Pruning Parameterization with Bi-level Optimization for Efficient Semantic Segmentation on the Edge

WED-PM-290

Changdi Yang*[1] Pu Zhao*[1] Yanyu Li[1] Wei Niu[2] Jiexiong Guan[2] Hao Tang[3] Minghai Qin[1] Bin Ren[2] Xue Lin[1] Yanzhi Wang[1]
[1]Northeastern University [2]College of William & Mary [3]CVL, ETH Zurich

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Preview

- We aim to derive ViTs with fewer computations and fast inference speed
- We propose a pruning parameterization method to formulate the pruning problem of semantic segmentation.
- We adopt a bi-level optimization method to solve this problem with the help of implicit gradients.

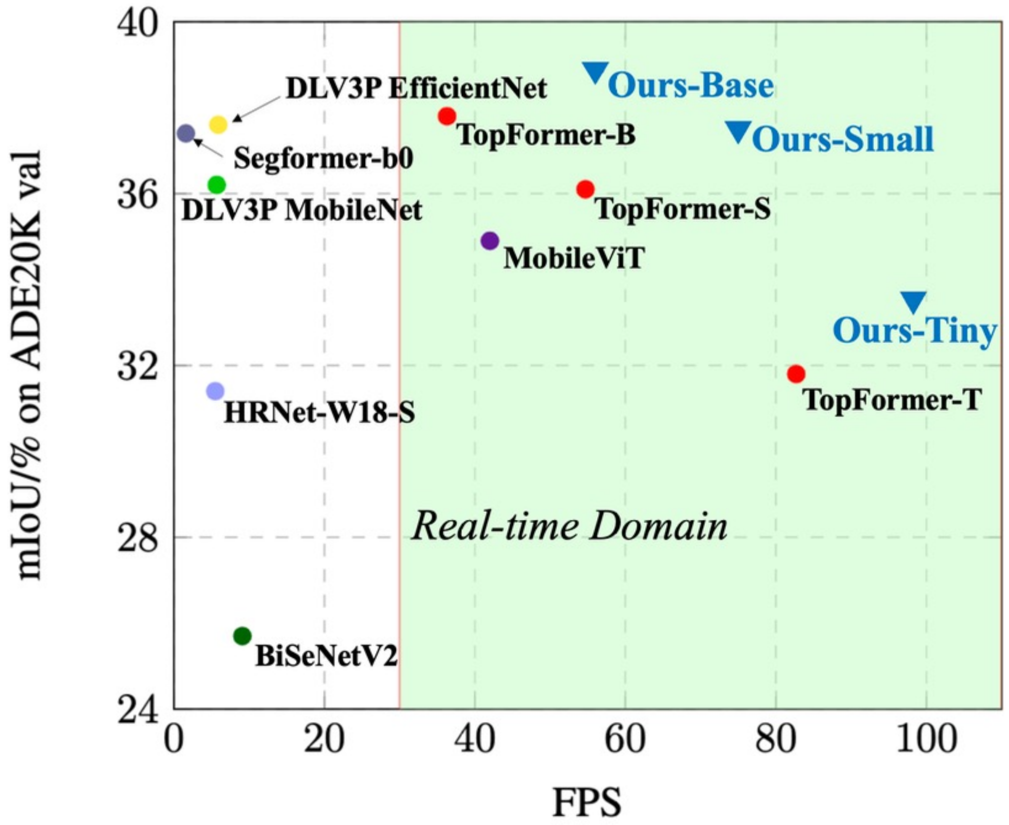


Figure 1. Comparison of accuracy versus FPS on ADE20K.

Introduction

- **Background**
  - Segmentation models usually have heavy computational cost.
  - Many light-weight models are designed for desktop GPUs.
  - NAS and pruning methods cost huge memory and training time.

- **Goal**
  - Derive ViTs with fast inference speed on edge devices.

Problem Formulation

- **Previous methods**
  - Depend on the magnitudes of model weight.
  - Adopt the in-differentiable sorting operations.
  - Leads to inconsistent performance and additional overhead.
- **Our method**
  - Uses a soft mask to indicate whether to prune.
  - Get rid of sorting operations.

Problem Formulation

- **Soft mask construction**
  - Adopt channel pruning to search for a suitable width for each convolution (CONV) layer.
  - Insert a depth-wise CONV layer following each CONV layer.
  - s_l can serve as a soft mask or pruning indicator

$$\boldsymbol{a}_l = \boldsymbol{s}_l \odot (\boldsymbol{w}_l \odot \boldsymbol{a}_{l-1})$$

Problem Formulation

- **Forward and backward propagation**
  - We adopt a threshold τ, and the forward pass with the mask is represented as:

$$\boldsymbol{b}_l = \begin{cases} 1, \boldsymbol{s}_l > \tau \\ 0, \boldsymbol{s}_l \le \tau \end{cases} \quad \text{(element-wise)}, \qquad \boldsymbol{a}_l = \boldsymbol{b}_l \odot (\boldsymbol{w}_l \odot \boldsymbol{a}_{l-1}),$$

Problem Formulation

- **Training Loss**
  - **Cross-Entropy loss and regularization loss**

$$\mathcal{L}_m(\boldsymbol{w}, \boldsymbol{s}) = \mathcal{L}(\boldsymbol{w}, \boldsymbol{s}) + \beta \cdot \mathcal{L}_{reg}(\boldsymbol{s})$$

$$\mathcal{L}_{\text{reg}} = \left| \sum_l o_l' \times i_l \times t_l \times t_l' \times k^2 - C \right|^2$$

Problem Formulation

- **Bi-Level optimization**

$$\min_{\boldsymbol{s}} \quad \mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s}),$$

$$\text{s.t.} \quad \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ \mathcal{L}(\boldsymbol{w}, \boldsymbol{s}) + \frac{1}{2\lambda}\|\boldsymbol{w}\|_2^2.$$

Proposed method

- **Bi-Level optimization method**
  - In the first step, we update the weights w with a few training steps for a fixed mask s.
  - Next in the second step, we update s with implicit gradients.

$$\frac{d\mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s})}{d\boldsymbol{s}} = \frac{d\boldsymbol{w}^*}{d\boldsymbol{s}} \nabla_{\boldsymbol{w}} \mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s}) + \nabla_{\boldsymbol{s}} \mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s})$$

$$\frac{d\mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s})}{d\boldsymbol{s}} = \nabla_{\boldsymbol{s}} \mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s}) - \lambda \boldsymbol{b} \cdot [\nabla_{\boldsymbol{w}} \mathcal{L}_m(\boldsymbol{w}^*, \boldsymbol{s})]^2$$

Problem Formulation

- **Difference with other pruning works**
  - Decouple the pruning policy from model parameter magnitudes.
  - Our method does not have such a constraint that pruned weights should be zero.
  - Our mask method is more straightforward and effective.

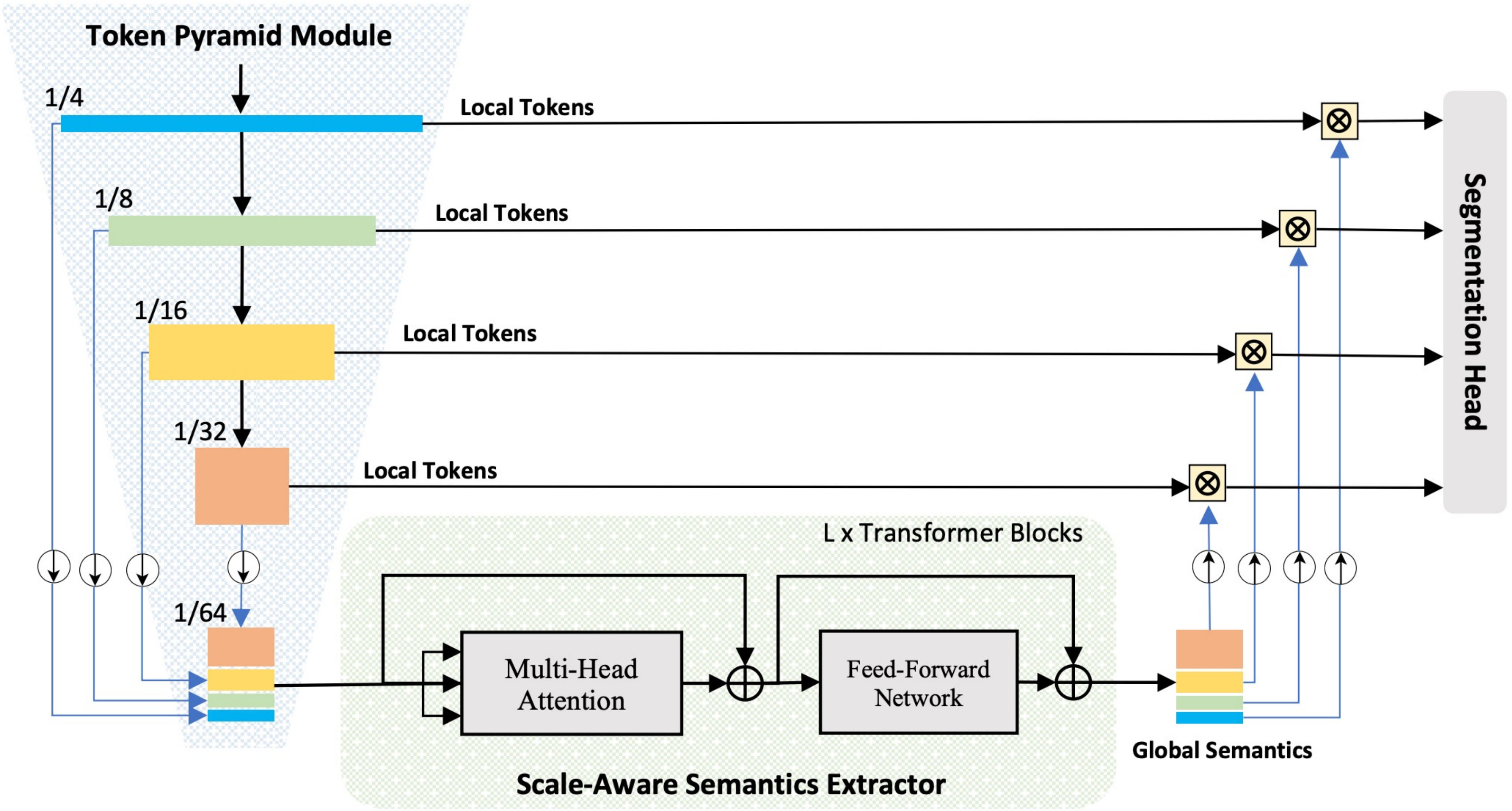
$$\boldsymbol{b}_l = \begin{cases} 1, \boldsymbol{s}_l > \tau \\ 0, \boldsymbol{s}_l \leq \tau \end{cases} \quad \text{(element-wise)}, \qquad \boldsymbol{a}_l = \boldsymbol{b}_l \odot (\boldsymbol{w}_l \odot \boldsymbol{a}_{l-1}),$$

Experiments & Results

- **Datasets**
  - **ADE20K:** 25k training, 2k validation with 150 label classes.
  - **Cityscapes:** 2,975 training, 500 validation, and 1,525 for testing with 19 label classes.
  - **Pascal VOC 2012:** 1,464 training and 1,449 images validation.

Experiments & Results

- **Tested model**
  - TopFormer



Experiments & Results

Table 1. Comparison of our searched model and prior arts on the ADE20K val dataset. We compare with popular handcraft baselines in the first segment, NAS-based models in the second segment, pruning-based methods in the third segment and lightweight ViT-based models in the fourth segment. We measure the FPS on the Qualcomm Adreno 660 GPU of the Samsung Galaxy S21 mobile phone. Some FPS results are not available due to unsupported operations on the mobile device.

Category	Method	Backbone	Parameters	GMACs	FPS	val mIoU (%)
Human Design	PSPNet [71]	ResNet50-D8	49.1M	178.8	0.4	41.1
	DeepLabV3+ [6]	EfficientNet	17.1M	26.9	5.9	37.6
	BiSeNetV2 [66]	N.A.	3.34M	12.4	9.1	25.7
	SFNet [40]	ResNet-50	–	75.7	–	42.8
	HRNet-W18-S [60]	HRNet-W18-S	4.0M	10.2	5.5	31.4
NAS	HR-NAS-A [15]	Searched	2.5M	1.4	–	33.2
	HR-NAS-B [15]	Searched	3.9M	2.2	–	34.9
	NASViT [24]	Searched	–	2.5	–	37.9
	HRViT-b1 [26]	Searched	8.2M	14.6	–	45.9
Prune	EagleEye [38]	N.A.	3.4M	1.2	59.2	34.3
	DMCP [28]	N.A.	3.3M	1.2	63.8	33.9
	ResPep [16]	N.A.	3.3M	1.2	64.9	35.0
	CHEX [32]	N.A.	3.3M	1.2	64.2	35.2
ViT	Segmenter [57]	Searched	6.7M	4.6	4.1	39.9
	MobileViT [48]	Searched	3.9M	2.2	41.8	34.9
	SegFormer-B0 [63]	MiT-B0	3.8M	8.4	2.6	37.4
	TopFormer-Base [68]	N.A.	5.1M	1.8	36.3	37.8
	TopFormer-Small [68]	N.A.	3.1M	1.2	54.7	36.1
	TopFormer-Tiny [68]	N.A.	1.4M	0.6	82.7	31.8
Ours	**Ours-Base**	Searched	**3.7M**	**1.8**	**56.5**	**38.9**
	Ours-Small	Searched	**3.3M**	**1.2**	**75.2**	**37.5**
	Ours-Tiny	Searched	**1.3M**	**0.7**	**98.0**	**33.5**

Experiments & Results

Table 2. Our search results on the Cityscapes val dataset. We compare with popular handcraft baselines, NAS-based models, pruning based methods and lightweight ViT-based models. We measure the FPS on the Qualcomm Adreno 660 GPU of the Samsung Galaxy S21 mobile phone. Some FPS results are not available due to unsupported operations on the mobile device.

Category	Method	Backbone	Resolution	#params	GMACs	FPS	mIoU%
Human Design	ENet [50]	N.A.	512×1024	$354.9K$	5.9	–	58.5
	PSPNet [71]	ResNet101	1024×2048	$68.07M$	525.0	0.2	78.8
	BiSeNetV2 [66]	N.A.	512×1024	$3.34M$	24.6	5.0	73.4
	DeepLabV3+ [6]	MBv2	512×1024	$2.26M$	9.5	9.4	69.0
	STDC1-Seg50 [21]	STDC1	512×1024	$12.05M$	31.1	4.3	72.2
	STDC2-Seg50 [21]	STDC2	512×1024	$16.08M$	44.3	3.7	74.2
NAS	Auto-DeepLab-S [45]	N.A.	1024×2048	$10.15M$	333.3	–	79.7
	FasterSeg [7]	N.A.	1024×2048	–	28.2	–	73.1
	DCNAS [69]	N.A.	1024×2048	–	294.6	–	85.0
Prune	EagleEye [38]	N.A.	512×1024	$3.6M$	2.4	27.6	69.6
	DMCP [28]	N.A.	512×1024	$3.5M$	2.4	32.0	70.3
	ResPep [16]	N.A.	512×1024	$3.5M$	2.4	28.2	71.3
	CHEX [32]	N.A.	512×1024	$3.4M$	2.4	35.5	71.7
ViT	HRViT-b1 [26]	N.A.	512×1024	$8.1M$	28.2	–	81.6
	SegFormer-B0 [63]	MiT-B0	512×1024	$3.8M$	17.7	0.9	71.9
	TopFormer-Base [68]	N.A.	512×1024	$5.1M$	2.7	21.5	70.6
	TopFormer-Tiny [68]	N.A.	512×1024	$1.4M$	1.2	42.8	66.1
Ours	**Ours-Base**	Searched	512×1024	$\mathbf{3.7M}$	**3.6**	**30.8**	**74.7**
	Ours-Small	Searched	512×1024	$\mathbf{3.3M}$	**2.4**	**38.7**	**73.6**
	Ours-Tiny	Searched	512×1024	$\mathbf{1.3M}$	**1.4**	**52.6**	**71.5**

y

Experiments & Results

Table 3. Comparison of search cost on the Cityscapes val dataset.

Method	GPU Days	GMACs	mIoU
Auto-DeepLab [45]	3	695.0	82.1
GAS [44]	6.7	-	73.5
FasterSeg [7]	2	28.2	73.1
Fast-NAS [49]	8	435.7	78.9
SparseMask [62]	4.2	36.4	68.6
DCNAS [69]	5.6	294.6	85.0
LDP [34]	4.3	—	75.8
Without implicit gradients	**1.1**	**2.4**	**71.9**
Ours-Small	**1.3**	**2.4**	**73.6**

Experiments & Results

Table 4. Results on the PASCAL VOC 2012 test dataset. We compare our results with popular CNN-based models and lightweight ViT-based models.

Method	#params	GMACs	mIoU%	FPS
EfficientNet-B7 [59]	66.0M	194.0	85.2	0.1
EMANet [41]	10.0M	43.1	80.1	2.5
PSANet [2]	18.5M	56.3	78.5	1.4
DeepLabV3+ R101 [6]	43.9M	58.5	77.4	2.2
DeepLabV3+ R50 [6]	24.9M	37.8	76.3	3.1
DeepLabV3+ MBv2 [6]	2.3M	5.7	70.5	5.1
TopFormer-B [68]	5.1M	1.8	71.0	36.8
TopFormer-S [68]	3.1M	1.2	69.8	55.2
TopFormer-T [68]	1.4M	0.6	65.7	81.5
MobileViT-XXS [48]	1.9M	1.7	73.6	43.8
Ours-Base	**3.7M**	**1.8**	**74.3**	**56.8**
Ours-Small	**3.3M**	**1.2**	**73.4**	**75.0**
Ours-Tiny	**1.3M**	**0.7**	**70.5**	**97.6**

Experiments & Results

Table 5. Our searched results for DeepLabV3+ with MobileNetV2 backbone on Cityscapes val. The input resolution is 512×1024.

Method	#params	GMACs	mIoU%	FPS
BiSeNetV2 [66]	$3.34M$	24.6	73.4	5.0
STDC1-Seg50 [21]	$12.05M$	31.1	72.2	4.3
STDC2-Seg50 [21]	$16.08M$	44.3	74.2	3.7
DeepLabV3+ [6]	$2.26M$	9.5	69.0	9.4
Ours-Base-DeepLab	**$1.21M$**	**7.6**	**70.9**	**22.3**
Ours-Small-DeepLab	**$569.0K$**	**4.3**	**70.2**	**28.1**

Experiments & Results

(a) Input (b) Ours-Base (c) TopFormer-Base

Figure A2. Visualization results on samples of Cityscapes validation dataset.

Ablation Study

Table A1. Results on Cityscapes with different β.

β	0.001	0.01	0.1	1.0
mIoU	73.1	73.6	72.8	71.2

Table A2. Results on Cityscapes with different λ.

λ	0.01	0.1	1.0
mIoU	73.1	73.6	70.8

Thanks