_{target})}{FLOPs_{target}} \tag{4}$$

where $FLOPs_{target}$ is the target FLOPs of the lightweight model, $FLOPs_{pred}$ is the estimated FLOPs with (α, β). For backbone and attention search space, the FLOPs is estimated by weighted summation of candidate operations' FLOPs which is similar to the process of formula(1) and formula(2).

3.3.3 Architecture derivation

After optimizing the architecture parameters, we derive the final semantic segmentation model. For the backbone part, we select the operation j with the largest value in α_i. The blue box in Figure 1 shows one derived path in the backbone.

For the attention part, we select the top-4 attentions from $AttSet_1$ and $AttSet_2$ according to β. The green boxes in Figure 1 illustrate one derived attention structure. It contains three attentions in the first layer and one attention in the second layer.

As defined in formula(2), each $\beta_{i,j}$ is normalized by all $\beta_{1,j}$ ($AttSet_1$) and all $\beta_{2,j}$ ($AttSet_2$). We can just select the top-4 $\beta_{i,j}$ values by $topk_{i\in 1,2,j\in 1...13}(\beta_{i,j})$ and then find their corresponding attentions. The combination patterns are selected automatically. Assume the attention corresponding to $\beta_{i,j}$ is selected, if i equals to 1, it will be placed in the first layer. If i equals to 2, it will be placed in the second layer.

4 Experiments

4.1 Datasets

Our experiments are conducted on Cityscapes, CamVid and Pascal VOC 2012. While Cityscapes and CamVid are two outdoor driving datasets, Pascal VOC 2012 is a common scene dataset. Cityscapes contains 5,000 fine annotated images with resolution 1024×2048. They are split into training, validation and testing sets with 2,979, 500 and 1,525 images respectively. There are 30 classes in Cityscapes and 19 of them are used for training and evaluation in semantic segmentation. CamVid contains images extracted from video sequences with resolution up to 720×960. It contains 701 images in total, including 367 for training, 101 for validation and 233 for testing. There are 11 classes in the dataset. Pascal VOC 2012 includes 20 object categories and one background class. We use the original dataset in our experiments. It contains 1464 images for training and 1449 images for validation.

4.2 Implementations Details

We use Adam optimizer to optimize architecture parameters and SGD to optimize network weights. We do warm-up training for 30 epochs without architecture parameter optimization, and then search for 100 epochs with alternate optimization of architecture parameters and network weights. The initial learning rates for architecture parameters and network weights are set to 0.0003 and 0.01. The image size is (224,448). Half of the images in Cityscapes training set are used as training set and the other half are used as validation set.

When we get the searched architectures, we train them from scratch for 200 epochs with Cityscapes training set (2979 images). The initial learning rate is 0.01, momentum is 0.9 and weight decay is 0.0005. The image size for training is (512,1024). After training, we can get the performance on Cityscapes validation dataset. We further train the model with both the training and validation set to get the result on Cityscapes test set. The image size for evaluation is the same with the original image size (1024,2048). After that, we transfer the searched architecture on Cityscapes to CamVid and Pascal VOC 2012 by only adjusting the output class number to 11 and 21.

4.3 Ablation Studies

4.3.1 Effectiveness of Attention Search and Backbone Search

We first validate the effectiveness of our design on Cityscapes validation set. Table 2 shows the results of models searched with different attention search spaces: 1) searching AttSet1 with one layer attentions (s-mv2-att1); 2) searching AttSet1 and AttSet2 with two layer attentions (s-mv2-att2); 3) searching AttSet1, AttSet2 and AttSet3 (same attention candidates

Method	Backbone	Attention	mIoU(%)	FLOPs	#Params
baseline(mv2+aspp)	mv2	ASPP[1]	73.4	41.4G	4.5M
s-mv2-att1	mv2(2,3,5,6,4,6,3,1,1,4,3,1,1,6,1,1,6)	AttSet1(7,8,11,12)	72.2	30.9G	2.9M
s-mv2-att2(AttNAS-S)	mv2(3,3,2,3,2,3,2,3,2,6,3,6,5,3,2,5,1)	AttSet1(12,13)+AttSet2(3,7)	73.8	**28.2G**	**2.3M**
s-mv2-att2(AttNAS-L)	mv2(2,2,2,3,3,3,3,1,3,2,3,1,1,3,3,2,2)	AttSet1(7,8,13)+AttSet2(7)	**74.4**	40.6G	4.2M
s-mv2-att3	mv2(1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,2,1)	AttSet1(8,13)+AttSet2(8)+AttSet3(7)	67.2	29.8G	3.4M

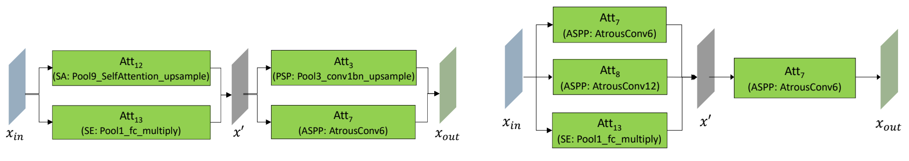
Table 2: Results with Different Searching Spaces on Cityscapes Validation Set. mv2+aspp is the baseline method based on MobileNetV2[18] backbone and ASPP[1]. In the Backbone column, the numbers after mv2 indicate the index of the selected operation in each backbone block as defined in Figure 1. In the Attention column, the numbers after AttSet1 and AttSet2 refer to the indexes of the selected attentions as defined in Table 1 and Figure 1.

with AttSet1 and AttSet2) with three layer attentions (s-mv2-att3); AttNAS-S and AttNAS-L are two searched models by s-mv2-att2 with 30G and 40G FLOPs constraints. All the attention search spaces share the same backbone search space.

Results show that our AttNAS can get higher performance than the baseline mv2+aspp model with smaller FLOPs and parameters. While the mIoU of the baseline model is 73.4% with 41.4G FLOPs and 4.5M parameters, our AttNAS-S achieves 73.8% with 28.2G FLOPs and 2.3M parameters and our AttNAS-L has 74.4% with 40.6G FLOPs and 4.2M parameters.

Regarding attention search space, we find that our two-layer attention search space is the best, as both s-mv2-att2 with different FLOPs are better than s-mv2-att1 and s-mv-att3. The results show that three layer attention space(s-mv2-att3) is even worse than one layer space, we suppose that it is too hard to search top-4 attentions from too many candidates(39 in s-mv2-att3). The searched attentions by s-mv2-att2 all include attentions from both AttSet1 and AttSet2. This means that our search algorithms prefer hybrid ways (parallel+sequential) to organize attentions other than the parallel style used in existing methods [1, 5, 28].

4.3.2 Analysis of Searched Attentions


x_{in}
Att$_{12}$ (SA: Pool9_SelfAttention_upsample)
Att$_{13}$ (SE: Pool1_fc_multiply)
x'
Att$_3$ (PSP: Pool3_conv1bn_upsample)
Att$_7$ (ASPP: AtrousConv6)
x_{out}
x_{in}
Att$_7$ (ASPP: AtrousConv6)
Att$_8$ (ASPP: AtrousConv12)
Att$_{13}$ (SE: Pool1_fc_multiply)
x'
Att$_7$ (ASPP: AtrousConv6)
x_{out}

(a) Searched Attention in AttNAS-S. (b) Searched Attention in AttNAS-L.

Figure 2: Illustration of two searched attentions by AttNAS. They both compose of attentions from AttSet1 and AttSet2.

Figure 2(a) demonstrates the attention found in AttNAS-S. It consists of global spatial attentions (Att12), local spatial attentions (Att3 and Att7) and global channel-wise attentions(Att13). In particular, two global attention modules (Att12 and Att13)are followed by two local attention module (Att3 and Att7). It is reasonable that the global attention modules in the first layer capture the long-range context information and then the local attention modules in the second layer further explore their dependencies.

Figure 2(b) shows the attention found in AttNAS-L. It consists of local attentions(Att7 and Att8) and channel-wise attentions(Att13). The first layer includes global channel-wise attention (Att13) and two local attentions based on Atrous convolution (Att7 and Att8). The second layer is a local attention based on Atrous convolution (Att7). Similarly, global attentions appear in the first layer and local attention follows in the second layer. This finding can help guiding the design of attention modules.

4.3.3 Generalization of Searched Attentions

(a) MobileNetV2

Method	mIoU(%)	FLOPs	#Params
mv2+aspp	73.4	41.4G	4.5M
mv2+attS	73.6	**29.8G**	**3.1M**
mv2+attL	**74.7**	44.9G	4.8M

(b) ResNet18

Method	mIoU(%)	FLOPs	#Params
res18+aspp	69.7	80.0G	5.0M
res18+attS	69.4	**68.7G**	**3.6M**
res18+attL	**71.0**	78.4G	4.7M

Table 3: Results of Searched Attentions with MobileNetV2 and ResNet18.

We further verify the effectiveness of searched attentions on existing backbones. We replace the searched backbones to MobileNetV2 [18] and ResNet18 [4] in AttNAS-S and AttNAS-L. We get four models: mv2+attS, mv2+attL, res18+attS, res18+attL. We compare them with baseline mv2+aspp and res18+aspp. Table 3 shows the results of searched attentions with MobileNetV2 and ResNet18 on Cityscapes validation set. We can see that compared with baseline mv2+aspp method, mv2+attS can get higher performance with fewer FLOPs and parameters. mv2+attL can bring 1.3% performance improvement over mv2+aspp. As for the ResNet18 backbone, res18+attS gets similar performance with res18+aspp with fewer FLOPs and parameters. res18+attL can bring 1.6% performance improvement over res18+aspp.

Although attS and attL are obtained by jointly searching both backbone and attention space, they can work well with MobileNetV2 and ResNet18. The attentions found by AttNAS have good generalization ability.

4.4 Comparison with State-of-the-art

We compare our method with state-of-the-art methods on Cityscapes, CamVid and Pascal VOC 2012. Table 4 shows our comparison results. While ENet[16],BiSeNet[22], Fast-SCNN[17], ICNet[29], DFANet[8] and SFNet[10] are human designed lightweight semantic segmentation methods, CAS[27], GAS[12], DF1-Seg-d8[11] and FasterSeg[2] are NAS-based methods for lightweight semantic segmentation. Our AttNAS-S achieves a mIoU of 73.3% on Cityscapes test set and a mIoU of 73.8% on Cityscapes validation set, which are better than SOTA methods with similar FLOPs and Params. In particular, its mIoU on the test set is 1.8% higher than that of the NAS-based method FasterSeg [2] which has similar FLOPs. Moreover, our AttNAS-L further improves the performance to 74% on the test set with a larger FLOPs.

We transfer the architecture to CamVid and Pacal VOC 2012. The CamVid test set results in Table 4 are obtained at 720×960 input resolution.The Pascal VOC val results are obtained at 512×512 input resolution. Our methods achieve state-of-the-art results on both CamVid and Pascal VOC 2012.

(a) Cityscapes

Method	NAS based	InputSize	mIoU(%) val	mIoU(%) test	FLOPs	Params
ENet[16]		640×360	-	58.3	3.8G	**0.4M**
BiSeNet[22]		768×1536	69	68.4	14.8G	5.8M
Fast-SCNN[17]		1024×2048	68.6	68.0	-	1.1M
ICNet[29]		1024×2048	-	69.5	28.3G	26.5M
DFANet A[8]		1024×1024	-	71.3	**3.4G**	7.8M
SFNet(DF1)[10]		1024×2048	-	**74.5**	-	9.03M
GAS[12]	✓	769×1537	-	71.8	-	-
CAS[27]	✓	768×1536	71.6	70.5	-	-
DF1-Seg-d8[11]	✓	1024×2048	72.4	71.4	-	-
FasterSeg[2]	✓	1024×2048	73.1	71.5	28.2G	4.4M
AttNAS-S	✓	1024×2048	73.8	73.3	28.2G	2.3M
AttNAS-L	✓	1024×2048	**74.4**	74	40.6G	4.2M

(b) CamVid

Method	NAS based	mIoU(%)	FLOPs
ENet[16]		51.3	-
DFANet A[8]		64.7	-
BiSeNet[22]		65.6	**8.7G**
ICNet[29]		67.1	-
SFNet(DF2)[10]		70.4	-
FasterSeg[2]	✓	71.1	9.3G
CAS[27]	✓	71.2	-
GAS[12]	✓	72.8	-
AttNAS-S	✓	72.8	9.2G
AttNAS-L	✓	**73.8**	13.3G

(c) Pascal VOC 2012

Method	NAS based	mIoU(%)	FLOPs
baseline(mv2+aspp)		70.6	5.5G
FCN(mv2)[15]		63.8	-
SparseMask[21]	✓	73.18	-
AttNAS-L	✓	**73.3**	5.4G

Table 4: Comparison with State-of-the-art Lightweight Semantic Segmentation Methods.

5 Conclusion

In this paper, we propose a NAS-based semantic segmentation method AttNAS, which searches both backbone and attention architecture for a lightweight semantic segmentation model. Since attentions can capture context information and are important for semantic segmentation, we design a dedicated search space with a two-layer structure to support searching both attentions and the combination patterns of attentions. We perform a gradient-based search with a loss function that includes semantic segmentation loss and FLOPs loss. Results show that our method can find a lightweight architecture that can achieve state-of-the-art performance with low FLOPs on Cityscapes, CamVid, and Pascal VOC 2012. Moreover, the attention designs found by AttNAS have good generalization ability. They work well with existing backbone, such as MobileNetV2[18] and ResNet18[4], and achieve better results than the ASPP module[1].

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[2] Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang

Wang. Fasterseg: Searching for faster real-time semantic segmentation. In *International Conference on Learning Representations*, 2020.

[3] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4003–4012, 2020.

[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[7] H. Li, P. Xiong, J. An, and L. Wang. Pyramid attention network for semantic segmentation. 2018.

[8] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian 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, 2019.

[9] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu. Expectation-maximization attention networks for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9166–9175, 2019. doi: 10.1109/ICCV.2019.00926.

[10] Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision*, pages 775–793. Springer, 2020.

[11] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9145–9153, 2019.

[12] Peiwen Lin, Peng Sun, Guangliang Cheng, Sirui Xie, Xi Li, and Jianping Shi. Graph-guided architecture search for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2020.

[13] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 82–92, 2019.

[14] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[16] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[17] RPK Poudel, S Liwicki, and R Cipolla. Fast-scnn: Fast semantic segmentation network. In *30th British Machine Vision Conference 2019, BMVC 2019*, 2019.

[18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[20] Xiaofang Wang, Xuehan Xiong, Maxim Neumann, AJ Piergiovanni, Michael S Ryoo, Anelia Angelova, Kris M Kitani, and Wei Hua. Attentionnas: Spatiotemporal attention cell search for video classification. In *European Conference on Computer Vision*, pages 449–465. Springer, 2020.

[21] Huikai Wu, Junge Zhang, and Kaiqi Huang. Sparsemask: Differentiable connectivity learning for dense image prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6768–6777, 2019.

[22] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[23] Changqian Yu, Yifan Liu, Changxin Gao, Chunhua Shen, and Nong Sang. Representative graph neural network. In *European Conference on Computer Vision*, pages 379–396. Springer, 2020.

[24] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[25] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. doi: 10.1109/CVPR.2018.00747.

[26] Xiong Zhang, Hongmin Xu, Hong Mo, Jianchao Tan, Cheng Yang, Lei Wang, and Wenqi Ren. Dcnas: Densely connected neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13956–13967, 2021.

[27] Yiheng Zhang, Zhaofan Qiu, Jingen Liu, Ting Yao, Dong Liu, and Tao Mei. Customizable architecture search for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11641–11650, 2019.

[28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[29] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018.

[30] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Springer, Cham*, 2018.

[31] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, and A. Wong. Squeeze-and-attention networks for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.