Dense Prediction with Attentive Feature Aggregation - Supplementary Material

Yung-Hsu Yang[1] royyang@gapp.nthu.edu.tw
Thomas E. Huang[2] thomas.huang@vision.ee.ethz.ch
Min Sun[1] sunmin@ee.nthu.edu.tw
Samuel Rota Bulò[3] rotabulo@fb.com
Peter Kontschieder[3] pkontschieder@fb.com
Fisher Yu[2] i@yf.io
[1]National Tsing Hua University [2]ETH Zürich [3]Facebook Reality Labs

This supplementary material provides additional results on boundary detection benchmarks, ablation study on post-processing effects on semantic segmentation, training settings and more qualitative results of the attention maps and output predictions.

1. Full Results on Boundary Detection

To complete the results in Table 5 and Table 6 in the main paper, we provide the full evaluation results on the BSDS500 [1] and the NYUDv2 [10].

In Table 1, we show more evaluation results of using the PASCAL VOC Context dataset (PVC) [7] as additional training data and multi-scale inference for BSDS500. When using PVC, we double our training epochs to account for the additional data. For multi-scale inference, we use standard average pooling for fair comparison with other methods. AFA-DLA achieves state-of-the-art results on single-scale inference when not training with additional data. Surprisingly, using PVC does not further improve the results. Nevertheless, AFA-DLA achieves the same performance with the state-of-the-art method BDCN [5] when using multi-scale inference.

In Table 2, we report more evaluation results of using three different types of inputs. Our AFA-DLA model outperforms all other methods by a large margin across all three types of inputs, achieving state-of-the-art performances. When only using RGB images as input, AFA-DLA already outperforms some other methods using both RGB and HHA images.

Table 1. Boundary detection results on BSDS500. PVC indicates training with additional PASCAL VOC Context dataset. MS indicates multi-scale inference. AFA-DLA achieves state-of-the-art results on single-scale images without using additional data, and competitive results when using both PVC and MS.

Method	PVC	MS	ODS	OIS
Human			0.803	0.803
DLA [14]			0.803	0.813
LPCB [4]			0.800	0.816
BDCN [5]			0.806	**0.826**
AFA-DLA (Ours)			**0.812**	**0.826**
RCF [6]	✓		0.808	0.825
LPCB [4]	✓		0.808	0.824
BDCN [5]	✓		**0.820**	**0.838**
PiDiNet [11]	✓		0.807	0.823
AFA-DLA (Ours)	✓		0.810	0.826
RCF [6]	✓	✓	0.814	0.833
LPCB [4]	✓	✓	0.815	0.834
BDCN [5]	✓	✓	0.828	0.844
AFA-DLA (Ours)	✓	✓	**0.828**	**0.844**

2. Ablation Study on Post-processing of Semantic Segmentation

To have a fair competition with other methods, we exploit several post-processing techniques to pursue higher performance. We conduct an ablation study on how each technique affects the final performance on the Cityscapes [3] validation set in Table 3. The main improvement gains are from our Scale Space Rendering (SSR) for multi-scale inference, and the other techniques only bring minor improvements.

3. Training Losses

In this section, we describe in more detail the formulation of our loss function for AFA-DLA for both semantic segmentation and boundary detection.

3.1. Semantic Segmentation

We use k scales for training and RMI [17] to be the primary loss for our final prediction $P_{\texttt{final}}$, i.e.,

$$L_{\texttt{primary}} \triangleq L_{\texttt{rmi}}(\hat{P}, P_{\texttt{final}}), \tag{1}$$

Table 2. Boundary detection results on NYUDv2 using three different types of inputs. AFA-DLA achieves state-of-the-art results across all three settings.

Method	Input	ODS	OIS
AMH-Net [6]	RGB	0.744	0.758
BDCN [5]		0.748	0.763
PiDiNet [11]		0.733	0.747
AFA-DLA (Ours)		**0.762**	**0.775**
AMH-Net [6]	HHA	0.716	0.729
BDCN [5]		0.707	0.719
PiDiNet [11]		0.715	0.728
AFA-DLA (Ours)		**0.718**	**0.730**
AMH-Net [6]	RGB+HHA	0.771	0.786
BDCN [5]		0.765	0.781
PiDiNet [11]		0.756	0.773
AFA-DLA (Ours)		**0.780**	**0.792**

where $\hat{P}$ is the ground truth and $L_{\texttt{rmi}}$ is the RMI loss function. The first auxiliary cross-entropy loss is computed by using the generated scale-space rendering (SSR) attention to fuse the auxiliary per-scale predictions from the OCR [15] module, yielding

$$L_{\texttt{ocr}} \triangleq L_{\texttt{ce}}(\hat{P}, P^{\texttt{aux}}_{\texttt{ocr}}) \,, \tag{2}$$

where $L_{\texttt{ce}}$ denotes the cross-entropy loss. For the second auxiliary loss, we compute and sum up cross-entropy losses for each scale prediction P_i, where $1 \leq i \leq k$, yielding

$$L_{\texttt{scale}} \triangleq \sum_{i=1}^{k} L_{\texttt{ce}}(\hat{P}, P_i) \,. \tag{3}$$

Lastly, for the auxiliary loss inside AFA-DLA, we fuse the predictions of each auxiliary segmentation head with SSR across scales and get $P^{\texttt{aux}}_j$, where $1 \leq j \leq 4$. We compute the auxiliary loss for each prediction and sum them up as

$$L_{\texttt{aux}} \triangleq \sum_{j=1}^{4} L_{\texttt{ce}}(\hat{P}, P^{\texttt{aux}}_j)) \,. \tag{4}$$

Accordingly, the total loss function is the weighted sum as

$$L_{\texttt{seg}} \triangleq L_{\texttt{primary}} + \beta_o L_{\texttt{ocr}} + \beta_s L_{\texttt{scale}} + \beta_a L_{\texttt{aux}}, \tag{5}$$

where we set $\beta_o = 0.4$, $\beta_s = 0.05$, and $\beta_a = 0.05$.

3.2. Boundary Detection

For boundary detection, we opted to using a simpler version of the loss function for semantic segmentation. We use standard binary cross entropy (BCE) to be the primary loss for our final prediction $P_{\texttt{final}}$, i.e.,

$$L_{\texttt{primary}} \triangleq L_{\texttt{bce}}(\hat{P}, P_{\texttt{final}}) \,, \tag{6}$$

Table 3. Ablation study on Cityscapes validation set with AFA-DLA-X-102 for validating each post-processing technique. SSR indicates our Scale Space Rendering.

SSR	Flip	Seg-Fix [16]	mIoU
-	-	-	83.06
✓	-	-	84.81
✓	✓	-	85.00
✓	✓	✓	85.10

Table 4. Specific training settings for each dataset. BS stands for training batch size.

Dataset	Crop Size	BS	Training Epochs
Cityscapes	2048×1024	8	375
BDD100K	1280×720	16	200
BSDS500	416×416	16	10
NYUDv2	480×480	16	54

where $\hat{P}$ is the ground truth and $L_{\texttt{bce}}$ is the BCE loss function.

We also use auxiliary segmentation heads to make predictions at each feature level. Each prediction $P^{\texttt{aux}}_j$, where $1 \leq j \leq 4$, is upsampled to the original scale and the BCE loss is used to compute the auxiliary loss, i.e.,

$$L_{\texttt{aux}} \triangleq \sum_{j=1}^{4} L_{\texttt{bce}}(\hat{P}, P^{\texttt{aux}}_j)) \,. \tag{7}$$

Accordingly, the total loss function is the weighted sum as

$$L_{\texttt{bd}} \triangleq L_{\texttt{primary}} + \beta_a L_{\texttt{aux}}, \tag{8}$$

where we set $\beta_a = 0.05$.

4. Implementation Details

We provide the general training setting and procedure used for training on Cityscapes [3], BDD100K [13], BSDS500 [1], and NYUDv2 [10].

We use PyTorch [8] as our framework and develop based on the NVIDIA semantic segmentation codebase [1]. The general training procedure is using SGD [9] with momentum of 0.9 and weight decay of 10^{-4}. Specific settings for each dataset are shown in Table 4.

4.1. Semantic Segmentation

We use an initial learning rate of 1.0×10^{-2}. We use the learning rate warm-up over the initial 1K training iterations and the polynomial decay schedule, which decays the initial learning rate by multiplying $(1 - \frac{\texttt{epoch}}{\texttt{max\_epochs}})^{0.9}$ every

[1]NVIDIA license: https://github.com/NVIDIA/semantic-segmentation/blob/main/LICENSE

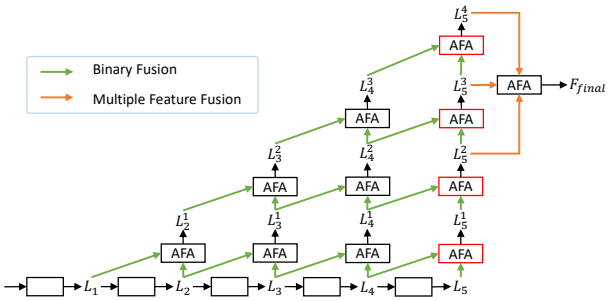

Figure 1. Notation of aggregated features of AFA-DLA.

epoch. We apply random image horizontal flipping, randomly rotating within 10 degrees, random scales from 0.5 to 2.0, random image color augmentation, and random cropping. As in [18], we also use class uniform sampling in the data loader to overcome the data class distribution unbalance problem. Due to limitations in computational power, we further use Inplace-ABN [2] to replace the batch norm and ReLU function to acquire the largest possible training crop size and batch size on 8 Tesla V-100 32G GPUs.

4.2. Boundary Detection

We use an initial learning rate of 1.0×10^{-2} and a batch size of 16 for both BSDS500 [1] and NYUDv2 [10]. We use the step decay schedule and drop the learning rate by 10 times at around $0.55 \times$ `max_epochs` and then again at $0.85 \times$ `max_epochs`. For augmentation, we follow the standard protocol in literature [12, 6] and apply random flipping, scaling by 0.5 and 1.5, and rotation by 16 different angles. We train all our models on a single GeForce RTX 2080Ti GPU.

5. Visualization of Attention Maps

In this section, we provide more visualizations of attention maps generated by our proposed AFA module. For reference, we provide a detailed architecture of AFA-DLA and denote the notation of aggregated features of different levels in Figure 1.

We first look at the attention maps generated by our binary fusion module which aggregates two features in Figure 2 and Figure 3. We provide the spatial attention maps for binary fusion at four different levels. When the difference of the level information between two input features is larger (e.g., L_4^3 and L_5^3), our attention mask will become more specific and be able to focus on the right place to be fused. Take the fusion of L_4^3 and L_5^3 as example. Since L_4^3 contains the information of the L_1 feature, our attention focuses on object boundaries on it and attend to the rest on L_5^3, which has richer semantic information. Compared to linear fusion operations, our AFA module provides a more expressive way of combining features.

We additionally look at the spatial attention maps generated by our multiple feature fusion module in Figure 4. Only using the final aggregated features for prediction may cause our model to overly focus on low level features. Thus, our multiple feature fusion module provides the model with more flexibility to select between the features that contain different low level information. For input features that contain L_1 information like L_5^4 and L_5^3, the attention focuses more on the object boundaries, similar to our binary fusion module. For other input features like L_5^2, the attention can focus on objects or the background. With our multiple feature fusion module, our model can strike a balance between the low-level and the high-level information and perform fusion accordingly.

6. Qualitative Results

We provide more qualitative results in this section to visualize AFA-DLA's predictions. We show full predictions of AFA-DLA on Cityscapes in Figure 5, BDD100K in Figure 6, BSDS500 in Figure 7, and NYUDv2 in Figure 8. The results on Cityscapes show that our model can handle both fine and coarse details well and is robust towards different input scenes. On BDD100K, the results show the ability of our model to handle more diverse urban scenes, with varying weather conditions and times of the day. On both BSDS500 and NYUDv2, our model can predict both fine-grained scene details as well as object-level boundaries. In particular, on NYUDv2, our model can recover more boundaries than the ground truth. With results across different types of datasets and both semantic segmentation and boundary detection, AFA-DLA demonstrates its strong performance and applicability for dense prediction tasks.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

[2] Samuel Rota Bulo, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5639–5647, 2018.

[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] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 562–578, 2018.

[5] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2019.

[6] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3000–3009, 2017.

[7] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[9] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[10] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[11] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5117–5127, 2021.

[12] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[13] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[14] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[15] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pages 173–190. Springer, 2020.

[16] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *European Conference on Computer Vision*, pages 489–506. Springer, 2020.

[17] Shuai Zhao, Yang Wang, Zheng Yang, and Deng Cai. Region mutual information loss for semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.

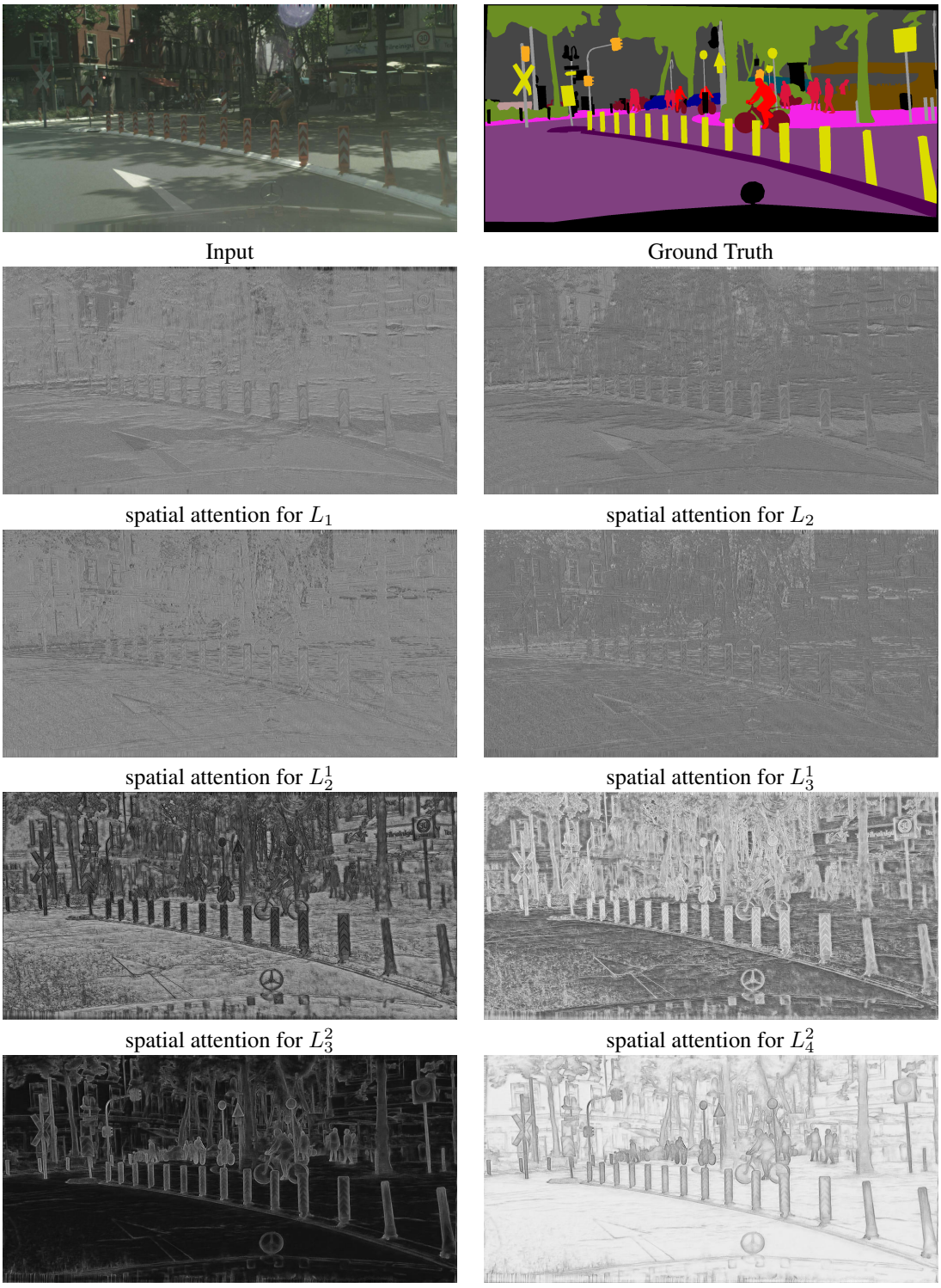

Figure 2. Spatial attention maps at four different levels generated by our binary fusion module which aggregates two features. Whiter regions denote higher attention. Compared to linear fusion operations, our AFA module provides a more expressive way of combining features.

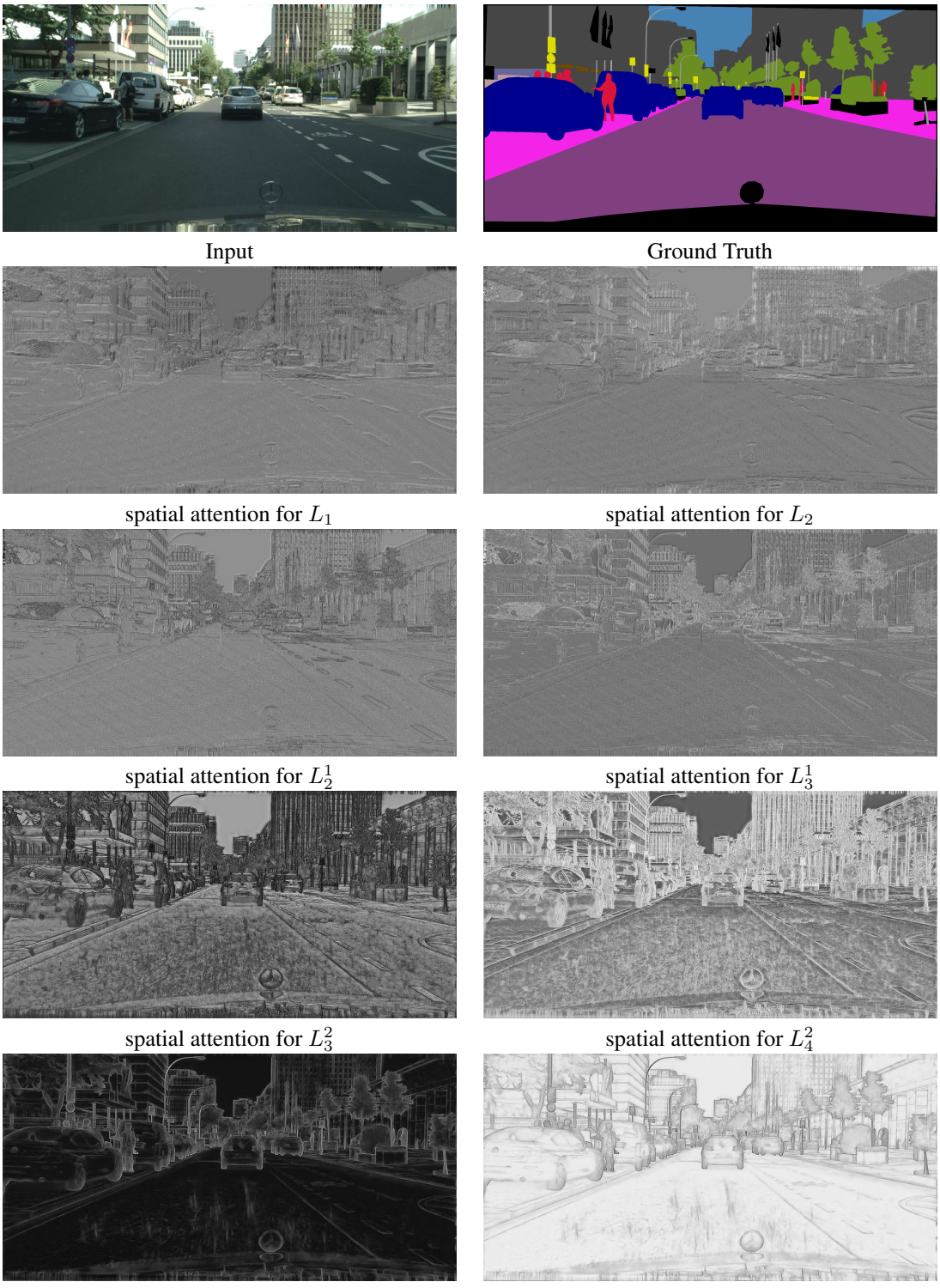

Figure 3. Spatial attention maps at four different levels generated by our binary fusion module which aggregates two features. Whiter regions denote higher attention. Compared to linear fusion operations, our AFA module provides a more expressive way of combining features.

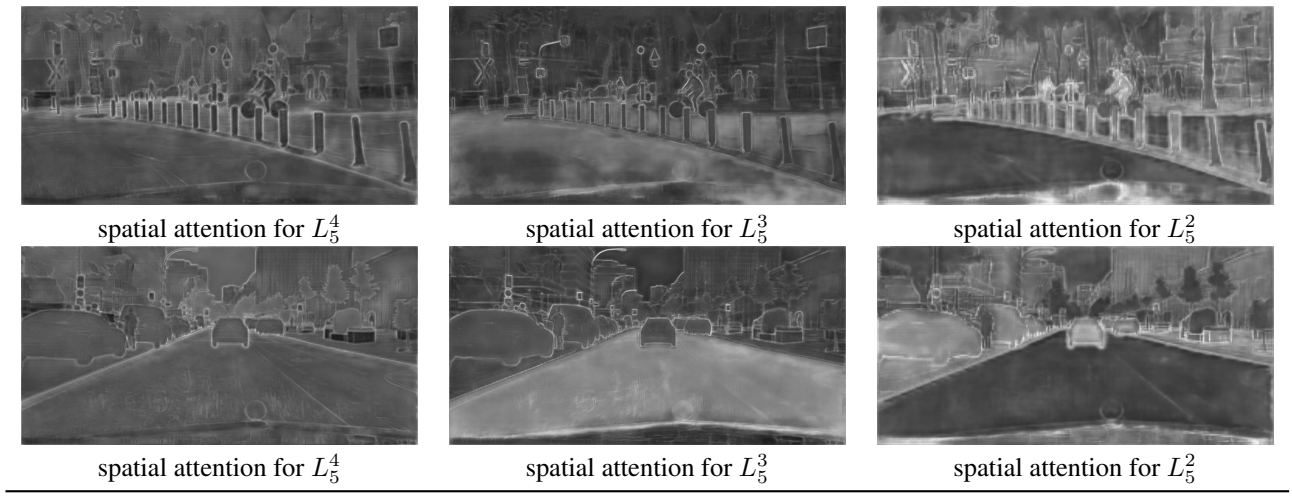

Figure 4. Spatial attention maps generated by our multiple feature fusion module which aggregates multiple features. Whiter regions denote higher attention. With our multiple feature fusion module, our model can strike a balance between the low-level and the high-level information and perform fusion accordingly.

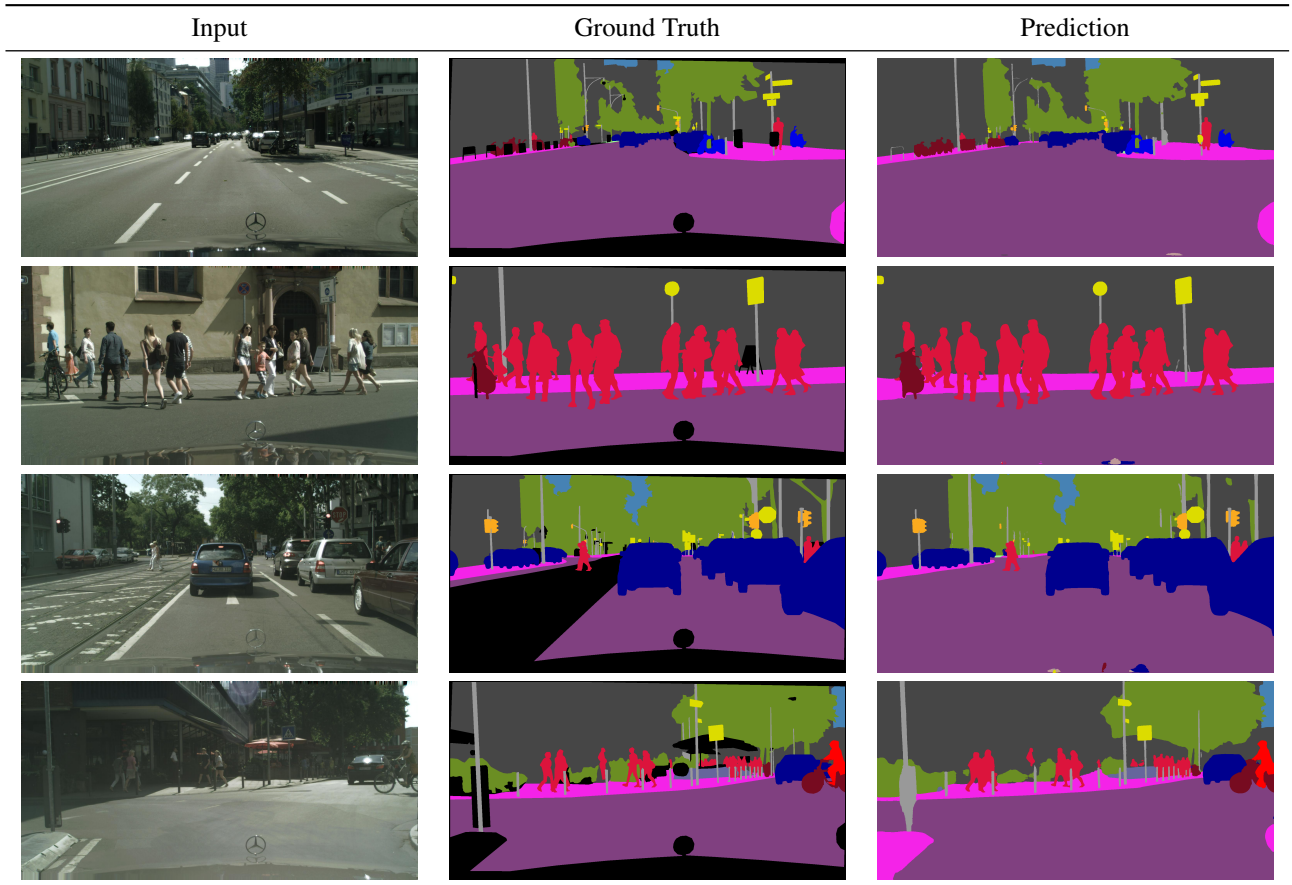

Figure 5. The qualitative results of AFA-DLA-X-102 on the Cityscapes validation set. Our model can handle both fine and coarse details well and is robust towards different input scenes.

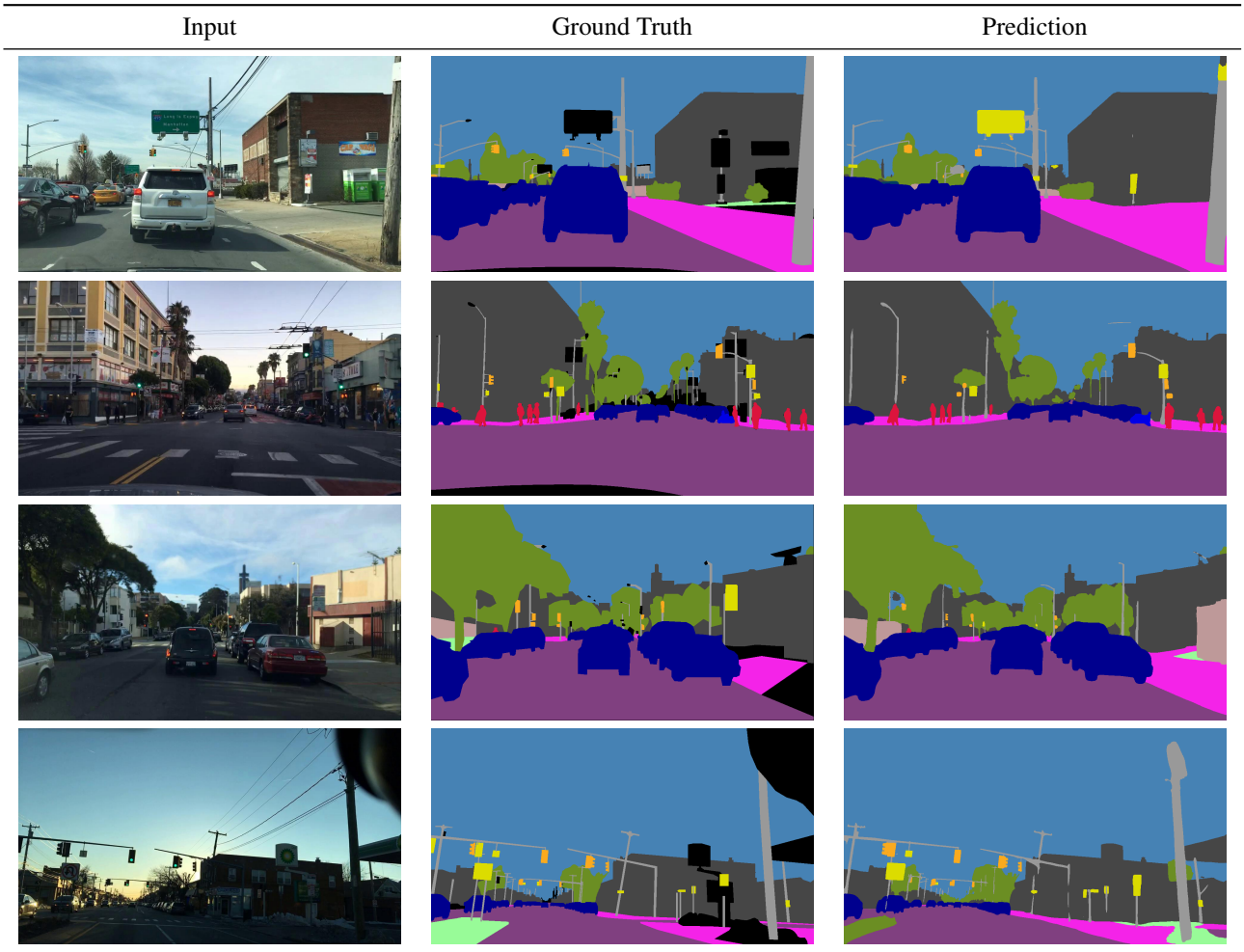

Figure 6. Qualitative results of AFA-DLA-169 on the BDD100K validation set. Our model can handle diverse urban scenes, with varying weather conditions and times of the day.

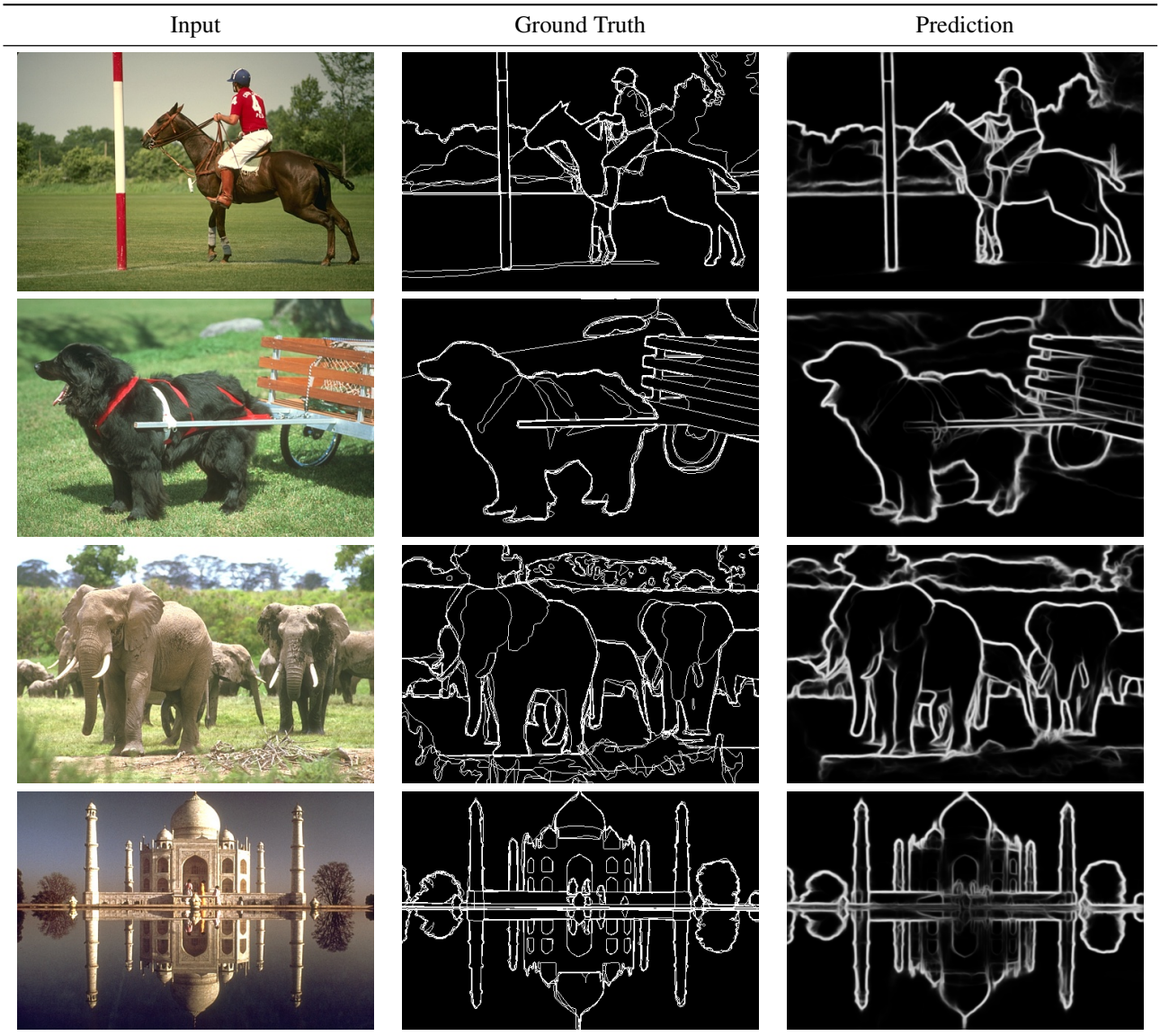

Figure 7. Qualitative results of AFA-DLA-34 on the BSDS500 test set. Results are raw boundary maps obtained using multi-scale inference before Non-Maximum Suppression. Our model can predict both fine-grained scene details and object-level boundaries.

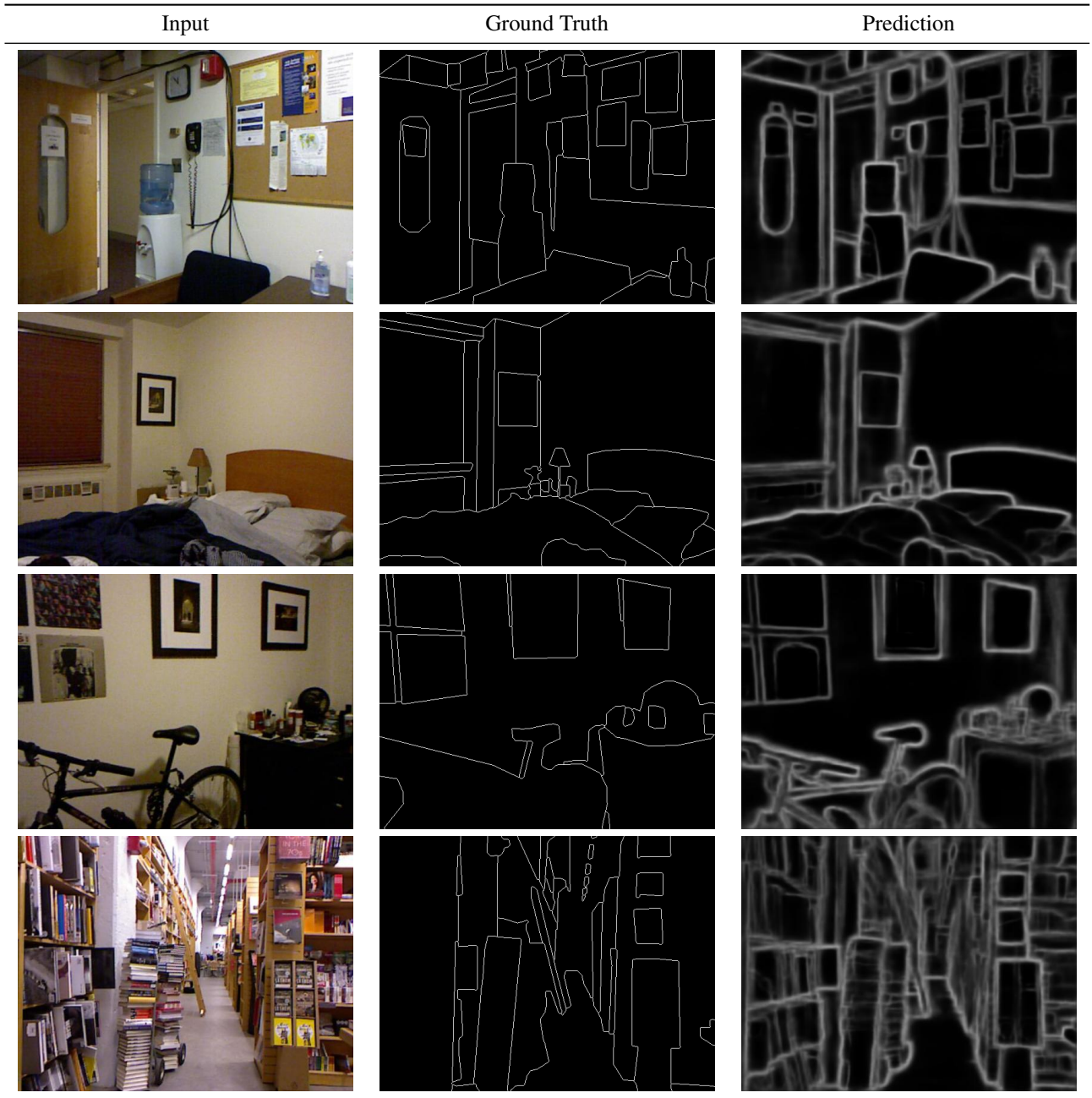

Figure 8. Qualitative results of AFA-DLA-34 on the NYUDv2 test set. Results are raw boundary maps obtained by averaging predictions on both RGB and HHA images before Non-Maximum Suppression. Our model can extract more boundaries than the ground truth.