Close the Loop: A Unified Bottom-up and Top-down Paradigm for Joint Image Deraining and Segmentation (Supplementary Materials)

Yi Li[1], Yi Chang[1,2*], Changfeng Yu[1], Luxin Yan[1]

[1] National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2] Pengcheng Laboratory
{li_yi, yichang, ycf, yanluxin}@hust.edu.cn

1 Introduction

For supplementary materials, we firstly present analyze to further demonstrate the influence of different degradations types on semantic segmentation in Sec. 2. Then we provide more implementation details about the proposed unified bidirectional cooperation network (UBCN) in the Sec. 3. Moreover, limitations about UBCN are analyzed in the Sec. 4. Finally, more visualized results on real-world rainy images are presented in the Sec. 5.

2 Influence of Different Degradations Types to Semantic Segmentation

In the main manuscript, we have analyzed the influence of different rain levels on segmentation performance. In this supplementary, we move one step forward to analyze the influence of different degradation types on segmentation.

We evaluate the performance of the segmentation model when applying images with different degradation types and levels for training and testing. Specifically, we mainly focus on four kinds of degradations: rain, pepper noise, haze, and gaussian noise. Specifically, we synthesize four levels of degradation on *Cityscapes* (Cordts et al. 2016) dataset. Rain: 0mm/hr, 50mm/hr, 150mm/hr, 250mm/hr. Pepper noise ratio: 0%, 30%, 50%, 70%. Haze beta[1]: 0.0, 0.05, 0.01, 0.03. Gaussian noise sigma: 0, 30, 50, 70. For each degradation, we train four PSPNet (Zhao et al. 2017) models on four levels of synthetic degradation images respectively. Then we test segmentation performance on images with different degradation levels and show segmentation results, corresponding to different color lines in Fig. 1.

We have several observations. First, same observation as in the main manuscripts, the best performance of all cases are acquired through adopting both clear images for training and testing. This phenomenon indicates that all degradation will do harm to the segmentation performance, and the clean image is still most discriminative for segmentation without any ambiguity caused by the artifacts. Second, the best segmentation result for each degradation type and degradation level is always obtained only when the training dataset and the testing dataset match, which denotes segmentation models need to be adaptively associated with the degradation dataset. Third, it is interesting that the curves from different degradation types possess different trends. While the results of all segmentation models decrease monotonically in rain and pepper noise, results of haze and gaussian noise are highly related to the gap between training and testing sets. Specifically, the smaller the gap is, the better the segmentation result is. Thus we can conclude that *different degradation types have a different impact on semantic segmentation.* For rain and pepper noise, as non-uniform degradations applying to part of the pixels, the segmentation performance decreases monotonically as the degradation level increases, indicating the lighter the degradation in test images is, the better the segmentation result is. For haze and gaussian noise, as uniform degradations applying to the whole image, adopting corresponding levels of degradation images for training and testing can achieve better performance.

The first two phenomenons demonstrate that both proper degradation removal and segmentation adaptation could indeed facilitate semantic segmentation performance, which further proves the rationality of the proposed joint image & feature domain adaptation strategy in main manuscripts. The third phenomenon indicates that we should treat different degradation types differently, such as non-uniform degradation (rain, pepper noise) and uniform degradation (haze, gaussian noise), which is an interesting phenomenon and can be further explored in our future work.

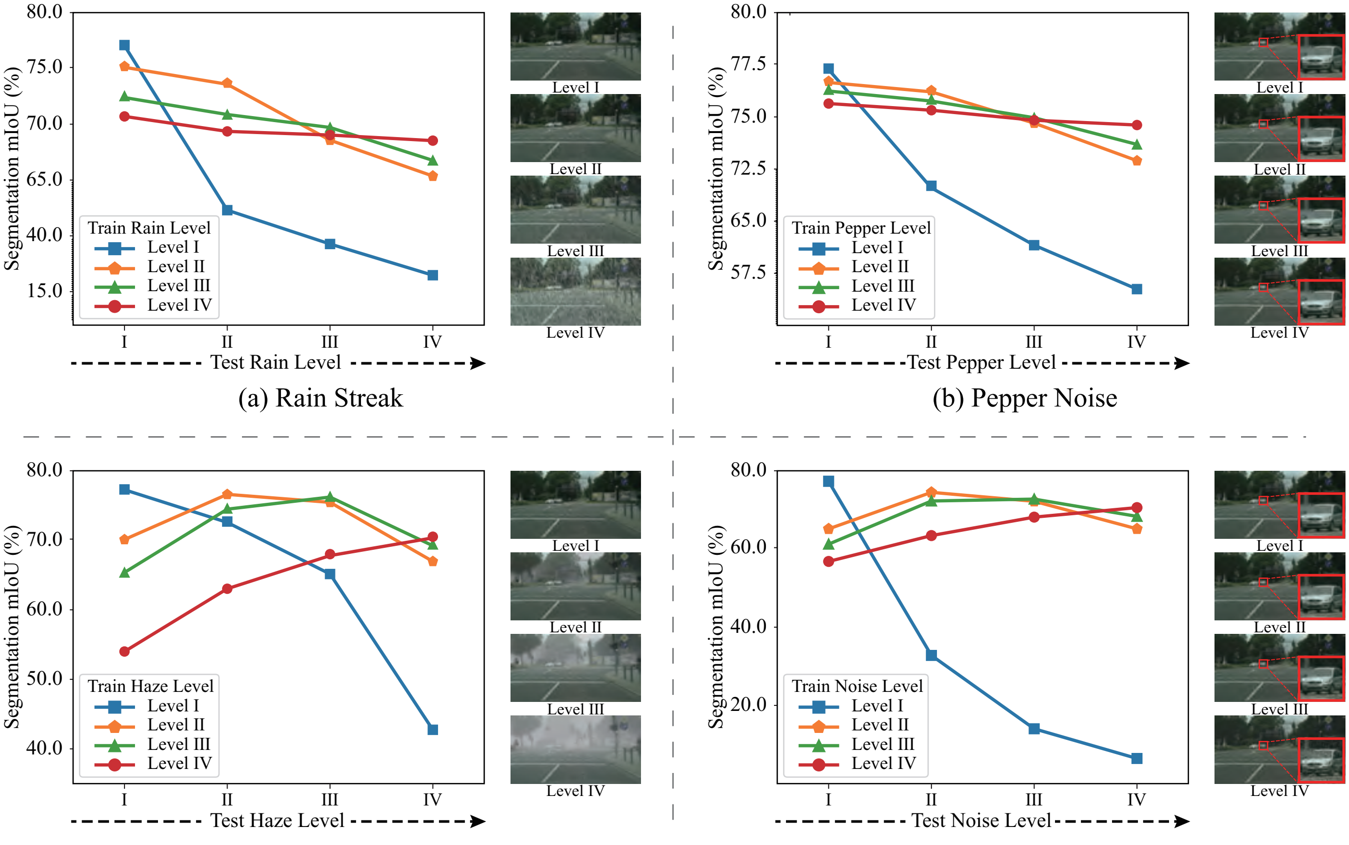
3 More Implementation Details

3.1 Network Settings

The proposed UBCN mainly consists of three parts: image-level coarse derain module, feature-level adaptation segmentation module, and semantic attentive module. Specifically, the image-level deraining module is composed of 22 residual blocks, as shown in Tab. 1. Since the segmentation model in the feature-level adaptation segmentation module is replaceable, we mainly formulate the discriminator part, detail structure as shown in Tab. 2. As for the proposed SAM, different datasets result in different number of paths. For *Cityscapes* dataset, which is composed of 19 semantic categories, we set the number of paths to be 10 for network efficiency, including 9 dominate categories (road, sidewalks,

[*] Corresponding Author

[1] Same definition in (Sakaridis, Dai, and Van Gool 2018)



(a) Rain Streak

(b) Pepper Noise

(c) Haze

(d) Gaussian Noise

Figure 1: Analysis of the influence of different degradations on segmentation. We train the same segmentation model (Zhao et al. 2017) with four different kinds of degradations: (a) rain streak, (b) pepper noise, (c) haze, and (d) gaussian noise. For each degradation type, we train segmentation models on four degradation levels (colored lines), and show the corresponding segmentation results in a confusing matrix format.

building, *etc.*) and 1 ”others” categories (fence, pole, *etc.*) according to the percentage of corresponding pixels. For *VOC2012* dataset, we set the path number to be 2 corresponding 1 foreground category and 1 background category. The detailed architecture of SAM is shown in Tab. 3.

3.2 Training Details

We implement the proposed UBCN in Pytorch. The experiments are carried out with 4 NVIDIA RTX 2080Ti GPUs. *We first pretrain each module in the proposed UBCN, and then finetune the whole network.* Specifically, we train the image-level derain module with Adam optimizer and 1e-2 learning rate for 100 epochs. During training, we initialize the weights through normalized initialization and set mini-batch size to 24. The learning rate decrease to 1/5 every 30 epochs. Then we train the feature-level adaptation module for 100 epochs with SGD optimizer. The loss weight of the original segmentation model and discriminator is set to be 1 and 1e-3 respectively. And the learning rate of the segmentation model and discriminator are 1e-2 and 1e-3 respectively. Moreover, we train the proposed semantic attentive module with the output of image-level derain module and feature-level adaptation module. And we adopt Adam optimizer and 1e-2 learning rate for 100 epochs training. Finally, we finetune the whole network for 50 epochs. We adopt SGD as the optimizer with an initial learning rate 1e-4. The size of mini-batch is 12 and the learning rate is decayed through poly strategy with 0.9 momentum.

4 Limitation

Although the proposed UBCN has achieved state-of-the-art deraining performance, there is no free lunch. We report parameters comparison in Table 4. Due to the large segmentation sub-network, the proposed UBCN achieves 191.40MB parameters, including 13.40MB for image-level derain module, 0.40 MB for SAM, and 177.60MB for feature-level adaptation segmentation module, which needs more resources for network training and inference. Therefore, adopting a light-weight and efficient segmentation model for joint deraining and segmentation might be a promising solution and can be further explored in the future.

Moreover, we report the inference time comparison in Table 4 with 1024 * 2048 images in the *Cityscapes* dataset. Since the iterative bottom-up and top-down cooperation in multiple steps, the proposed UBCN requires a longer inference time compared to most of the other methods, which can be further considered and ameliorated. Note that the iterative steps and the depth of the network in RCDNet is far more than UBCN, which results in the longest inference time.

5 More Qualitative Experimental Results

5.1 Qualitative Comparisons on Real-world Data

To further illustrate the applicability of the proposed UBCN in real-world rain conditions, we evaluate the performance of UBCN on more self-collect real-world rainy images. We also provide both deraining and segmentation results to demonstrate the effectiveness of UBCN on both real-world image deraining and segmentation tasks. As shown in Fig. 2, we can observe that UBCN is capable of removing most of the rain streaks in the real rainy image, and also can acquire satisfying segmentation prediction as well, which further demonstrates the practicality of UBCN in the real scene such as autonomous driving.

5.2 Qualitative Comparisons on Synthetic Data

We provide more visual comparisons from Fig. 3 to 7. Figure 3 and 4 are the rain removal and semantic segmentation results on Cityscapes dataset. The Fig. 5 to 7 show the deraining and segmentation results on VOC2012 dataset. Specifically, Fig. 3 show the comparison results on Cityscapes under light rain, while Fig. 4 present the results under heavy rain. Most of the comparable methods can handle light rain well. However, when encounter heavy rain, other methods often leave severe residual in the derain image, leading to worse restoration and segmentation performance. Thanks to the unified bidirectional cooperation mechanism, the proposed UBCN not only able to perform adaptive restoration via semantic guidance to acquire better deraining result but also can obtain satisfying segmentation result through joint image- and feature-level deraining. Therefore, the proposed UBCN can acquire better restoration results so as to better semantic segmentation results simultaneously.

5.3 Analysis of Iterative Cooperation

Furthermore, we provide more visual analysis of the deraining and segmentation performance in each iterative step. Figure 8 and 9 show two samples of visual comparison among different steps on *Cityscapes* dataset. We can observe that the performance of both image deraining and segmentation are improved step by step, further revealing the effectiveness of the proposed unified bidirectional cooperation paradigm.

References

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3213–3223.

Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; and Paisley, J. 2017. Removing rain from single images via a deep detail network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3855–3863.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 770–778.

Li, X.; Wu, J.; Lin, Z.; Liu, H.; and Zha, H. 2018. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Eur. Conf. Comput. Vis.*, 254–269.

Ren, D.; Zuo, W.; Hu, Q.; Zhu, P.; and Meng, D. 2019. Progressive image deraining networks: A better and simpler baseline. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3937–3946.

Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Semantic foggy scene understanding with synthetic data. *Int. J. Comput. Vis.*, 126(9): 973–992.

Wang, H.; Xie, Q.; Zhao, Q.; and Meng, D. 2020. A Model-driven Deep Neural Network for Single Image Rain Removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3103–3112.

Yang, W.; Tan, R. T.; Feng, J.; Guo, Z.; Yan, S.; and Liu, J. 2019. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6): 1377–1393.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2881–2890.

Layer	Input	CR	CR	RES	RES	RES	RES	RES	RES
Kernel Size	-	3*3	3*3	3*3	3*3	3*3	3*3	3*3	3*3
Input Filter Number	-	3	32	32	32	32	32	32	32
Output Filter Number	-	32	32	32	32	32	32	32	32
Stride	-	1	1	1	1	1	1	1	1
Padding	-	1	1	1	1	1	1	1	1
Layer	RES	RES	RES	RES	RES	CBR	Conv	Output	
Kernel Size	3*3	3*3	3*3	3*3	3*3	3*3	3*3	-	
Input Filter Number	32	32	32	32	32	32	32	-	
Output Filter Number	32	32	32	32	32	32	3	-	
Stride	1	1	1	1	1	1	1	-	
Padding	1	1	1	1	1	1	1	-	

Table 1: Architecture of image-level derain module. CBR means convolution + batch normalization + ReLU. RES represents conventional residual block proposed in ResNet (He et al. 2016), including two CRP and a skip connection.

Layer	Input	Conv	CBLR	CBLR	CBLR	CBLR	Conv
Kernel Size	-	4*4	4*4	4*4	4*4	4*4	4*4
Input Filter Number	-	2048	64	128	256	512	1
Output Filter Number	-	64	128	256	512	512	1
Stride	-	2	1	1	1	1	1
Padding	-	1	1	1	1	1	1
Negative Slope	-	-	0.2	0.2	0.2	0.2	-

Table 2: Architecture of discriminator in feature-level adaptation module. CBLR means convolution + batch normalization + LeakyReLU. Negative slope represents the LeakyReLU coefficient.

Layer	Input	CR	CR	DOT	RES	RES	RES	RES	RES
Kernel Size	-	3*3	3*3	-	3*3	3*3	3*3	3*3	3*3
Input Filter Number	-	6	32	-	32	32	32	32	32
Output Filter Number	-	32	32	-	32	32	32	32	32
Padding	-	1	1	-	1	1	1	1	1
Stride	-	1	1	-	1	1	1	1	1
Multi-Path	False	False	False	True	True	True	True	True	True
Layer	RES	RES	RES	RES	Add	CBR	Conv	Output	
Kernel Size	3*3	3*3	3*3	3*3	-	3*3	3*3	-	
Input Filter Number	32	32	32	32	-	32	32	-	
Output Filter Number	32	32	32	32	-	32	3	-	
Padding	1	1	1	1	-	1	1	-	
Stride	1	1	1	1	-	1	1	-	
Multi-Path	True	True	True	True	False	False	False	False	

Table 3: Architecture of semantic attentive module. CBR means convolution + batch normalization + ReLU. RES represents conventional residual block proposed in ResNet (He et al. 2016), including two CRP and a skip connection. ADD and DOT represent point-wise addition and dot production. Multi-path represents whether features are handled in multiple paths.

Methods	Parameters (MB)	Running Time (s)
DDN (Fu et al. 2017)	0.23	0.17
RESCAN (Li et al. 2018)	0.60	1.12
PReNet (Ren et al. 2019)	0.67	1.52
JORDER-E (Yang et al. 2019)	16.73	9.12
RCDNet (Wang et al. 2020)	13.11	23.29
UBCN	13.40 (IDM) + 0.4 (SAM) + 177.60 (FAM)	11.78

Table 4: Comparison on parameters and running time. IDM and FAM represents the image-level derain module and the feature-level adaptation module respectively.

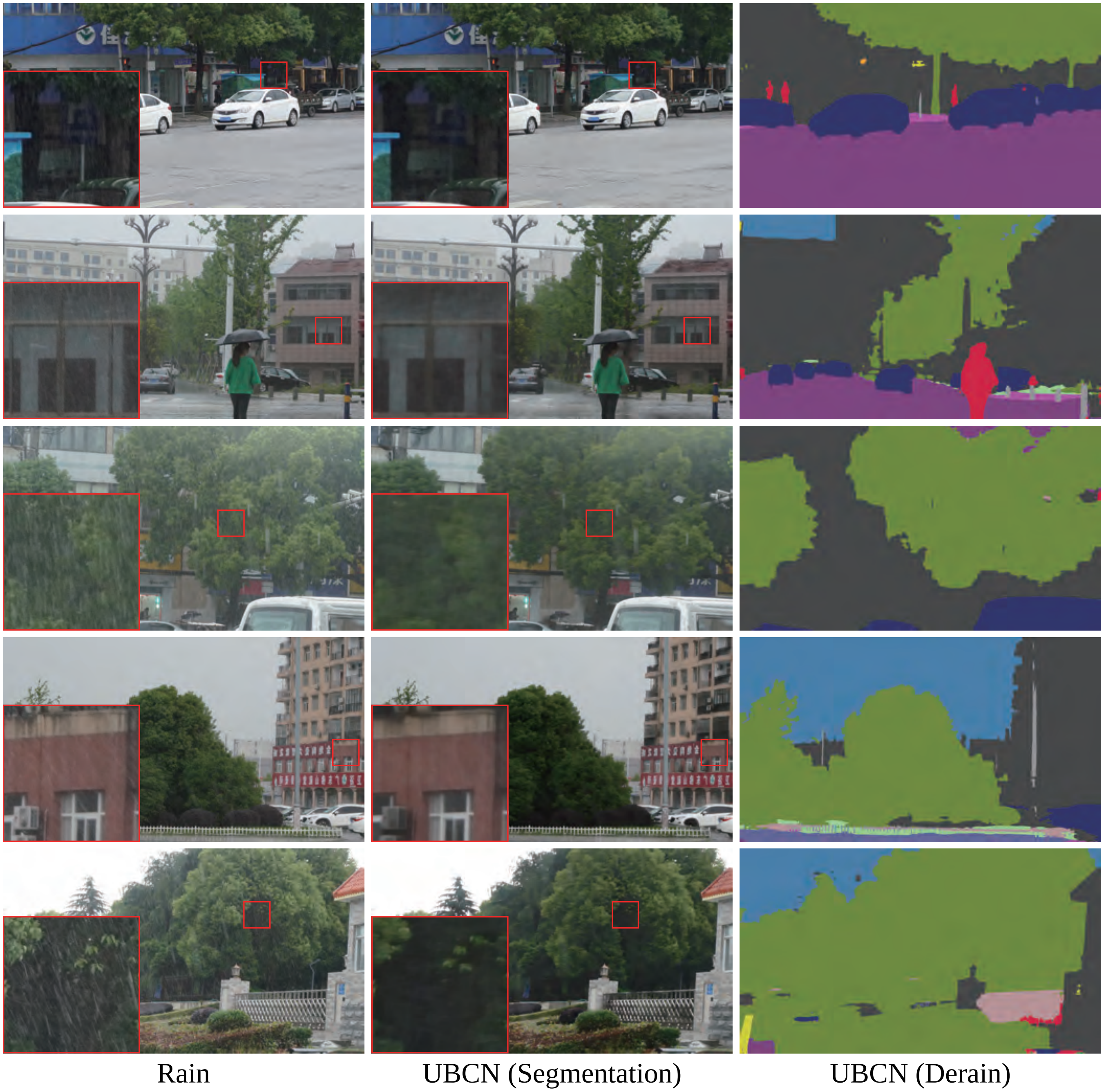


Figure 2: Rain removal and semantic segmentation results on self-collected real-world rainy images. The derain results are better observed by zooming in on screen.

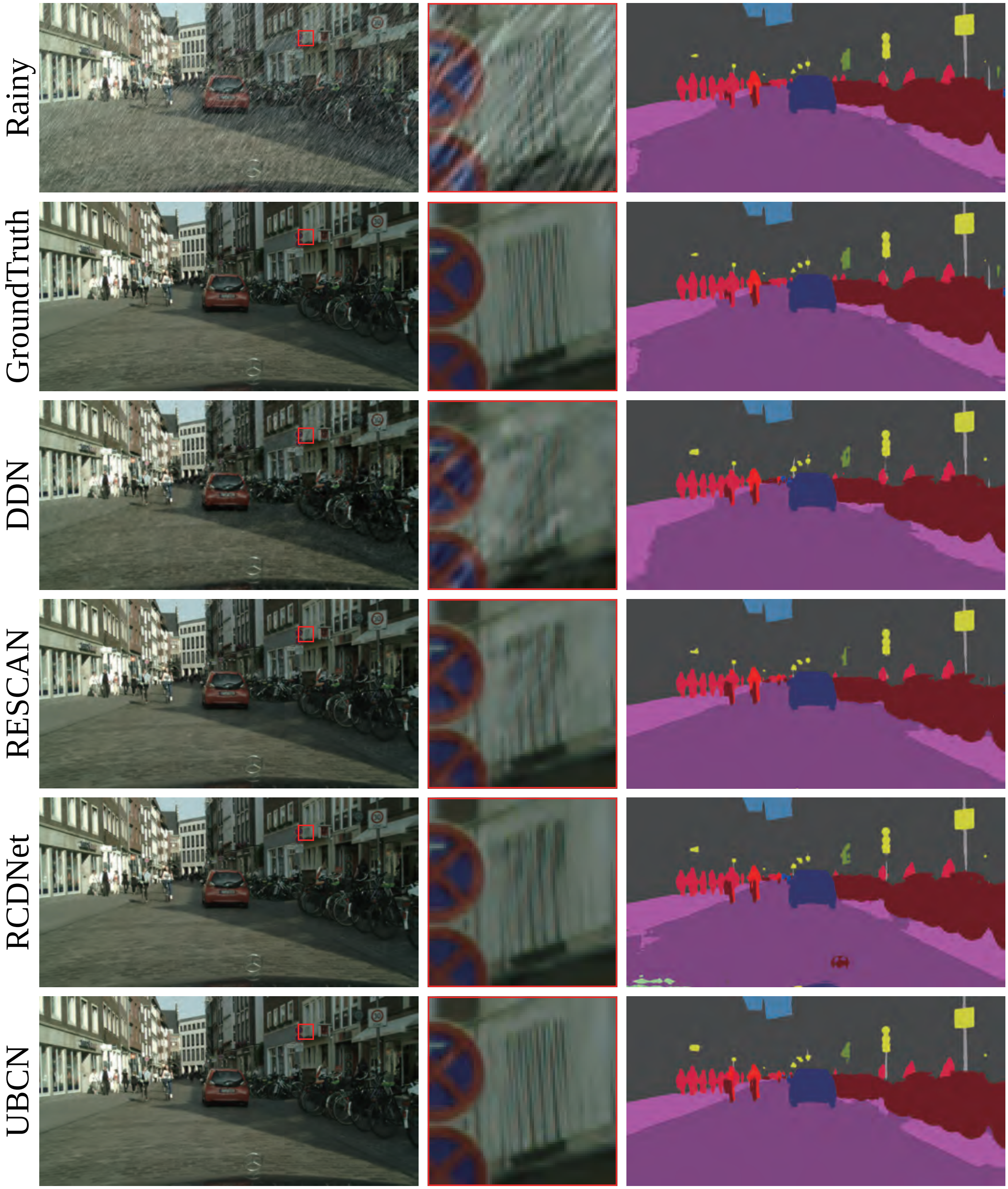


Figure 3: Rain removal and semantic segmentation comparisons under light rain on Cityscapes dataset. The images are better observed by zooming in on screen.

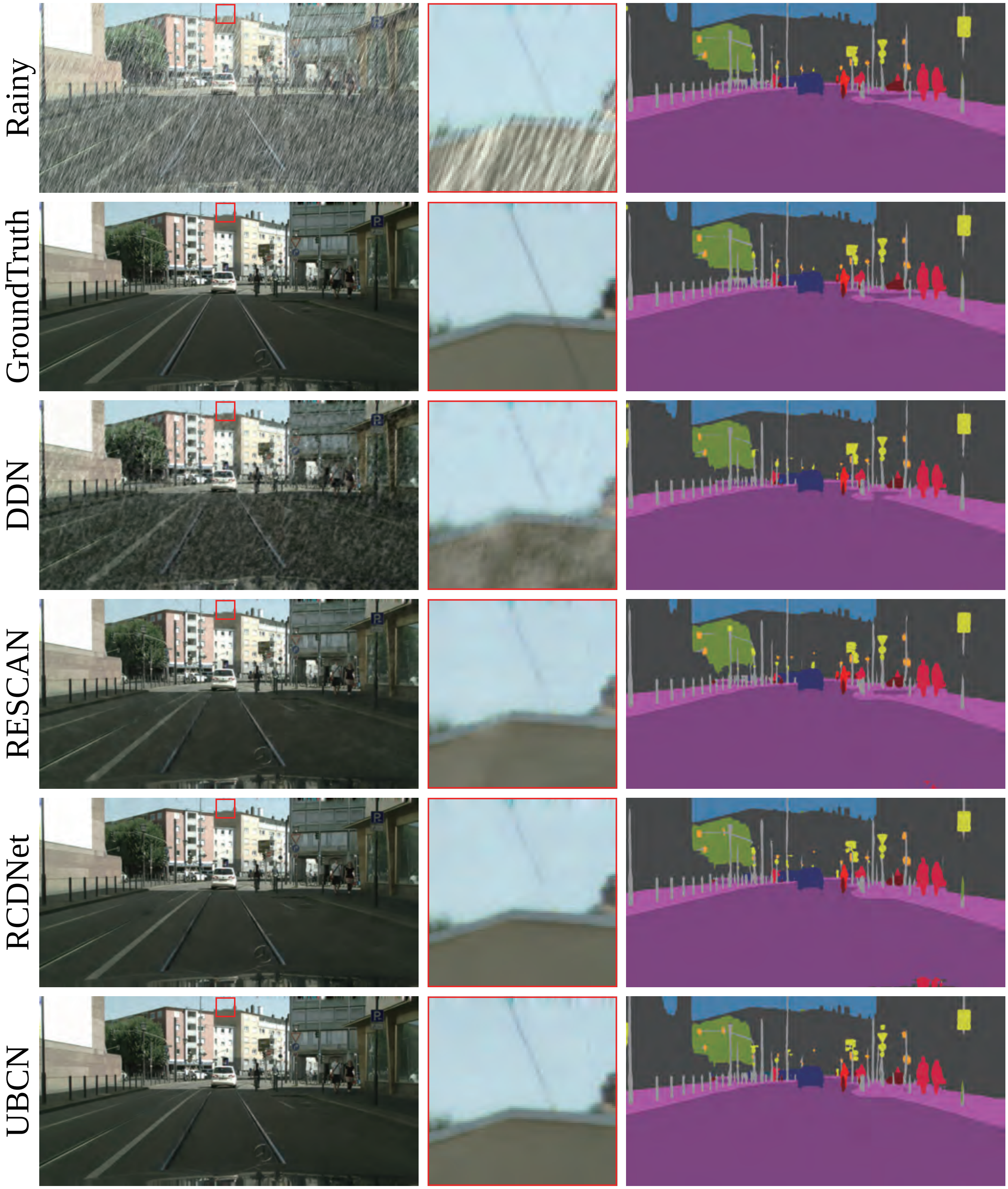


Figure 4: Rain removal and semantic segmentation comparisons under heavy rain on Cityscapes dataset. The images are better observed by zooming in on screen.

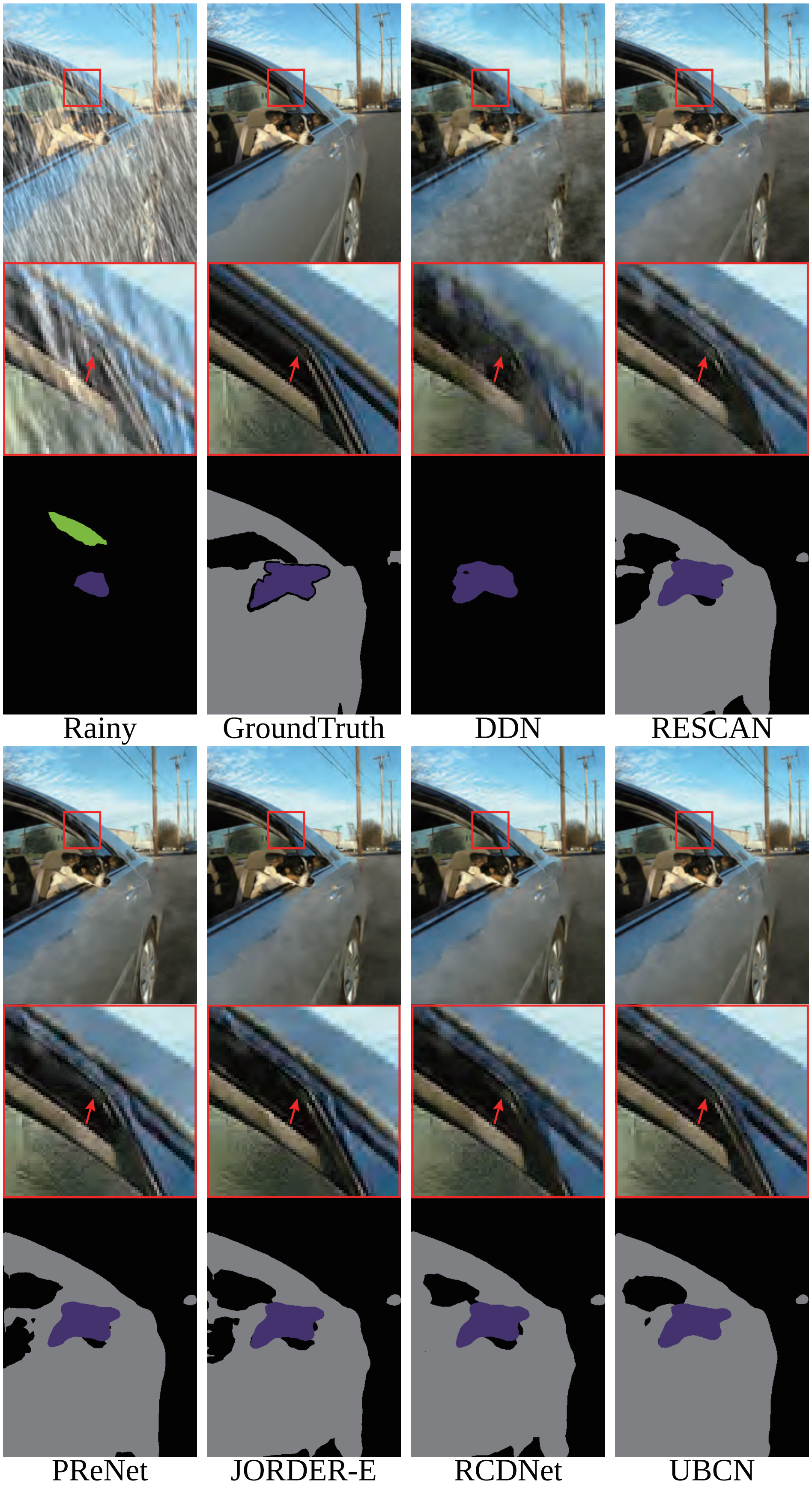


Figure 5: Deraining and semantic segmentation comparisons on VOC2012 dataset. The images are better observed by zooming in on screen.

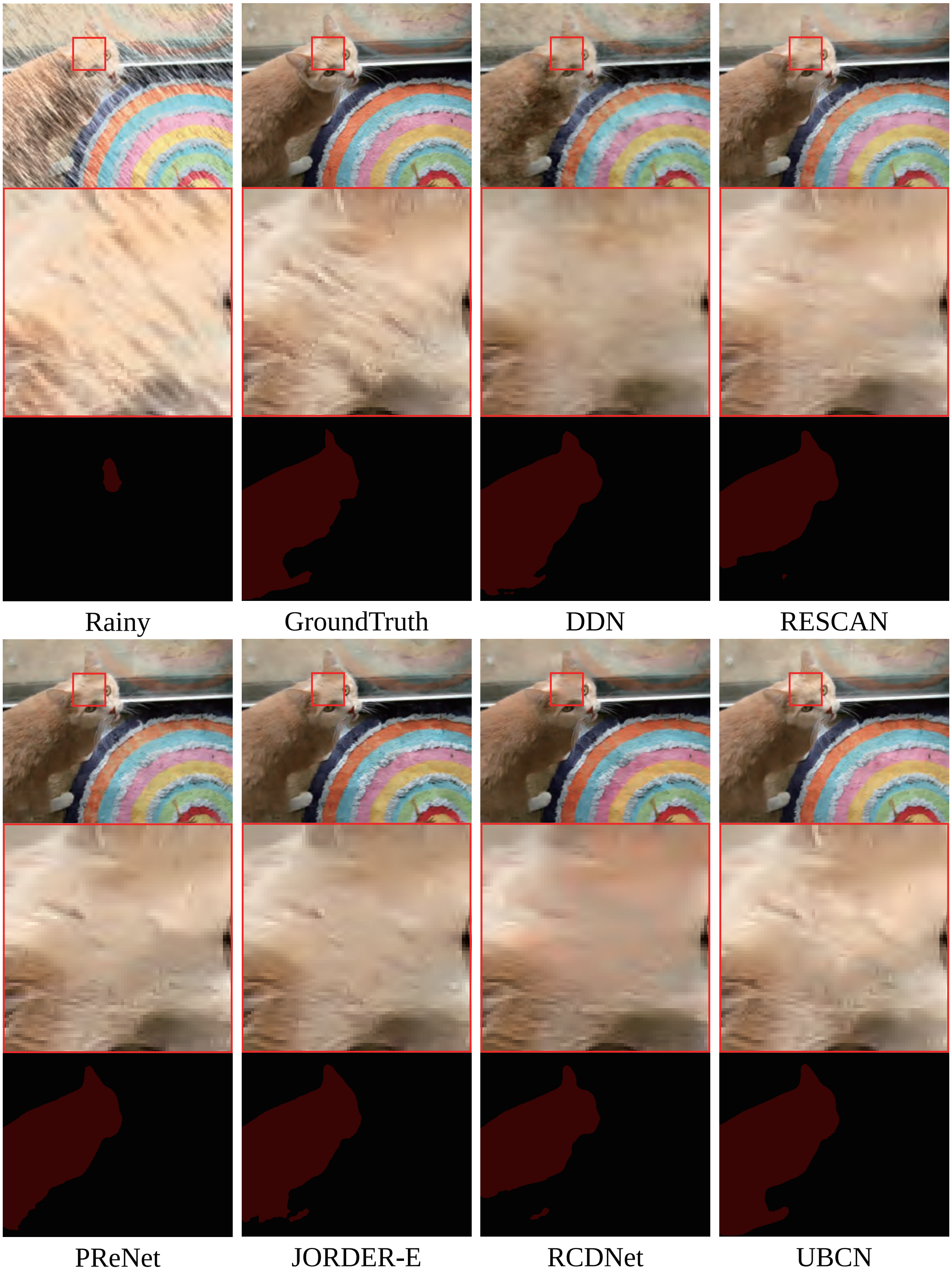


Figure 6: Deraining and semantic segmentation comparisons on VOC2012 dataset. The images are better observed by zooming in on screen.

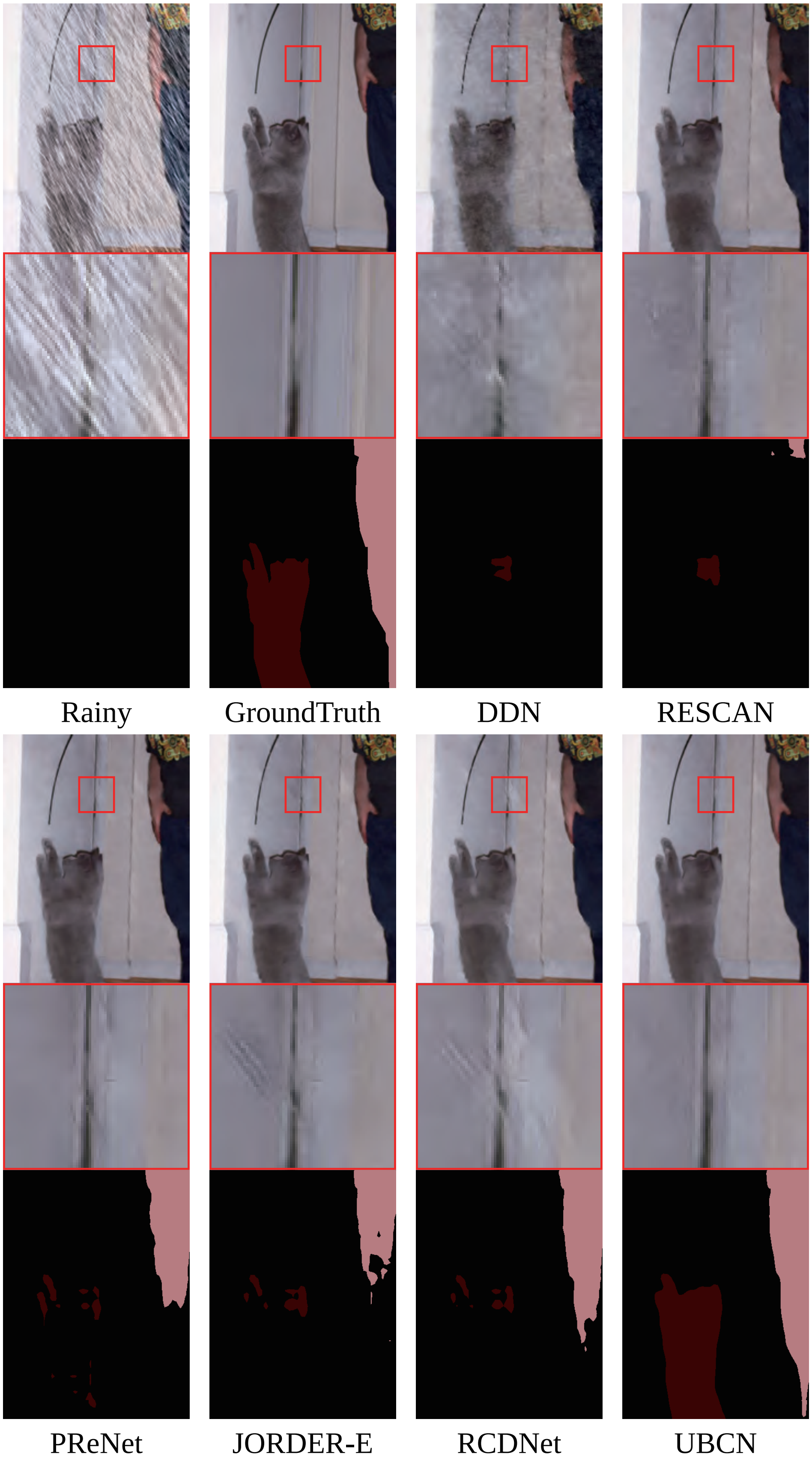


Figure 7: Deraining and semantic segmentation comparisons on VOC2012 dataset. The images are better observed by zooming in on screen.

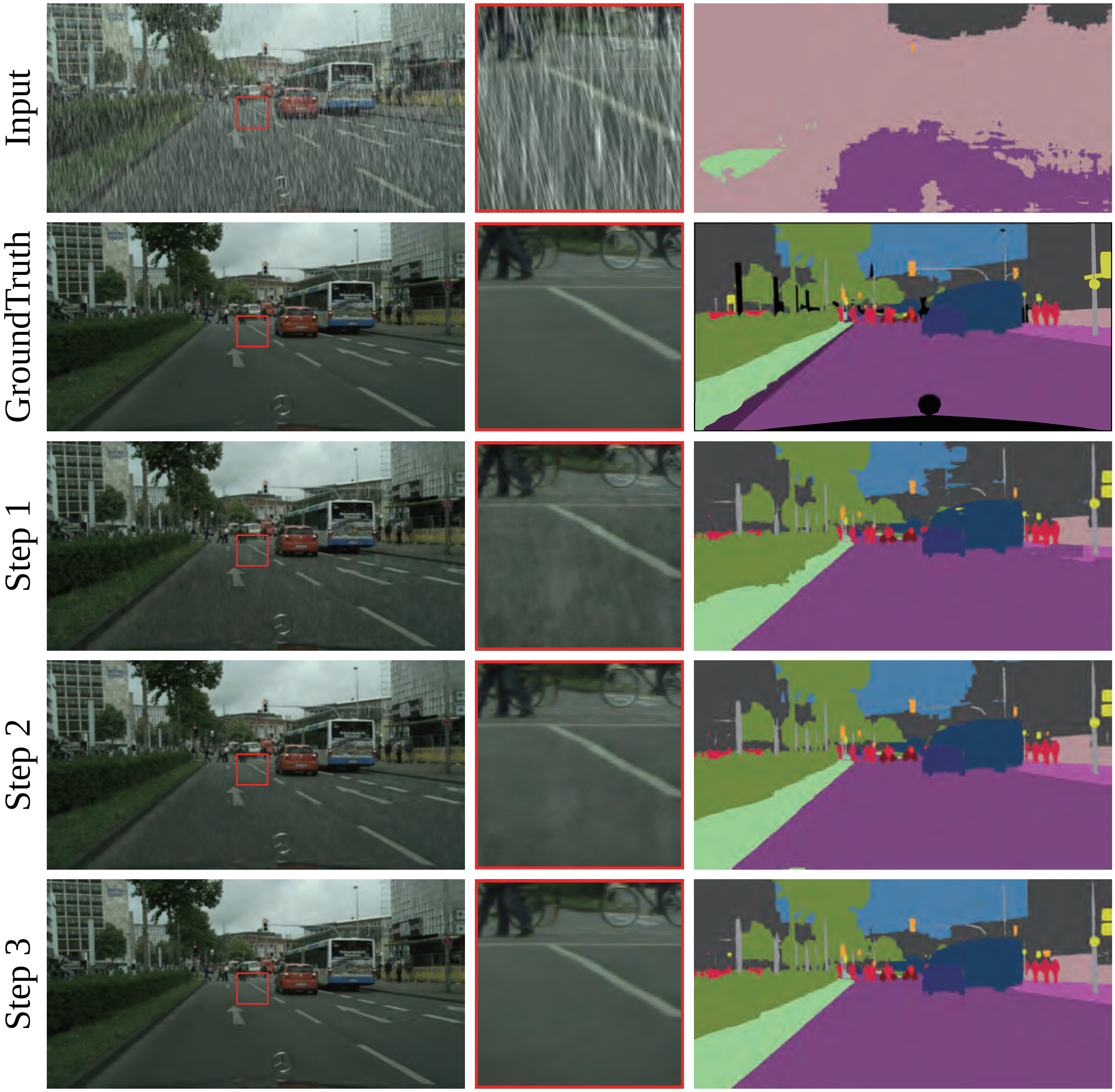


Figure 8: Visual comparison of different steps. With iterative bidirectional cooperation, the deraining and segmentation results are facilitated simultaneously and progressively.

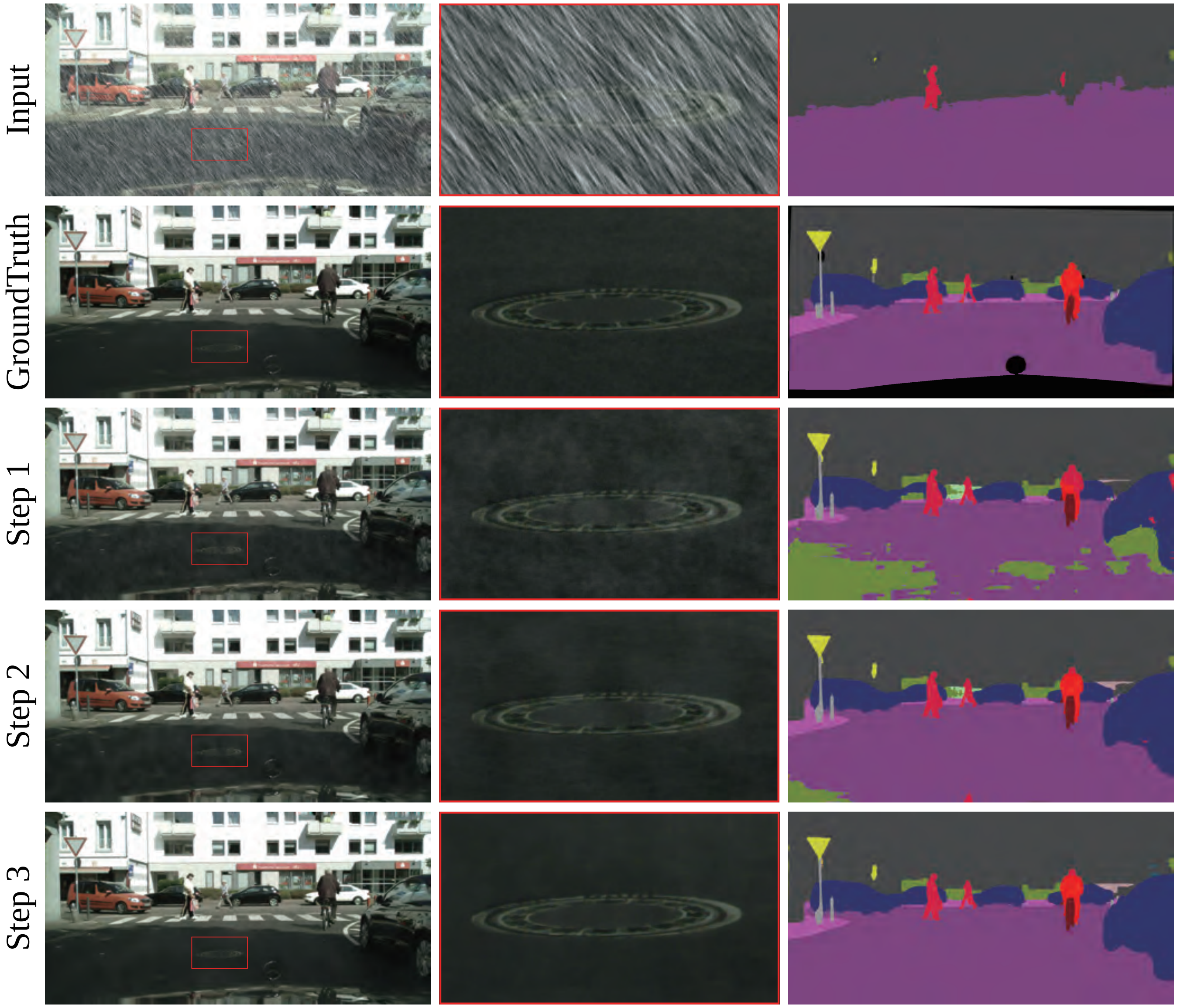


Figure 9: Visual comparison of different steps. With iterative bidirectional cooperation, the deraining and segmentation results are facilitated simultaneously and progressively.