CAP: Context-Aware Pruning for Semantic Segmentation

Wei He Meiqing Wu Mingfu Liang Siew-Kei Lam

School of Computer Science and Engineering, Nanyang Technological University,
50 Nanyang Ave, Singapore

{wei005, n18061811}@e.ntu.edu.sg {meiqingwu, assklam}@ntu.edu.sg

A. Supplementary

A.1. Benchmarks details

CamVid [2] is a road scene dataset, which contains 367 training and 233 testing images at 360×480. The database provides ground truth labels associating each pixel with one semantic class. Eleven semantic classes are of common interest, including pavement, pedestrian, tree, building, sky, *etc*.

Cityscapes [3] is a large-scale urban scene dataset in driving view. There are 2,975 well-annotated and high-resolution images in the train set, 500 images in the validation set, and 1,525 images in the test set. For semantic segmentation, 19 common classes are trained and evaluated, including humans, vehicles, constructions, objects, nature, sky, *etc*.

A.2. Per-Class quantitative results comparison

In Table 6, we provide a detailed comparison of different pruning methods on the segmentation performance of each class. The results show our method's advantage that can preserve the closest accuracy from the original model in general and reduce the redundancy efficiently. Note that some baselines like NS may suffer from unrecoverable performance loss on uncommon classes for pruning lightweight models, while our pruned models can still maintain the discriminative ability in different classes.

A.3. Additional qualitative results comparison

In Figure 9, we show the visualization of predicting different images in Cityscapes. The unpruned model in Figure 9 is PSPNet101, and the pruned models are FPGM, NS-60%, BN-Scale-60% and Ours-60% in Table 1, respectively. As shown from the comparison, our method can obtain efficient compact models that perform better than other baseline pruning methods. Note that, in addition to better preserving the representation ability of unpruned model with much lesser parameters, our compact models can generate a more consistent prediction on various classes, while other baselines may suffer from the prediction loss due to misclassification or inconsistency.

A.4. Runtime acceleration results comparison

Dataset	Methods	mIoU(%)	Time(ms)	Speed(fps)(%↑)
CamVid	SegNet(Unpruned)	55.60	18	54.79
	SegNet(Ours-20%)	57.12	12	81.34(48.46%↑)
	SegNet(Ours-30%)	56.37	9	105.97(93.41%↑)
Cityscapes	PSPNet101(Unpruned)	78.40	885	1.13
	PSPNet101(Ours-60%)	77.82	658	1.52(34.51%↑)
	PSPNet101(Ours-70%)	75.27	595	1.68(48.67%↑)
	PSPNet50(Unpruned)	76.99	735	1.36
	PSPNet50(Ours-70%)	73.94	485	2.06(51.47%↑)
	ICNet(Unpruned)	64.59	14	69.91
	ICNet(Ours-60%)	62.38	13	76.27(9.10%↑)
	SegNet(Unpruned)	56.10	83	12.02
	SegNet(Ours-20%)	61.16	59	17.04(41.76%↑)

Table 4. Runtime acceleration per image inference in Cityscapes (1024×2048) and CamVid (360×480) **test set**. All inference speeds are measured by a single Tesla V100 GPU.

A.5. Comparison with other baselines

In Table 6, we compare our methods with the FPGM [5] variants in an automatic pruning manner. Due to the space limit in the main paper body, we discuss more details of FPGM [5] and CCGN [1] in this section.

A.5.1 Baselines details

Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration [5] (denoted as FPGM) indicates filter importance via its Euclidean distances to other filters in the same layer. As FPGM was originally implemented in the image classification task only, we re-implemented it for semantic segmentation. In Table 1, we showed the performance of FPGM with the setting reported in the original paper, where the filter pruning ratio in

	Method	Average	road	swalk	build.	wall	fence	pole	light	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike.	bike
PSPNet101	Unpruned	78.40	98.60	86.20	92.90	50.80	58.80	64.00	75.60	79.00	93.40	72.30	95.40	86.50	71.30	95.90	68.20	79.50	73.80	69.50	77.20
	FPGM	74.84	98.34	84.36	92.16	53.15	56.94	60.44	69.28	74.58	93.04	70.38	94.74	83.74	65.73	95.35	58.97	75.12	59.89	62.80	72.91
	NS-60%	75.70	98.21	83.09	91.99	51.85	56.00	60.45	69.23	73.82	92.84	69.04	95.02	83.51	64.94	95.05	63.55	80.45	74.21	62.49	72.62
	BN-Scale-60%	74.88	98.15	83.06	91.83	49.09	54.50	60.62	68.91	73.56	92.93	71.20	94.73	83.37	64.79	95.09	65.04	81.07	61.96	60.75	72.08
	Ours-60%	**77.82**	**98.40**	**84.78**	**92.65**	**56.74**	**57.79**	**63.58**	**72.05**	**76.57**	**93.22**	**71.60**	**95.04**	**84.91**	**67.91**	**95.60**	**68.81**	**83.82**	**75.66**	**65.23**	**74.18**
	Ours-70%	75.27	98.24	83.65	92.08	55.53	53.93	60.93	69.95	73.86	92.94	71.25	94.82	83.58	65.64	95.16	64.38	77.62	64.33	60.21	72.03
PSPNet50	Unpruned	76.99	98.41	84.52	92.64	51.01	56.53	64.13	73.19	77.18	93.27	70.45	95.21	85.43	69.80	95.72	69.55	79.52	65.86	66.14	74.33
	FPGM	74.59	98.17	83.27	92.00	54.96	54.08	59.20	70.06	72.80	92.96	70.91	94.76	83.82	65.61	95.22	62.71	**75.29**	56.24	**62.43**	72.76
	NS-50%	73.57	98.15	83.20	91.46	50.19	54.16	59.48	68.91	73.09	92.65	69.77	94.26	82.31	63.63	94.75	62.54	71.84	57.95	58.05	71.40
	BN-Scale-50%	73.85	98.19	83.50	91.54	51.64	52.85	58.84	69.02	72.96	92.73	70.31	94.33	82.15	63.27	94.86	**63.20**	74.54	**61.54**	57.14	70.64
	Ours-60%	**75.59**	**98.38**	**84.71**	**92.58**	**56.14**	**56.64**	**63.14**	**72.50**	**76.23**	**93.23**	**71.19**	**94.90**	**84.46**	**66.09**	**95.38**	60.05	73.80	61.27	62.35	**73.27**
	Ours-70%	73.94	98.21	83.55	91.95	50.88	54.67	61.34	70.90	74.81	92.89	70.86	94.76	83.73	64.79	94.95	63.06	71.55	51.13	58.90	71.87
ICNet	Unpruned	64.59	97.71	79.95	88.87	37.52	40.78	43.87	51.12	57.80	90.71	67.13	93.83	72.65	52.47	92.08	50.65	60.16	48.17	43.33	58.45
	FPGM	62.00	**97.39**	77.72	87.63	35.95	**38.51**	39.66	**44.13**	**53.36**	**90.04**	64.84	**93.05**	**70.02**	47.32	**91.22**	**49.45**	56.48	**47.74**	38.89	54.62
	NS-60%	60.02	97.20	76.73	87.08	40.95	36.56	37.27	40.28	49.25	89.44	63.56	92.68	67.79	46.05	90.18	39.96	53.68	42.49	35.64	53.59
	BN-Scale-60%	59.68	97.21	76.83	87.10	35.56	34.46	37.66	40.84	49.13	89.49	64.43	92.70	67.71	45.69	90.21	43.98	50.07	39.56	36.55	54.73
	Ours-60%	**62.38**	97.36	**78.00**	**87.88**	**41.32**	37.93	**40.83**	43.93	52.29	89.98	**65.58**	92.75	69.91	**48.46**	90.79	46.17	**59.01**	47.19	**40.27**	**55.51**
SegNet	Unpruned	56.09	95.65	70.10	82.81	29.87	31.88	38.06	43.05	44.58	87.32	62.30	91.68	67.28	50.75	87.89	21.70	29.03	34.73	40.47	56.63
	FPGM	51.60	96.20	71.11	84.02	26.19	26.67	33.26	32.17	43.78	88.24	61.92	91.10	57.16	32.50	88.28	21.68	28.31	29.41	21.25	47.17
	NS-20%	56.85	96.32	**77.51**	**88.43**	**35.00**	**37.18**	49.41	53.27	60.57	**91.18**	**67.07**	**93.91**	71.57	**46.32**	**91.42**	27.67	0.00	0.00	**35.52**	57.89
	BN-Scale-20%	59.95	97.12	77.03	88.17	33.31	36.38	48.80	50.34	59.19	90.80	66.30	93.68	70.51	44.78	91.30	29.46	41.20	29.36	33.68	57.66
	Ours-20%	**61.16**	**97.19**	77.32	88.37	32.68	37.15	**49.88**	**53.64**	**61.80**	90.95	66.23	93.74	**71.69**	45.26	91.31	**30.39**	**42.66**	**37.59**	34.95	**59.21**

Table 5. Per-class results after pruning on Cityscapes **test set**.

	Methods	mIoU(%)	#Params(M)(%↓)	#FLOPs(G)(%↓)
PSPNet101	Unpruned	77.48	70.44	557.04
	FPGM-A-20%	70.94	53.10(24.62%↓)	397.22(28.69%↓)
	FPGM-A-30%	67.20	46.81(33.54%↓)	346.25(37.84%↓)
	Ours-60%	**78.23**	47.84(32.08%↓)	363.21(34.80%↓)
	Ours-70%	75.40	**39.74(43.58%↓)**	**296.25(46.82%↓)**
PSPNet50	Unpruned	76.57	51.45	403.0
	FPGM-A-30%	60.14	35.17(31.63%↓)	273.53(32.13%↓)
	FPGM-A-40%	61.36	23.56(54.21%↓)	185.69(53.92%↓)
	Ours-60%	**75.65**	27.31(46.92%↓)	233.67(42.02%↓)
	Ours-70%	74.31	23.78(53.78%↓)	203.19(49.58%↓)
	Ours-80%	70.83	**21.16(58.87%↓)**	**179.79(55.39%↓)**
ICNet	Unpruned	64.59	12.21	40.13
	FPGM-A-20%	49.20	10.47(14.30%↓)	27.03(32.64%↓)
	FPGM-A-30%	43.01	8.86(46.57%↓)	24.62(38.65%↓)
	Ours-60%	**63.26**	**5.56(54.46%↓)**	**21.16(47.27%↓)**
SegNet	Unpruned	56.10	29.45	326.59
	FPGM-A-10%	44.91	27.41(6.93%↓)	178.09(45.46%↓)
	FPGM-A-40%	37.17	10.85(63.18%↓)	**43.74(86.61%↓)**
	Ours-20%	**60.98**	**10.76(63.46%↓)**	178.23(45.43%↓)

Table 6. Quantitative pruning results on Cityscapes **validation set**.

each layer is predefined. The pruned architectures of FPGM in Table 1 are shown in Figure 5 and Figure 6. In Table 6, we show the results of FPGM using the automatic pruning method like ours, where we only set a global pruning ratio and prune filters in a global and greedy manner. We denote this method as FPGM-A. Same as other reported baselines in Table 1, the *x* in FPGM-A-x% stands for the global threshold ratio, and we also reserve 10% filters to prevent pruning out the whole layer.

Batch-Shaping for Learning Conditional Channel Gated Networks [1] (denoted as CCGN in Table 3) is the state-of-the-art conditional computing method, which estimates channel saliency by introducing a gated module similar to *Dynamic Channel Pruning* [4], but provides a better trade-off between simple and complex examples inference. Although it is not strictly a network pruning method, it is the latest work to provide comprehensive network acceleration results on the large-scale semantic segmentation benchmarks. Hence, we provide a comparison with this method as well. As stated in their paper, CCGN(Without pretrain) stands for the model that undertakes training without ImageNet-pretrained, while CCGN-1(With pretrain) and CCGN-2(With pretrain) are with pretrained and reduce FLOPs in different percentage to balance the performance.

A.5.2 Analysis

Table 6 shows that our method outperforms the above-mentioned state-of-the-art pruning methods. It can also be observed that when FPGM is implemented in an automatic pruning manner (*i.e*., FPGM-A-x%), the performance becomes worse (compared to FPGM in Table 1). From the observation, it is evident that our method serves as a better global indicator to identify the importance of channels, while some pruning criteria in image classification task may not be effective for semantic segmentation.

A.6. Pruned structures comparison

In Figure 5 to Figure 8, we visualize the pruned architectures using our framework, *i.e*., CAP, and the original FPGM.

References

[1] Babak Ehteshami Bejnordi, Tijmen Blankevoort, and Max Welling. Batch-shaping for learning conditional channel gated networks. In *International Conference on Learning Representations*, 2020.

[2] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng zhong Xu. Dynamic channel pruning: Feature boosting and suppression. In *International Conference on Learning Representations*, 2019.

[5] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

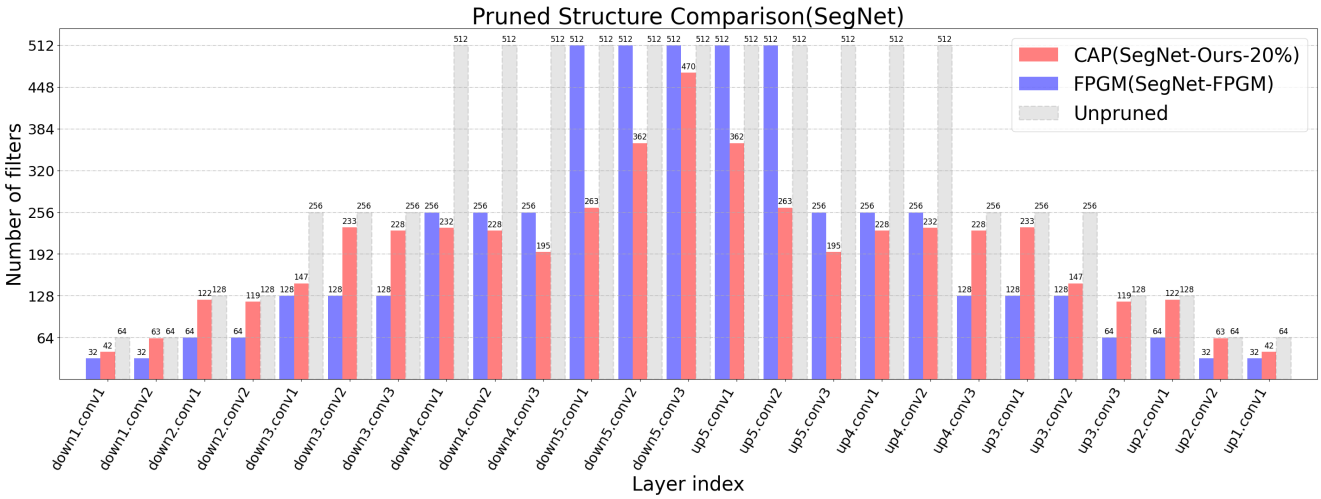


Figure 5. Pruned structure comparison (SegNet)

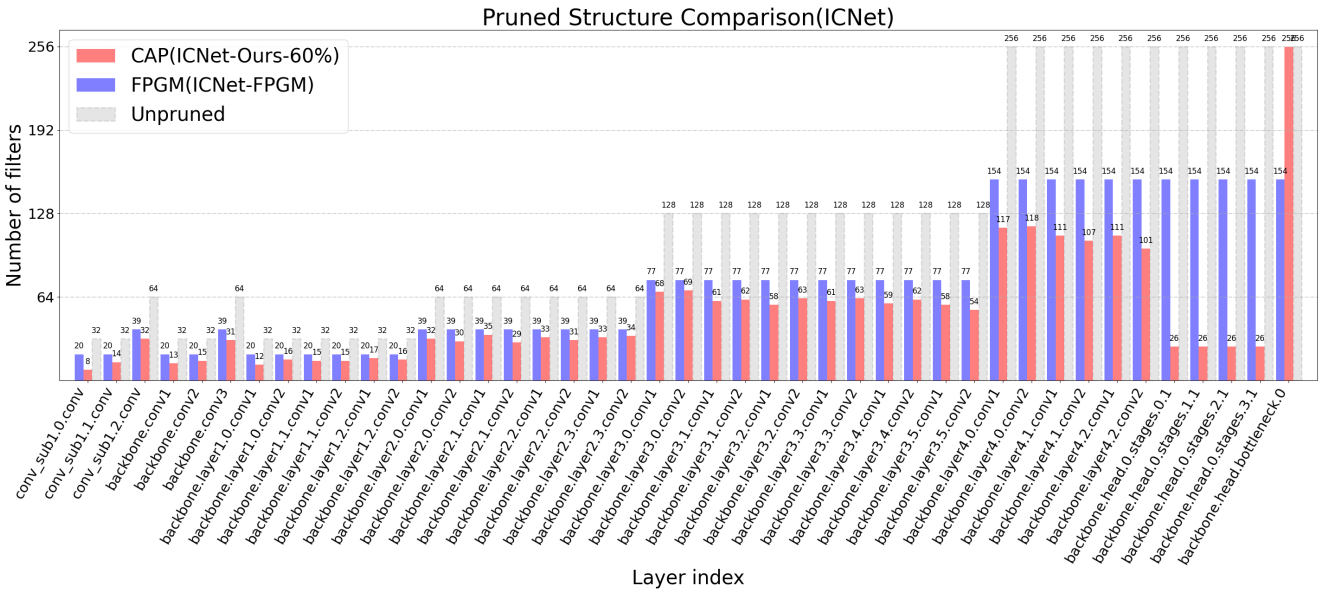


Figure 6. Pruned structure comparison (ICNet)

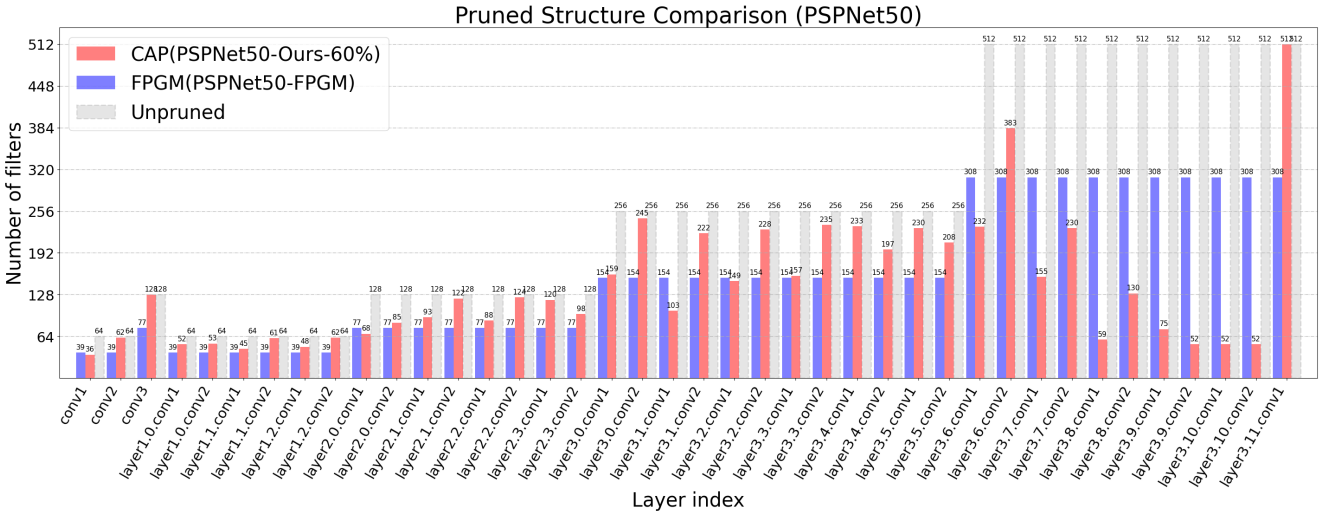


Figure 7. Pruned structure comparison (PSPNet50)

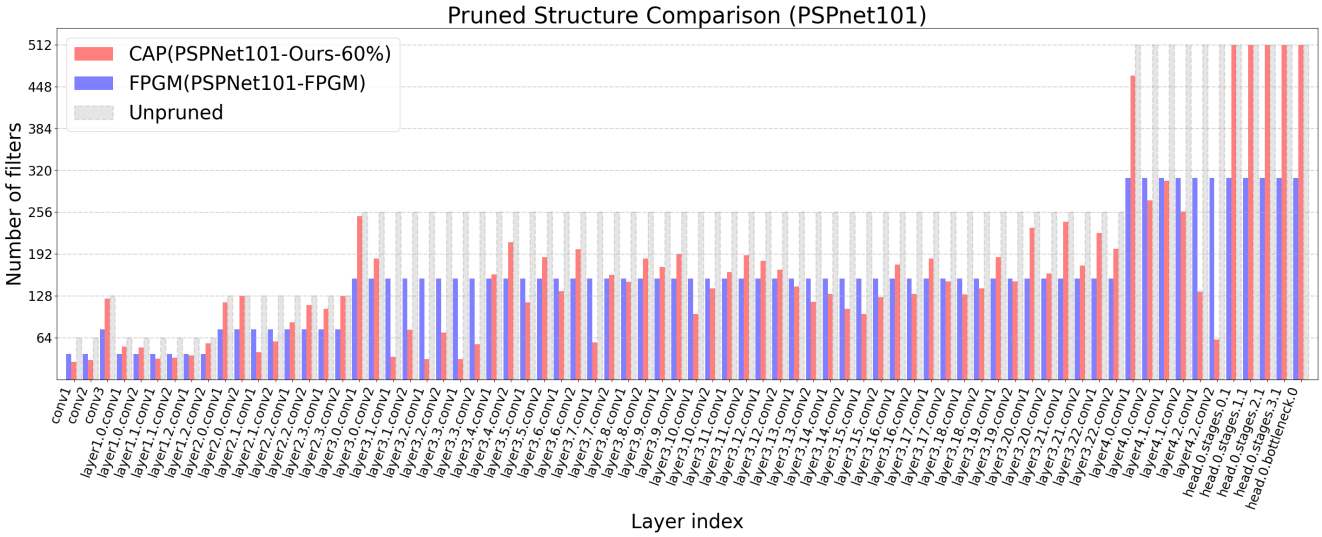
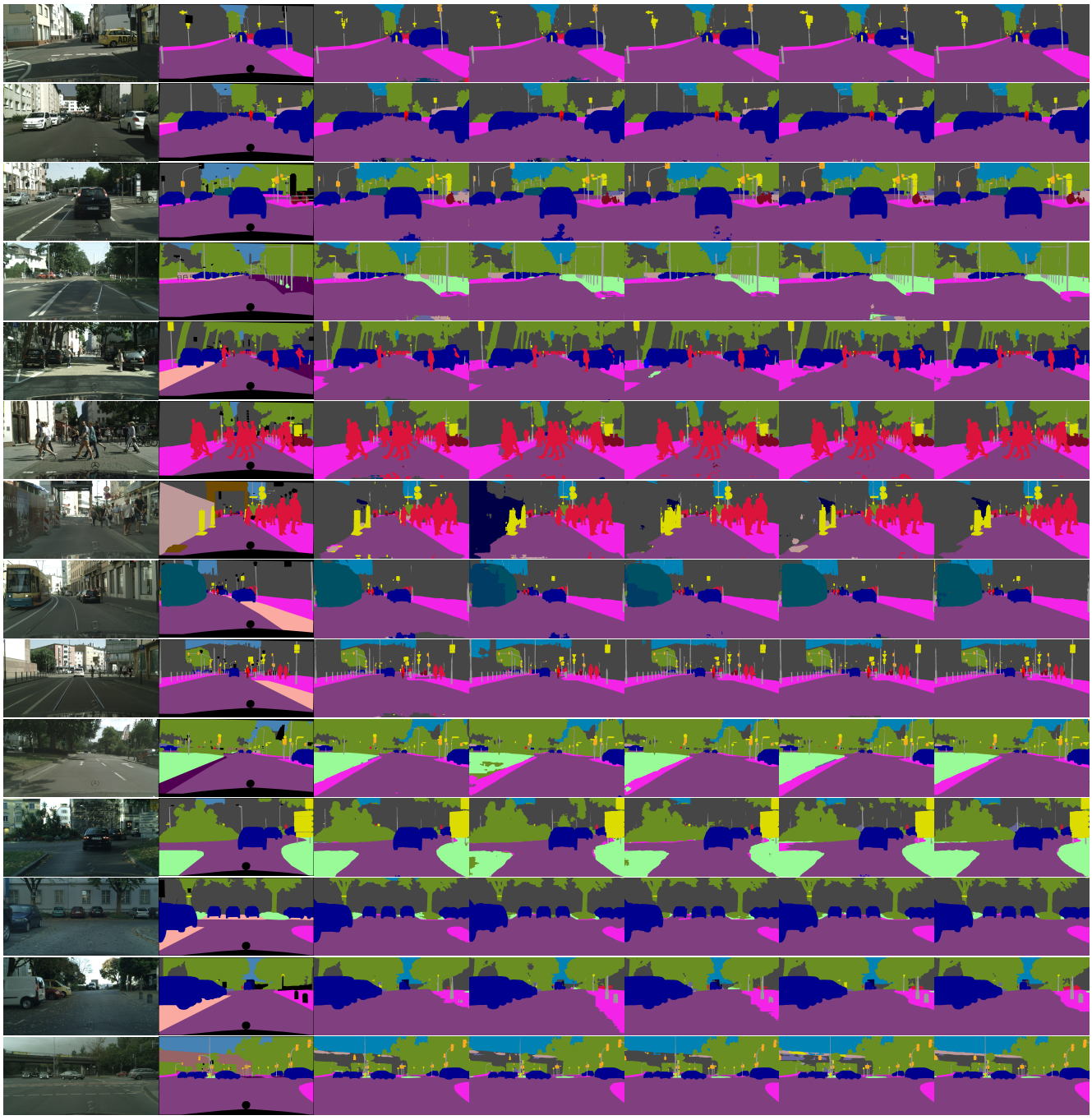


Figure 8. Pruned structure comparison (PSPNet101)



(a) Input Image (b) GT (c) FPGM (d) NS (e) BN-Scale (f) Unpruned (g) Ours

Figure 9. Extra qualitative comparison on different images on Cityscapes **validation set**.