CVPR2020

Joint Semantic Segmentation and Boundary Detection using Iterative Pyramid Contexts

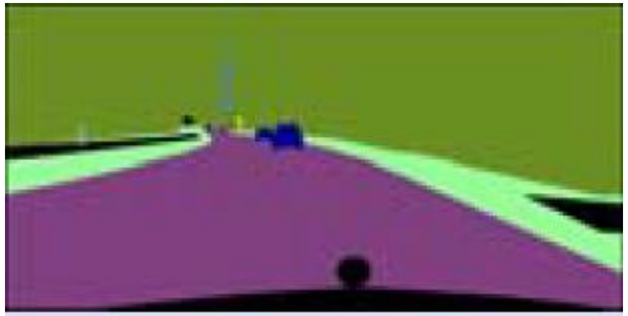
Mingmin Zhen[1] Jinglu Wang[2] Lei Zhou[1] Shiwei Li[3]
Tianwei Shen[1] Jiaxiang Shang[1] Tian Fang[3] Long Quan[1]
[1]Hong Kong University of Science and Technology [2]Microsoft Research Asia [3]Everest Innovation Technology
{mzhen, lzhouai, tshenaa, jshang, quan}@cse.ust.hk
Jinglu.Wang@microsoft.com {sli, fangtian}@altizure.com

Motivation

- As a dual problem of semantic segmentation, which means that the boundary always surrounds the mask, the goal of semantic boundary detection is to identify image pixels that belong to object(class) boundaries.

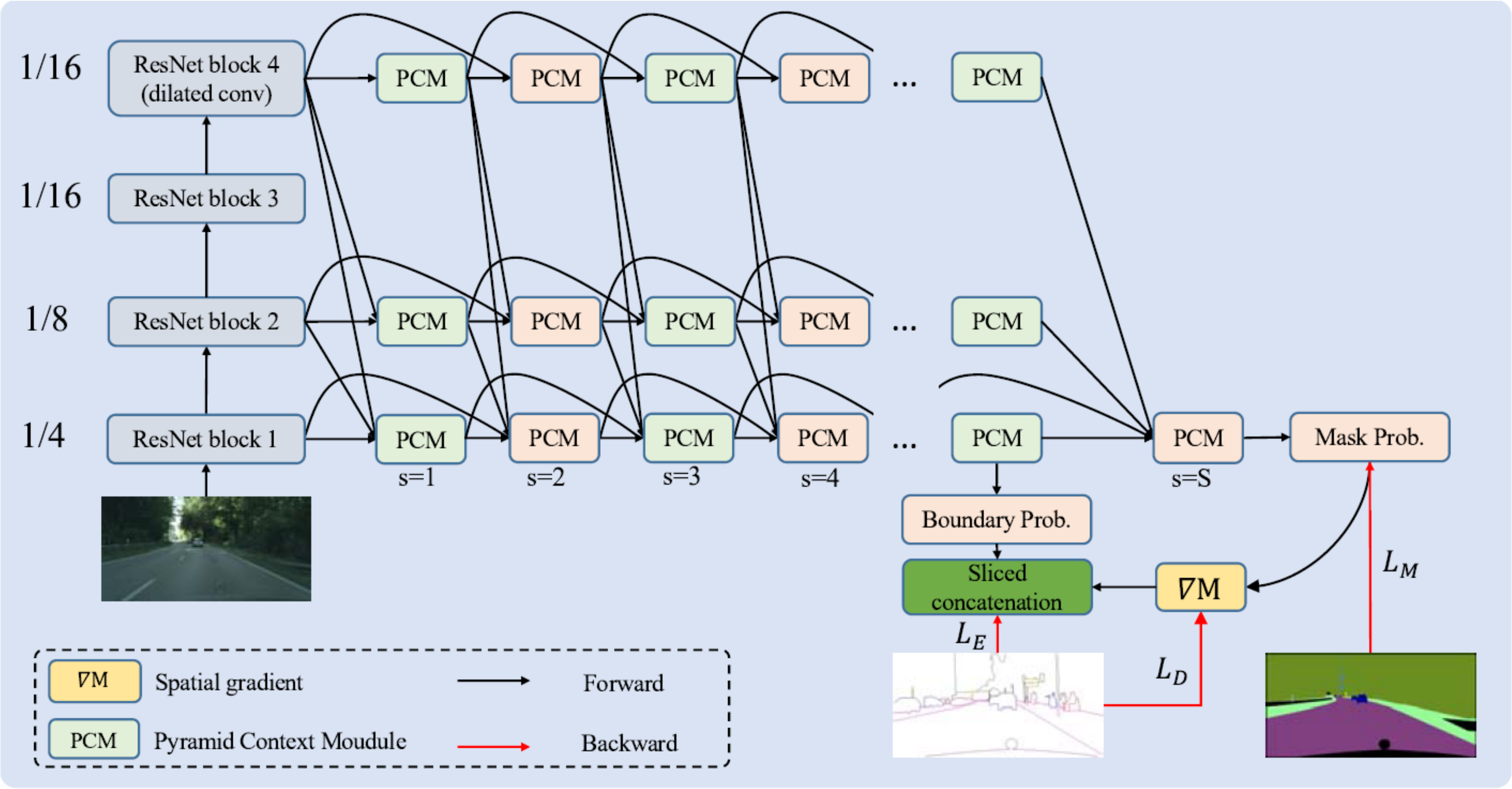
- In general, estimating the semantic label at image boundaries is challenging as it could be ambiguous between two sides.

- For semantic boundary detection, one challenging issue is to suppress the non-semantic edges, which are ambiguous to distinguish from semantic edges.

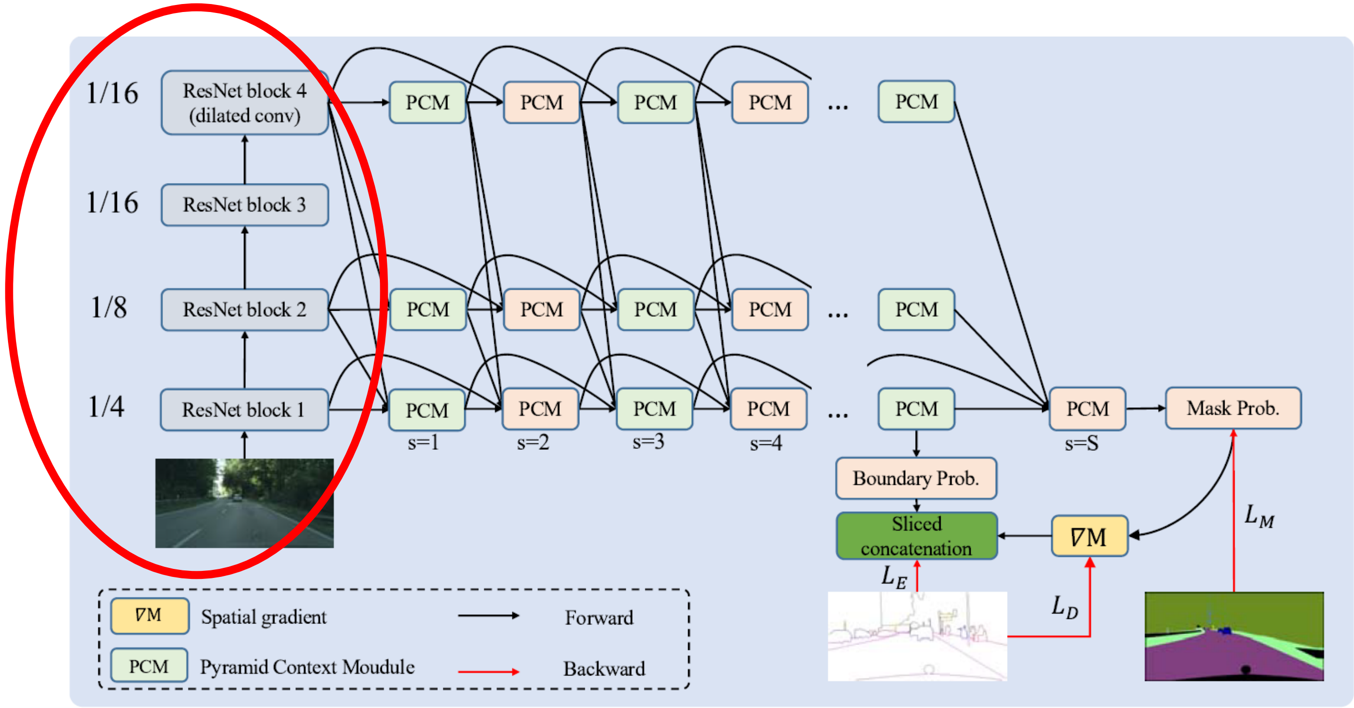

Method

PCM: capture the pyramid context from *multiscale* feature maps.



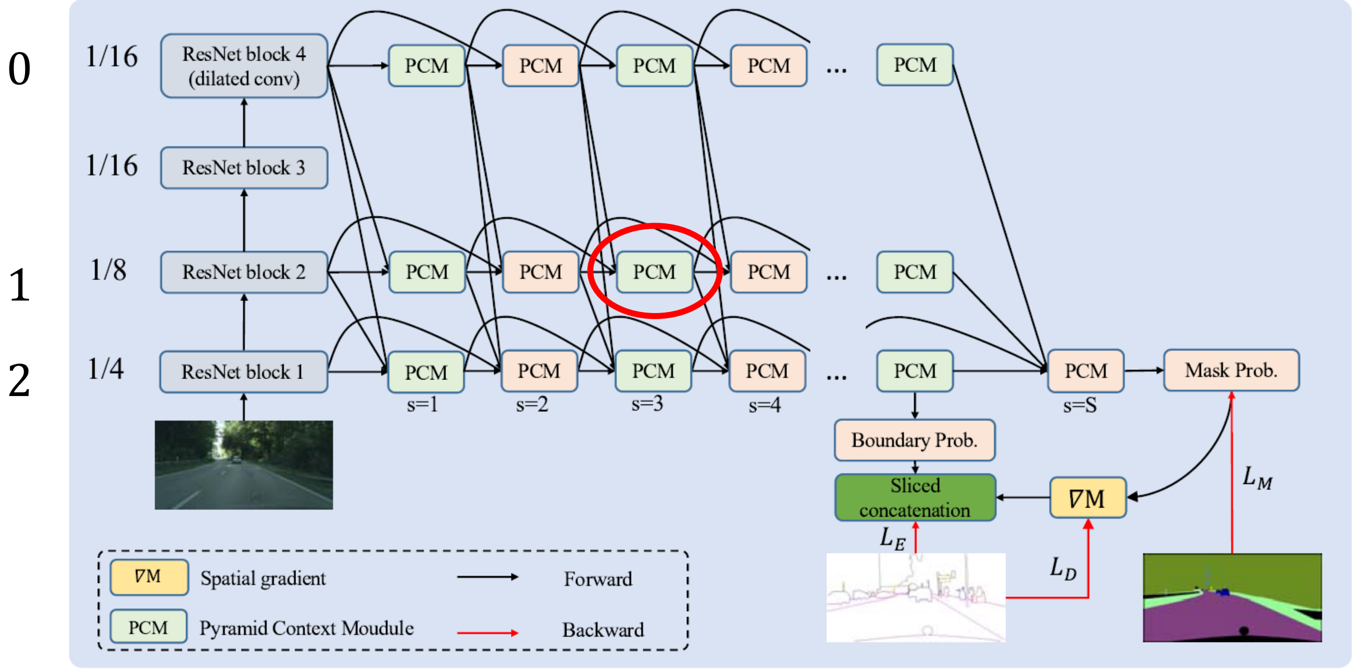
Method

Backbone: ResNet with dilated strategy.



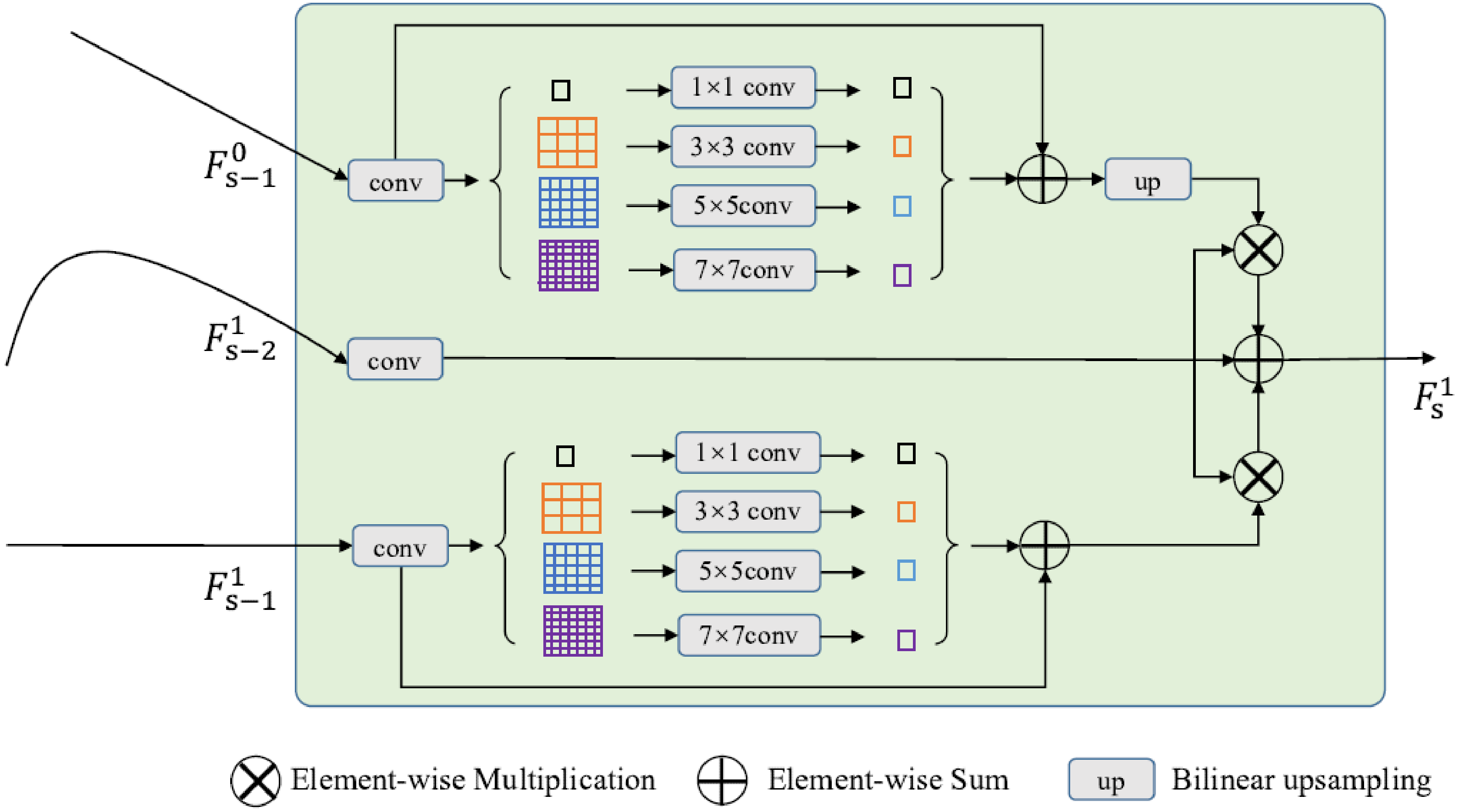
Method

PCM: capture the pyramid context from *multiscale* feature maps.



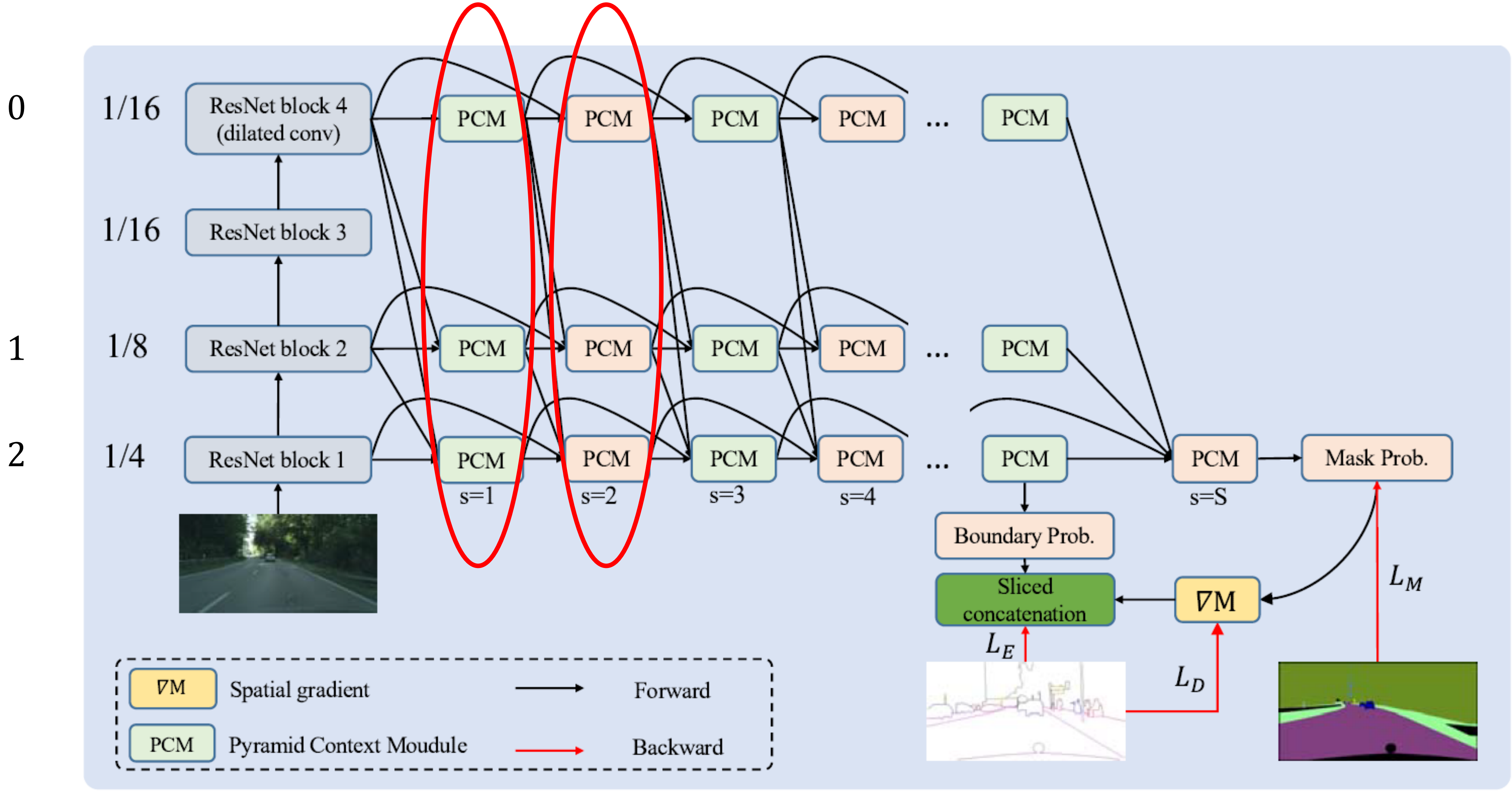
Method

$$P^{t'}_{G\times G}(x,y)=\frac{1}{|S(x,y)|}\sum_{(h,w)\in S(x,y)}F(h,w)$$



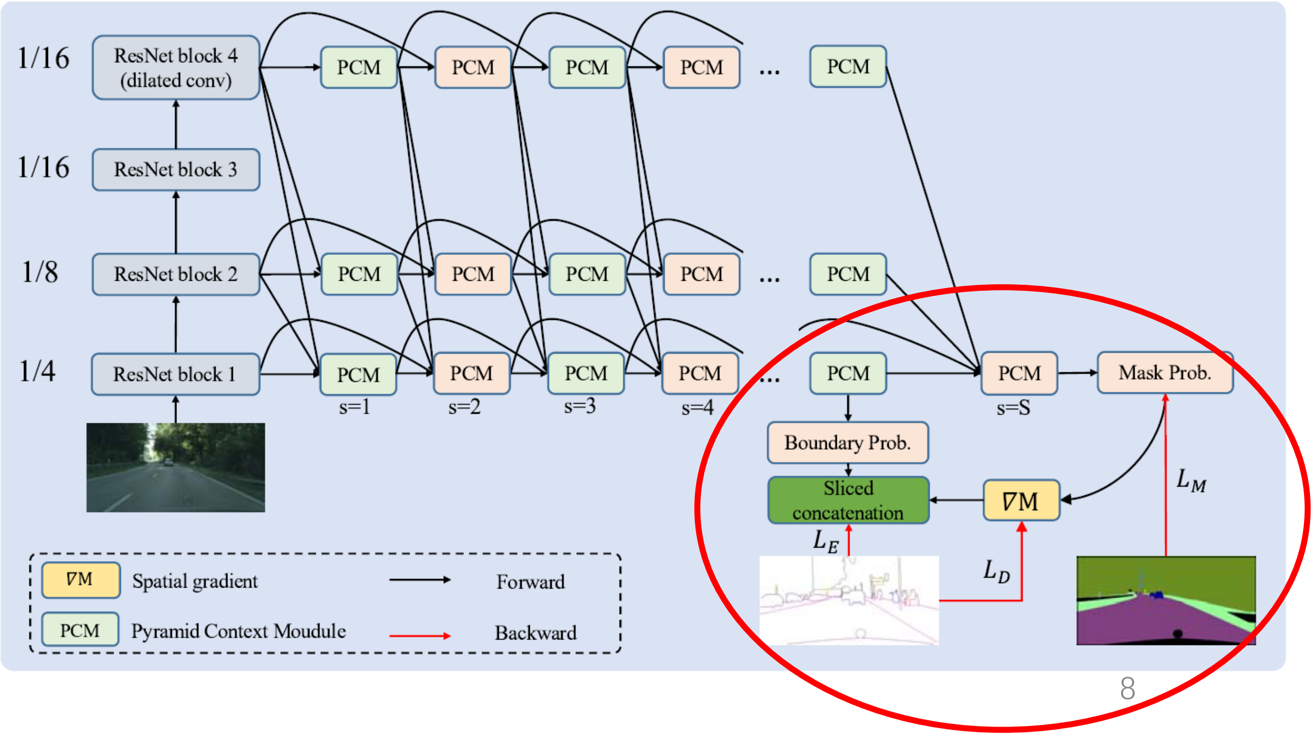
Method

Iterative???



Method

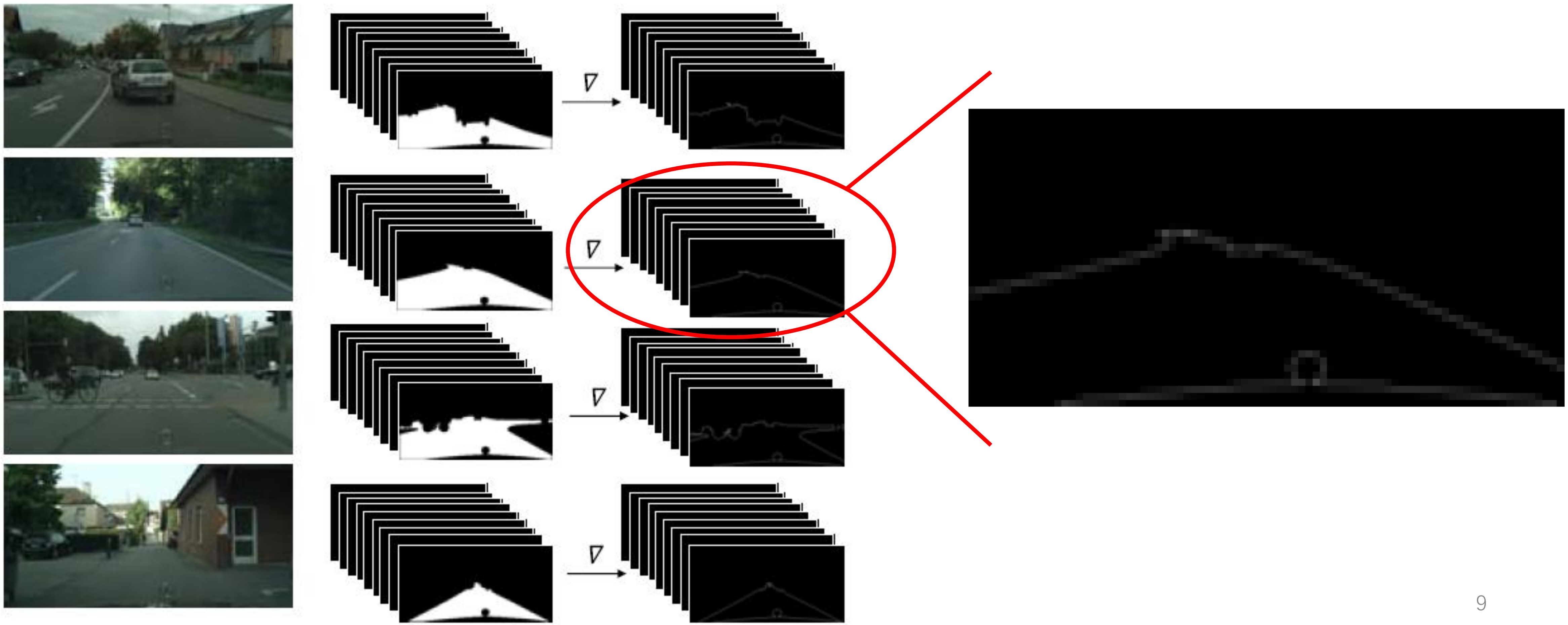
Loss Function: $L_{total} = L_M + \lambda_1 L_D + \lambda_2 L_E$



Method

Spatial Gradient ∇M:

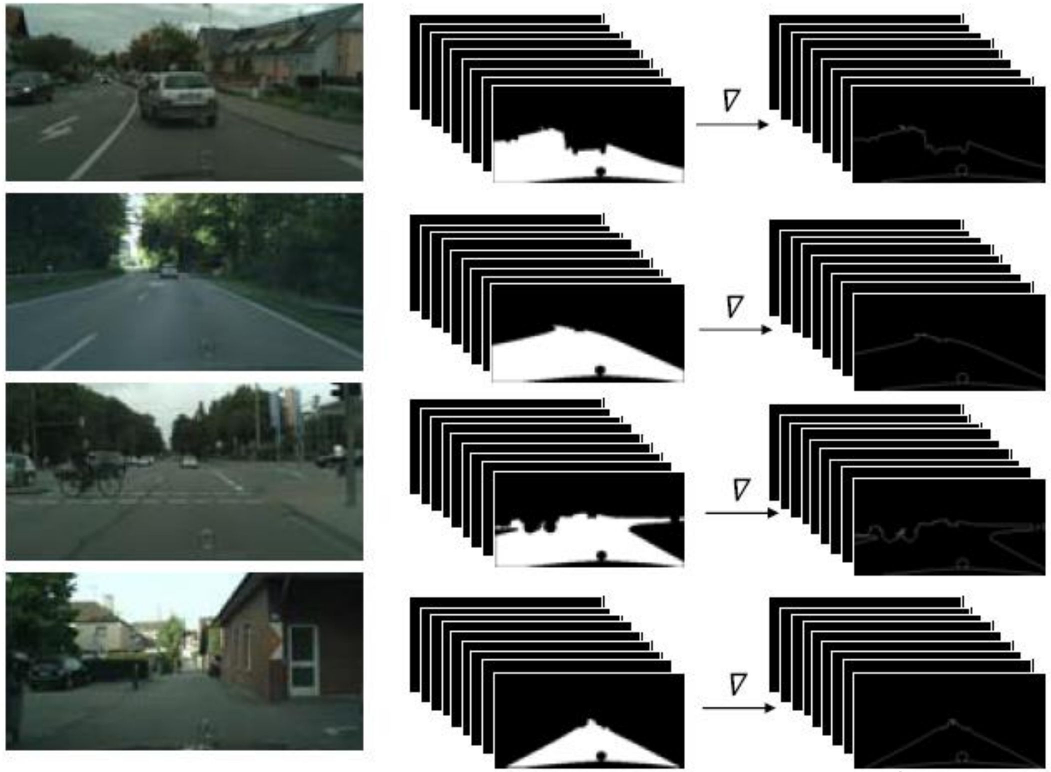
$$\nabla M = \left|M(x,y) - pool_k\big(M(x,y)\big)\right|$$



Method

$$L_D = \sum_{\mathrm{i}} \left| \nabla M_i - B_i^{gt} \right|$$

B_i^{gt} is the semantic boundary ground truth derived from semantic segmentation mask ground truth.



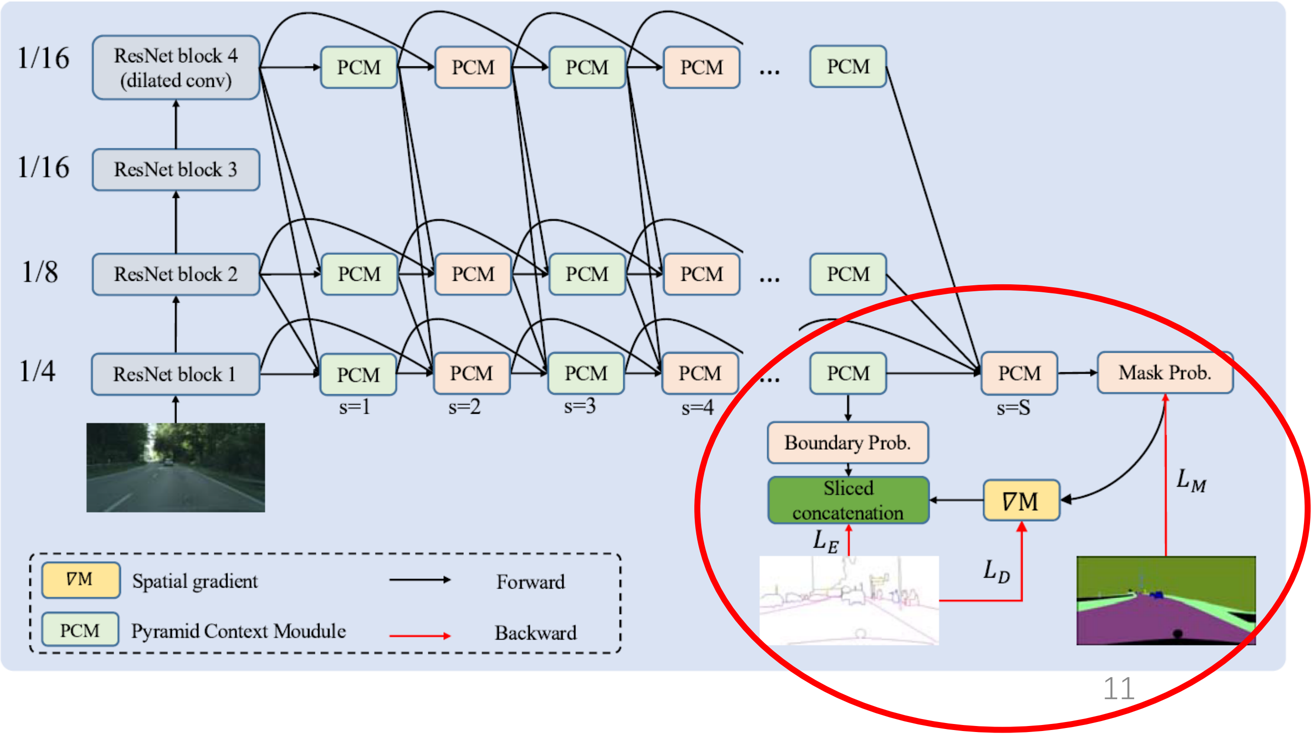
Method

∇M Fusion:

$$B = \{B_1, B_2, \dots, B_k\}$$
$$\nabla M = \{\nabla M_1, \nabla M_2, \dots, \nabla M_k\}$$

Sliced concatenation:

$$[B_1, \nabla M_1, B_2, \nabla M_2, \dots, B_k, \nabla M_k]$$



Experiments

- Cityscapes dataset contains **2975** training, **500** validation and **1525** test images. Each images has a high resolution of **2048**$\times$**1024** pixels with **19** semantic classes.

Experiments

Duality loss	∇M	PCM	mIoU	MF (ODS) / AP
-	-	-	78.14	73.61 / 72.81
✓	-	-	79.58	74.24 / 73.57
✓	✓	-	79.81	74.45 / 74.20
✓	✓	$\{1\}$	79.92	74.65 / 74.29
✓	✓	$\{1, 3\}$	80.20	74.80 / 74.54
✓	✓	$\{1, 3, 5, 7\}$	80.43	75.54 / 75.14

Table 1. Ablation experiments for duality loss, ∇M fusion and pyramid context module (PCM). We set S to 8 in the experiments.

Experiments

S	mIoU	MF (ODS) / AP
1	77.65	-
1	-	72.62 / 71.56
2	78.77	73.44 / 72.53
3	79.44	74.55 / 73.78
4	79.80	74.56 / 73.80
5	79.88	74.61 / 73.91
6	80.25	74.94 / 74.31
7	80.36	75.10 / 74.38
8	**80.43**	**75.54 / 75.14**

Table 2. Ablation experiments of iterative pyramid context module for semantic segmentation and semantic boundary. The iterative steps S is set from 1 to 8. For $S = 1$, only one task is trained and evaluated.

Method	Backbone	mIoU
DeeplabV2 [27]	ResNet101	70.4
Piecewise [33]	ResNet101	71.6
PSPNet [23]	ResNet101	78.8
DeeplabV3+ [19]	ResNet101	78.8
InPlaceABN [30]	WideResNet38	79.4
GSCNN [5]	ResNet101	80.8
DANet [32]	ResNet101	81.5
RPCNet (SS + Flip)	ResNet101	81.8
RPCNet (MS + Flip)	ResNet101	**82.1**

Table 3. Performance comparison between different strategies on Cityscape val set. "SS": single scale test. "MS": multi-scale test.

Experiments

Method	Backbone data	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
DeeplabV2 [27]	ResNet101	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
RefineNet [20]	ResNet101	98.2	83.3	91.3	47.8	50.4	56.1	66.9	71.3	92.3	70.3	94.8	80.9	63.3	94.5	64.6	76.1	64.3	62.2	70.0	73.6
PSPNet [23]	ResNet101	98.6	86.2	92.9	50.8	58.8	64.0	75.6	79.0	93.4	72.3	95.4	86.5	71.3	95.9	68.2	79.5	73.8	69.5	77.2	78.4
AAF [34]	ResNet101	98.5	85.6	93.0	53.8	58.9	65.9	75.0	78.4	93.7	72.4	95.6	86.4	70.5	95.9	73.9	82.7	76.9	68.7	76.4	79.1
DenseASPP [35]	DenseNet161	98.7	87.1	93.4	60.7	62.7	65.6	74.6	78.5	93.6	72.5	95.4	86.2	71.9	96.0	78.0	90.3	80.7	69.7	76.8	80.6
PSANet [24]	ResNet101	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	80.1
SeENet [36]	ResNet101	98.7	87.3	93.7	57.1	61.8	70.5	77.6	80.9	94.0	73.5	95.9	87.5	71.6	96.3	76.4	88.0	79.9	73.0	78.5	81.2
ANNNet [37]	ResNet101	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	81.3
CCNet [28]	ResNet101	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	81.4
BFP [6]	ResNet101	98.7	87.0	93.5	59.8	63.4	68.9	76.8	80.9	93.7	72.8	95.5	87.0	72.1	96.0	77.6	89.0	86.9	69.2	77.6	81.4
DANet [32]	ResNet101	98.6	87.1	93.5	56.1	63.3	69.7	77.3	81.3	93.9	72.9	95.7	87.3	72.9	96.2	76.8	89.4	86.5	72.2	78.2	81.5
Ours	ResNet101	98.7	86.7	93.9	62.4	62.8	70.5	77.5	81.1	94.0	72.3	95.9	87.8	74.1	96.3	76.5	88.0	85.2	71.0	78.6	**81.8**

Table 4. Comparison vs state-of-the-art methods without coarse data training on the Cityscapes test set.

Metric	Method	Test NMS	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
MF (ODS)	CASENet [3]		87.06	75.95	75.74	46.87	47.74	73.23	72.70	75.65	80.42	57.77	86.69	81.02	67.93	89.10	45.92	68.05	49.63	54.21	73.74	68.92
	CASENet* [4]		87.23	76.08	75.73	47.86	47.57	73.67	71.77	75.19	80.58	58.39	86.78	81.00	68.18	89.31	48.99	67.82	50.84	55.30	74.16	69.29
	CASENet* [4]	✓	88.13	76.53	76.75	48.70	48.60	74.21	74.54	76.38	81.32	58.98	87.26	81.90	69.05	90.27	50.93	68.41	52.11	56.23	75.66	70.31
	STEAL [4]		88.08	77.62	77.08	50.02	49.62	75.48	74.01	76.66	81.51	59.41	87.24	81.90	69.87	89.50	52.15	67.80	53.60	55.93	75.17	70.67
	STEAL [4]	✓	88.94	78.21	77.75	50.59	50.39	75.54	76.31	77.45	82.28	60.19	87.99	82.48	70.18	90.40	53.31	68.50	53.39	56.99	76.14	71.42
	Ours	✓	**90.86**	**82.32**	**82.11**	**57.15**	**58.97**	**84.48**	**83.34**	**82.26**	**84.88**	**64.22**	**89.87**	**86.28**	**78.47**	**92.61**	**67.75**	**82.79**	**68.48**	**69.20**	**80.09**	**78.22**
AP	CASENet [3]		54.58	65.44	67.75	37.97	39.93	57.28	64.65	69.38	71.27	50.28	73.99	72.56	59.92	66.84	35.91	56.04	41.19	46.88	63.54	57.65
	CASENet* [4]		68.38	69.61	70.28	40.00	39.26	61.74	62.74	73.02	72.77	50.91	80.72	76.06	60.49	79.43	40.86	62.27	42.87	48.84	64.42	61.30
	CASENet* [4]	✓	88.83	73.94	76.86	42.06	41.75	69.81	74.50	76.98	79.67	56.48	87.73	83.21	68.10	91.20	44.17	66.69	44.77	52.04	75.65	68.13
	STEAL [4]		89.54	75.72	74.95	42.72	41.53	65.86	67.55	75.84	77.85	52.72	82.70	79.89	62.59	91.07	45.26	67.73	47.08	50.91	70.78	66.44
	STEAL [4]	✓	90.86	78.94	77.36	43.01	42.33	71.13	75.57	77.60	81.60	56.98	87.30	83.21	66.79	91.59	45.33	66.64	46.25	52.07	74.41	68.89
	Ours	✓	**91.27**	**83.87**	**84.00**	**53.18**	**54.96**	**84.55**	**85.48**	**84.66**	**86.15**	**61.18**	**90.72**	**88.95**	**79.95**	**94.40**	**68.11**	**85.47**	**68.53**	**69.44**	**82.17**	**78.79**

Table 5. Quantitative results on the val set on the Cityscapes dataset. We use ResNet101 pretrained on ImageNet as backbone. CASENet* is the reimplementation of CASENet in [4]. Scores are measured by %.

Experiments

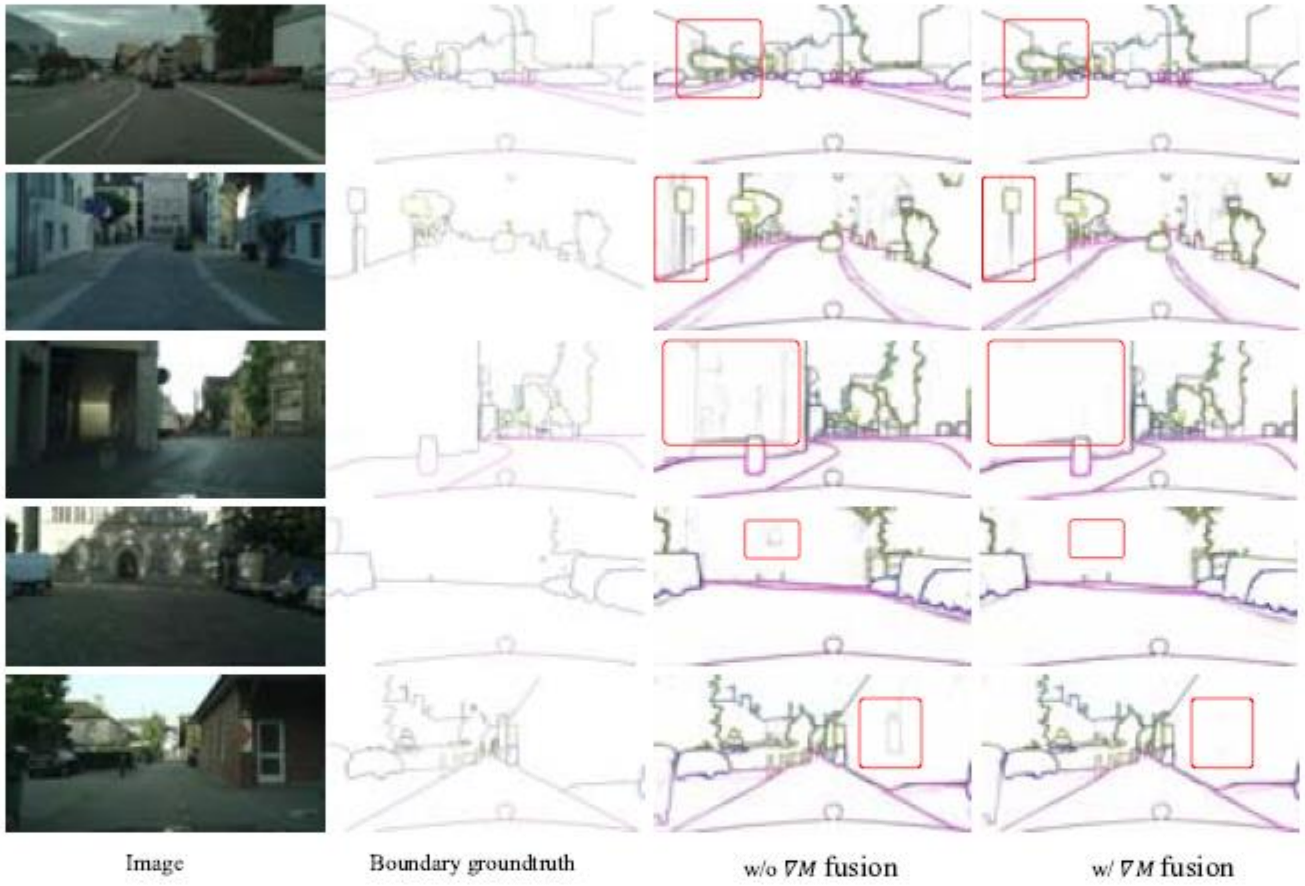


Figure 5. Some visualization comparison examples for semantic boundary detection with or without ∇M fusion (**best viewed in color**).

Experiments

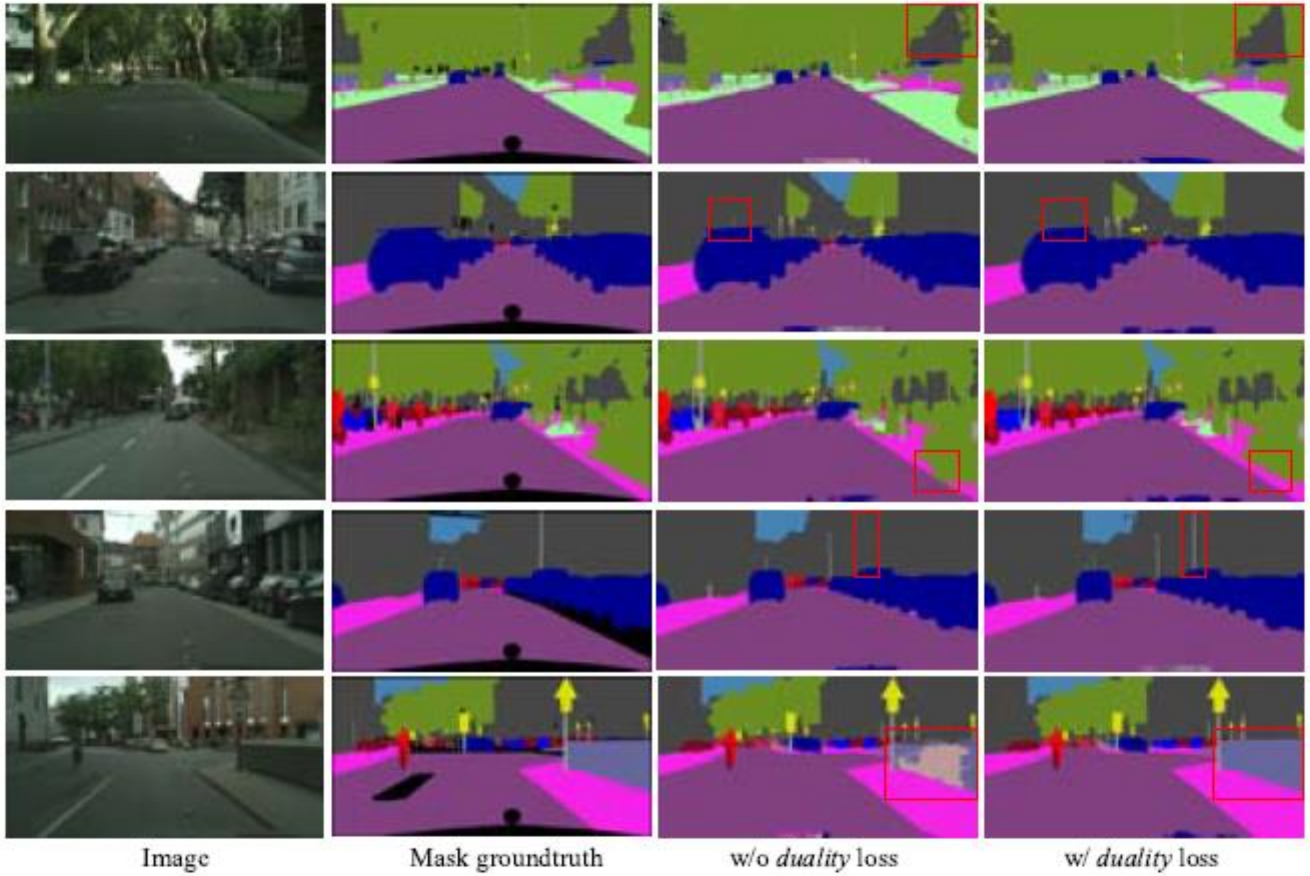


Figure 6. Some visualization comparison examples for semantic segmentation with or without duality loss used (**best viewed in color**).

Conclusion

- Good multi-task framework to joint dual problems.
- Can be applied to other tasks.