Rethinking BiSeNet For Real-time Semantic Segmentation

Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, Xiaolin Wei

Meituan

(CVPR 2021)

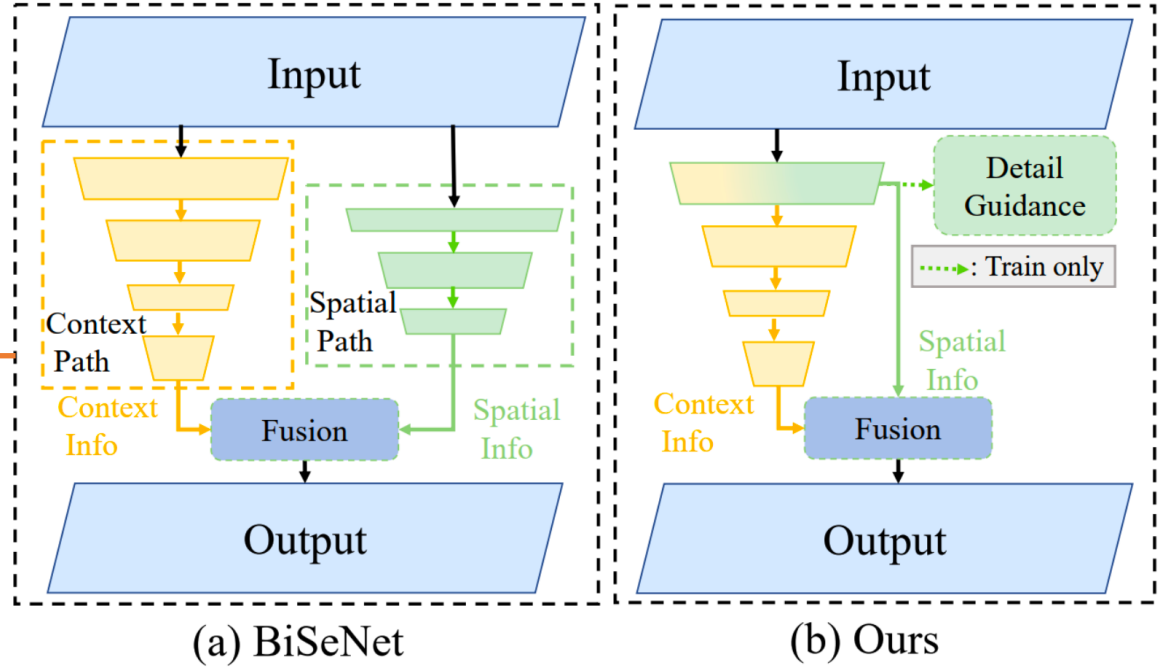
06.15.2022

전북대학교 학부생 김세희

Introduction

- Existing lightweight backbones borrowed from image classification task may not be perfect for image segmentation problem due to the deficiency of task-specific design. e.g. ResNet18
- Propose a novel hand-craft network for the purpose of faster inference speed, explainable structure, and competitive performance

Time-consuming



(a) BiSeNet (b) Ours

Method

(a)

(b)

(c)

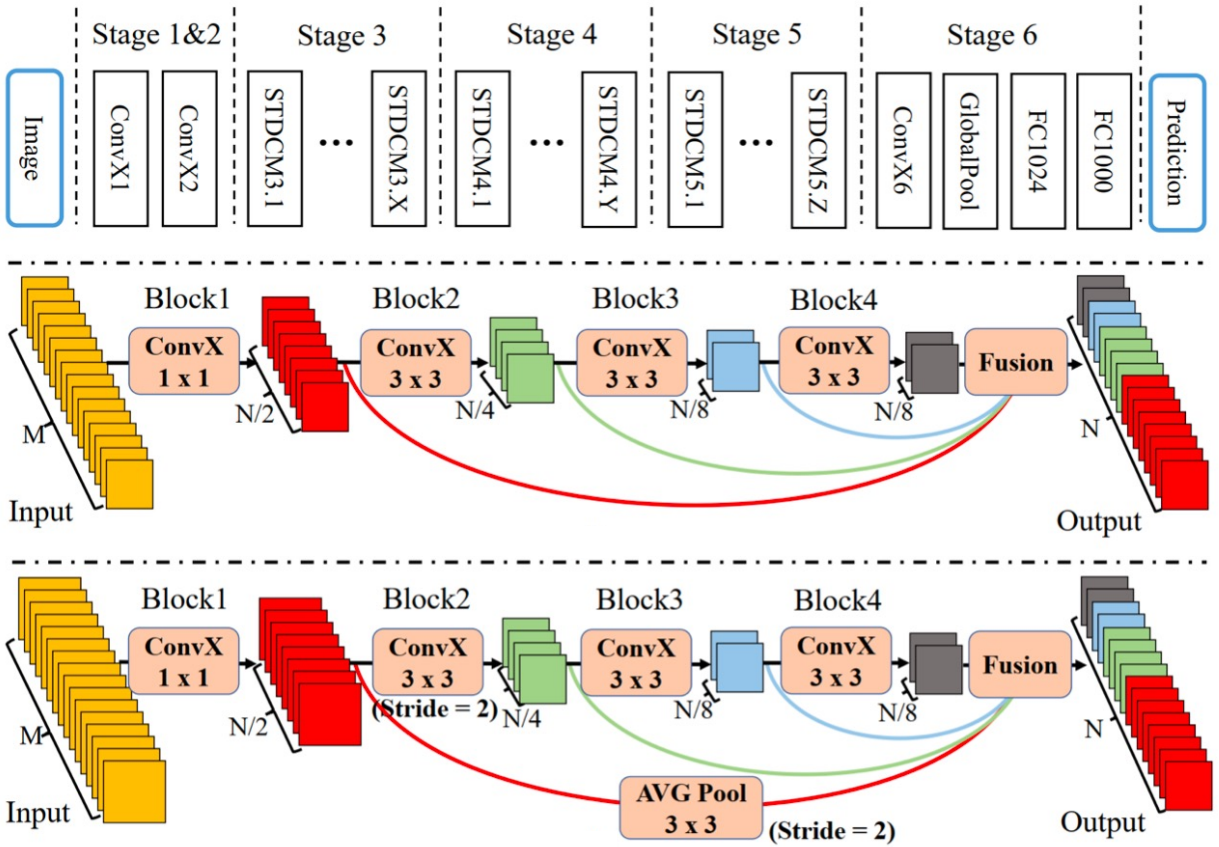


Figure 3. (a) General STDC network architecture. *ConvX* operation refers to the Conv-BN-ReLU. (b) Short-Term Dense Concatenate module (STDC module) used in our network. *M* denotes the dimension of input channels, *N* denotes the dimension of output channels. Each block is a *ConvX* operation with different kernel size. (c) STDC module with stride=2.

Method

- ConvXi denotes the operations of i-th block.
- The output of i-th block is calculated as follows:

$$x_i = ConvX_i(x_{i-1}, k_i)$$

where xi−1 and xi are the input and output of i-th block, separately.
ConvX includes one convolutional layer, one batch normalization layer and ReLU activation layer, and ki is the kernel size of convolutional layer.

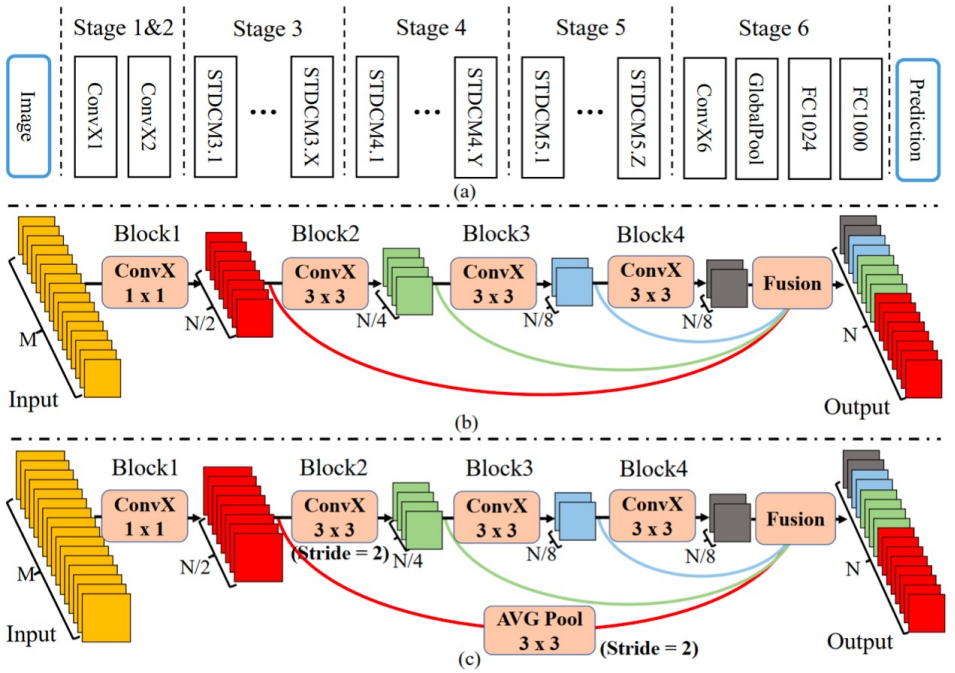


Figure 3. (a) General STDC network architecture. *ConvX* operation refers to the Conv-BN-ReLU. (b) Short-Term Dense Concatenate module (STDC module) used in our network. *M* denotes the dimension of input channels, *N* denotes the dimension of output channels. Each block is a *ConvX* operation with different kernel size. (c) STDC module with stride=2.

Method

- In STDC module, the kernel size of first block is 1, and the rest of them are simply set as 3.

- Given the channel number of STDC module's output N, the filter number of convolutional layer in i-th block is N/2i, except the filters of last convolutional layer, whose number is the same to that of previous convolutional layer.

- Down-sample is only happened in Block2.

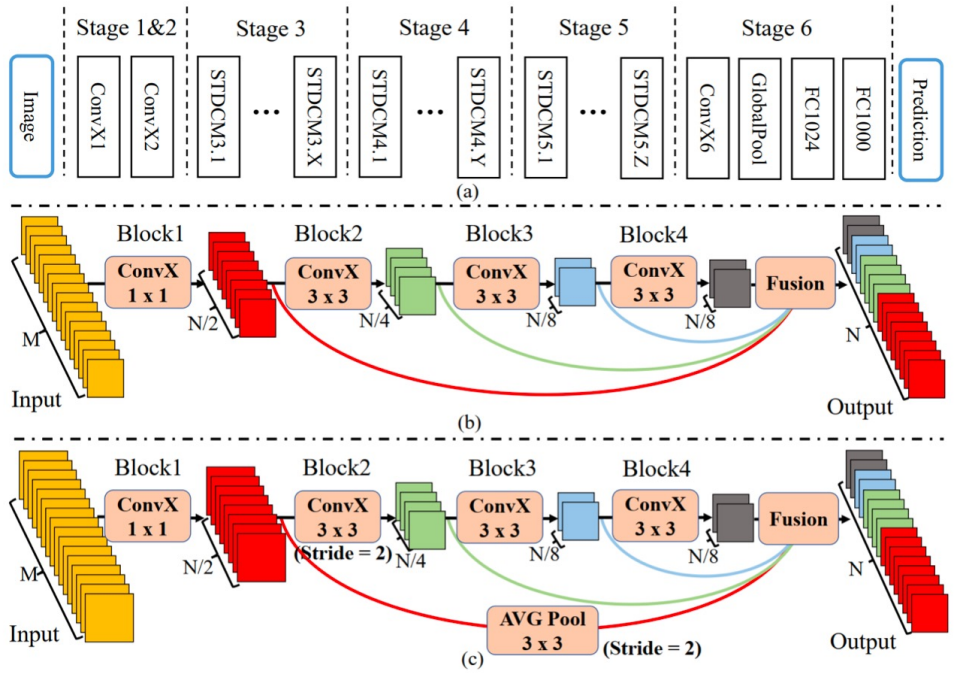


Figure 3. (a) General STDC network architecture. *ConvX* operation refers to the Conv-BN-ReLU. (b) Short-Term Dense Concatenate module (STDC module) used in our network. *M* denotes the dimension of input channels, *N* denotes the dimension of output channels. Each block is a *ConvX* operation with different kernel size. (c) STDC module with stride=2.

Method

- To enrich the feature information, we concatenate x1 to xn feature maps as the output of STDC module by skip-path.

- In our setting, the final output of STDC module is:

$$x_{output} = F(x_1, x_2, ..., x_n)$$

where xoutput denotes the STDC module output, F is the fusion operation in our method, while x1 , x2 , ..., xn are feature maps from all n blocks.

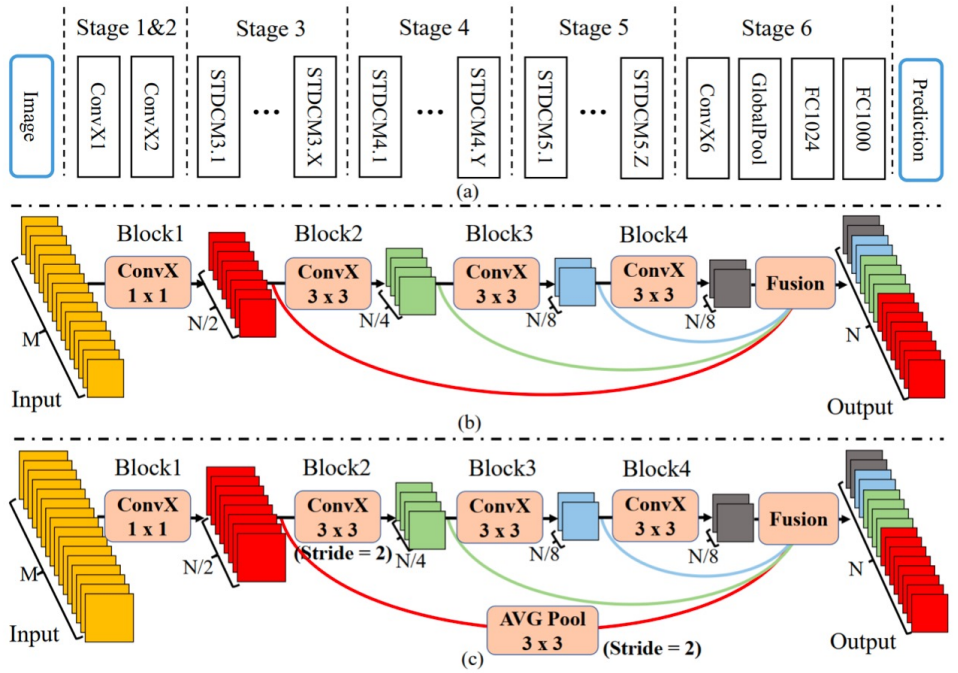


Figure 3. (a) General STDC network architecture. *ConvX* operation refers to the Conv-BN-ReLU. (b) Short-Term Dense Concatenate module (STDC module) used in our network. *M* denotes the dimension of input channels, *N* denotes the dimension of output channels. Each block is a *ConvX* operation with different kernel size. (c) STDC module with stride=2.

Method

- Low-level layers need enough channels to encode more fine-grained informations with small receptive field, while high-level layers with large receptive field focus more on high-level information induction, setting the same channel with low-level layers may cause information redundancy.

STDC module	Block1	Block2	Block3	Block4	Fusion
RF(S = 1)	1×1	3×3	5×5	7×7	$1 \times 1, 3 \times 3$ $5 \times 5, 7 \times 7$
RF(S = 2)	1×1	3×3	7×7	11×11	3×3 $7 \times 7, 11 \times 11$

Table 1. Receptive Field of blocks in our STDC module. *RF* denotes Receptive Field, *S* means stride, Note that if stride=2, the 1×1 *RF* of Block1 is turned into 3×3 *RF* by Average Pool operation.

Method

- The first STDC module in each stage down-samples the spatial resolution with a stride of 2. The following STDC modules in each stage keep the spatial resolution unchanged.

- In practice, we empirically set N6 as 1024, and carefully tune the channel number of rest stages, until reaching a good trade-off between accuracy and efficiency.

Stages	Output size	KSize	S	STDC1		STDC2	
				R	C	R	C
Image	224×224				3		3
ConvX1	112×112	3×3	2	1	32	1	32
ConvX2	56×56	3×3	2	1	64	1	64
Stage3	28×28 28×28		2 1	1 1	256	1 3	256
Stage4	14×14 14×14		2 1	1 1	512	1 4	512
Stage5	7×7 7×7		2 1	1 1	1024	1 2	1024
ConvX6	7×7	1×1	1	1	1024	1	1024
GlobalPool	1×1	7×7					
FC1					1024		1024
FC2					1000		1000
FLOPs				813M		1446M	
Params				8.44M		12.47M	

Table 2. Detailed architecture of STDC networks. Note that *ConvX* shown in the table refers to the Conv-BN-ReLU. The basic module of Stage 3, 4 and 5 is STDC module. KSize mean kernel size. S, R, C denote stride, repeat times and output channels respectively.

Method

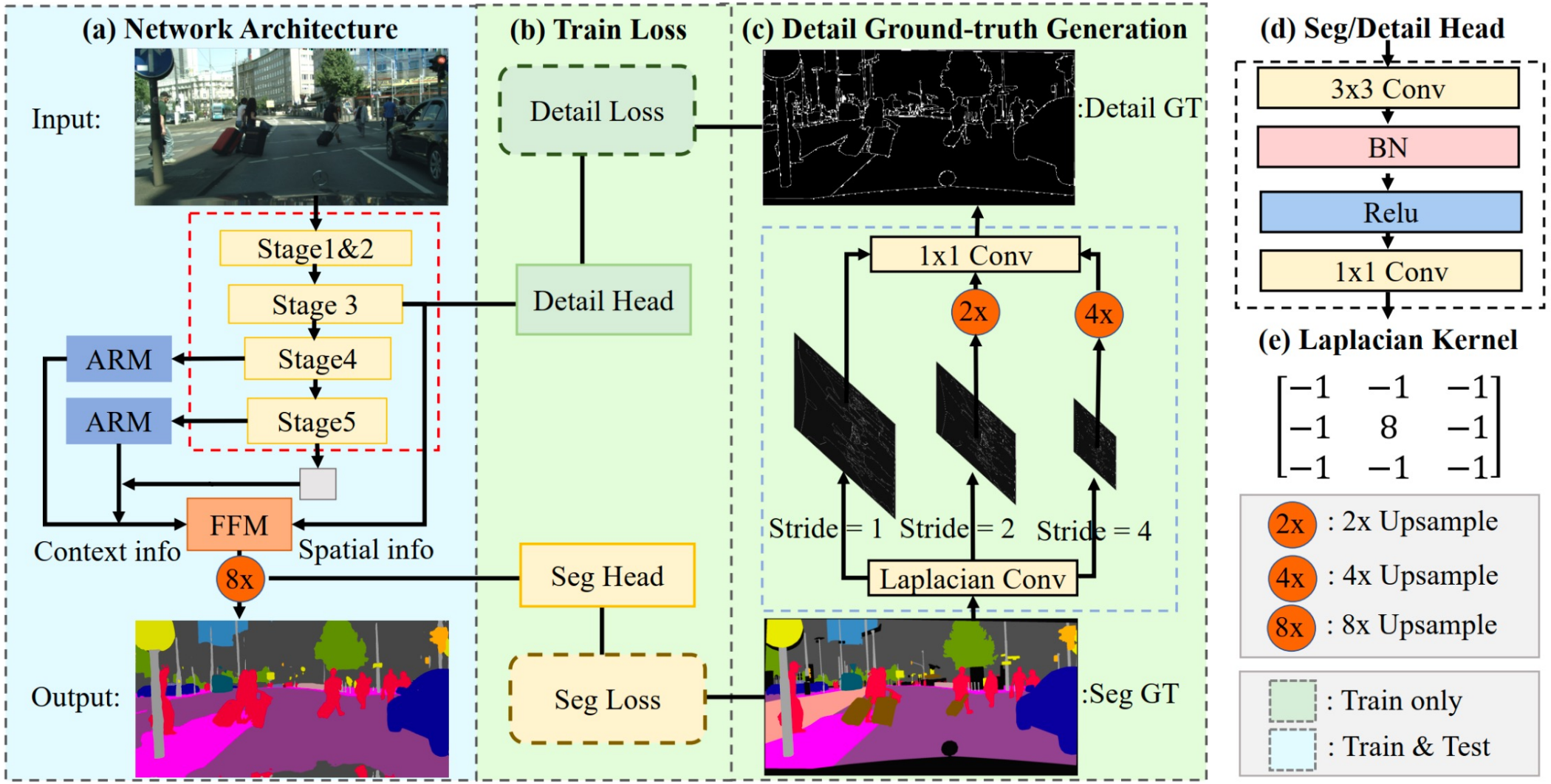
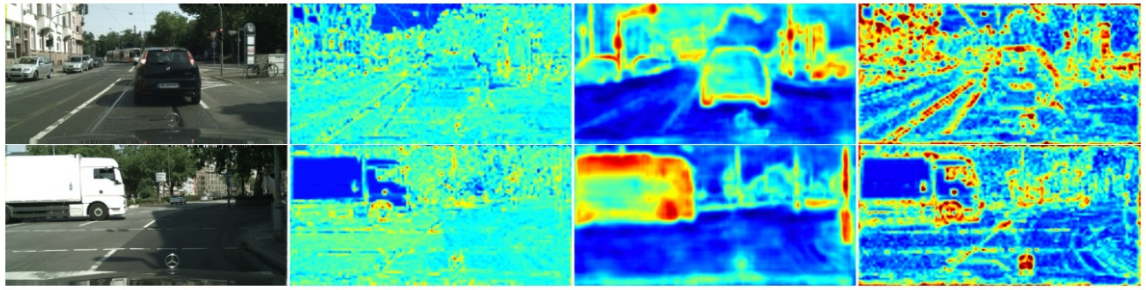


Figure 4. Overview of the STDC Segmentation network. *ARM* denotes *Attention Refine module*, and *FFM* denotes *Feature Fusion Module* in [28]. The operation in the dashed red box is our STDC network. The operation in the dashed blue box is *Detail Aggregation Module*.

Method

- We visualize the features of BiSeNet's spatial path in Figure 5(b).

- We propose a Detail Guidance module to guide the low-level layers to learn the spatial information in single-stream manner.



(a) Input (b) Spatial Path (c) Stage3 (d) Stage3$_D$

Figure 5. Visual explanations for features in the spatial path and Stage 3 without and with *Detail Guidance*. The column with subscript **D** denotes results with *Detail Guidance*. The visualization shows that spatial path can encode more spatial detail,*e.g.*, boundary, corners, than backbone's low-level layers, while our *Detail Guidance* module can do the same thing without extra computation cost.

Method

- Since the number of detail pixels is much less than the non-detail pixels, detail prediction is a class- imbalance problem.

- We adopt binary cross-entropy and dice loss to jointly optimize the detail learning.

- For the predicted detail map with the height H and the width W , the detail loss L_{detail} is formulated as follows:

$$L_{detail}(p_d, g_d) = L_{dice}(p_d, g_d) + L_{bce}(p_d, g_d)$$

where $p_d \in R^{H \times W}$ denotes the predicted detail and $g_d \in R^{H \times W}$ denotes the corresponding detail ground-truth. L_{bce} denotes the binary cross-entropy loss while L_{dice} denotes the dice loss.

Method

- L_{dice} is formulated as follows:

$$L_{dice}(p_d, g_d) = 1 - \frac{2\sum_i^{H\times W} p_d^i g_d^i + \epsilon}{\sum_i^{H\times W} (p_d^i)^2 + \sum_i^{H\times W} (g_d^i)^2 + \epsilon}$$

where i denotes the i-th pixel and ε is a Laplace smoothing item to avoid zero division. In this paper we set ε = 1.

Experiment

- We implement our method on three datasets: ImageNet, Cityscapes and CamVid to evaluate the effectiveness of our proposed backbone and segmentation network, respectively.

- For classification evaluation, we use evaluate top-1 accuracy as the evaluation metrics following.

- For segmentation evaluation, we adopt mean of class-wise intersection over union (mIoU) and Frames Per Second (FPS) as the evaluation metrics.

Experiment

- In this paper, we set the block number in STDC1 and STDC2 to 4

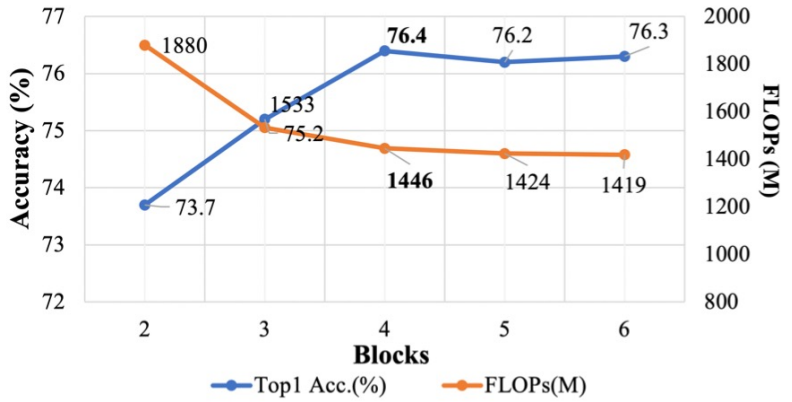


Figure 7. Comparisons with different block number of STDC2 on ImageNet.

- Our STDC2 yield the best speed-accuracy trade-off compared with other lightweight backbones.

Backbone	Resolution	mIoU(%)	FPS
GhostNet [8]	512 × 1024	67.8	135.0
MobileNetV3 [12]	512 × 1024	70.1	148.3
EfficientNet-B0 [26]	512 × 1024	72.2	99.9
STDC2	512 × 1024	**74.2**	**188.6**
GhostNet [8]	768 × 1536	71.3	60.9
MobileNetV3 [12]	768 × 1536	73.0	70.4
EfficientNet-B0 [26]	768 × 1536	73.9	45.9
STDC2	768 × 1536	**77.0**	**97.0**

Table 3. Lightweight backbone comparison on Cityscapes *val* set. All experients utilize the same decoder and same experiment settings.

Experiment

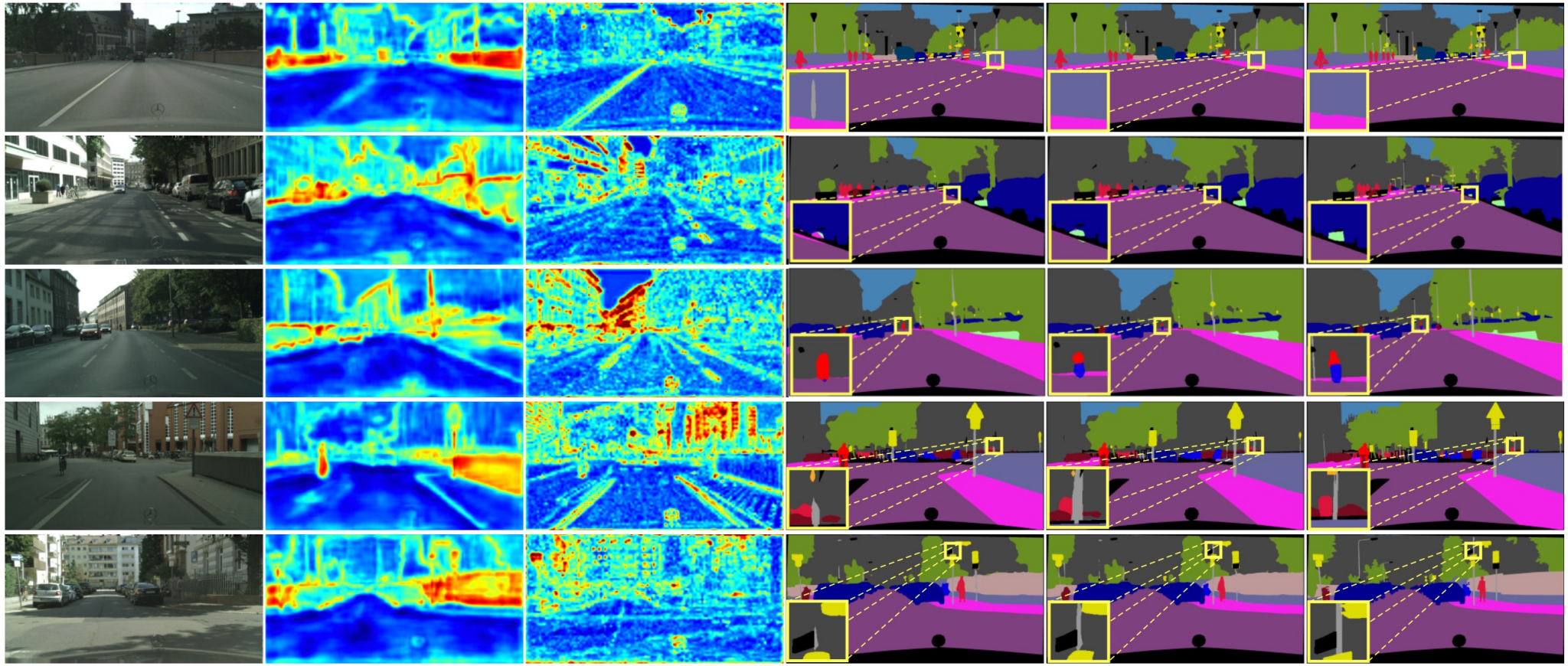


Figure 6. Visual comparison of our *Detail Guidance* on Cityscapes *val* set. The column with subscript **D** denotes results with *Detail Guidance*. The first row (a) shows the input images. (b) and (c) illustrate the heatmap of Stage 3 without and with *Detail Guidance*. (d) and (e) demonstrate the predictions without and with *Detail Guidance*. (f) is the ground-truth of input images.

Results

Model	Top1 Acc.	Params	FLOPs	FPS
ResNet-18 [9]	69.0%	11.2M	1800M	1058.7
ResNet-50 [9]	75.3%	23.5M	3800M	378.7
DF1 [21]	69.8%	8.0M	746M	1281.3
DF2 [21]	73.9%	17.5M	1770M	713.2
DenseNet121 [15]	75.0%	9.9M	2882M	363.6
DenseNet161 [15]	76.2%	28.6M	7818M	255.0
GhostNet(x1.0) [8]	73.9%	5.2M	141M	699.1
GhostNet(x1.3) [8]	75.7%	7.3M	226M	566.2
MobileNetV2 [25]	72.0%	3.4M	300M	998.8
MobileNetV3 [12]	75.2%	5.4M	219M	661.2
EfficientNet-B0 [26]	76.3%	5.3M	390M	443.0
STDC1	73.9%	8.4M	813M	**1289.0**
STDC2	**76.4%**	12.5M	1446M	813.6

Table 5. Comparisons with other popular networks on ImageNet Classification.

Results

Model	Resolution	Backbone	mIoU(%)		FPS
			val	test	
ENet [24]	512 × 1024	no	-	58.3	76.9
ICNet [31]	1024 × 2048	PSPNet50	-	69.5	30.3
DABNet [17]	1024 × 2048	no	-	70.1	27.7
DFANet B [18]	1024 × 1024	Xception B	-	67.1	120
DFANet A' [18]	512 × 1024	Xception A	-	70.3	160
DFANet A [18]	1024 × 1024	Xception A	-	71.3	100
BiSeNetV1 [28]	768 × 1536	Xception39	69.0	68.4	105.8
BiSeNetV1 [28]	768 × 1536	ResNet18	74.8	74.7	65.5
CAS [30]	768 × 1536	no	-	70.5	108.0
GAS [22]	769 × 1537	no	-	71.8	108.4
DF1-Seg-d8 [21]	1024 × 2048	DF1	72.4	71.4	136.9
DF1-Seg[21]	1024 × 2048	DF1	74.1	73.0	106.4
DF2-Seg1[21]	1024 × 2048	DF2	75.9	74.8	67.2
DF2-Seg2[21]	1024 × 2048	DF2	76.9	75.3	56.3
SFNet [20]	1024 × 2048	DF1	-	74.5	121
HMSeg [19]	768 × 1536	no	-	74.3	83.2
TinyHMSeg [19]	768 × 1536	no	-	71.4	172.4
BiSeNetV2 [27]	512 × 1024	no	73.4	72.6	156
BiSeNetV2-L [27]	512 × 1024	no	75.8	75.3	47.3
FasterSeg [4]	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

Table 6. Comparisons with other state-of-the-art methods on Cityscapes. *no* indicates the method do not have a backbone.

Model	Resolution	Backbone	mIoU(%)	FPS
ENet [24]	720 × 960	no	51.3	61.2
ICNet [31]	720 × 960	PSPNet50	67.1	34.5
BiSeNetV1 [28]	720 × 960	Xception39	65.6	175
BiSeNetV1 [28]	720 × 960	ResNet18	68.7	116.3
CAS [30]	720 × 960	no	71.2	169
GAS [22]	720 × 960	no	72.8	153.1
BiSeNetV2 [27]	720 × 960	no	72.4	124.5
BiSeNetV2-L [27]	720 × 960	no	73.2	32.7
STDC1-Seg	720 × 960	STDC1	73.0	**197.6**
STDC2-Seg	720 × 960	STDC2	**73.9**	152.2

Table 7. Comparisons with other state-of-the-art methods on CamVid. *no* indicates the method do not have a backbone.

Thank you