Towards Robust Rain Removal Against Adversarial Attacks: A Comprehensive Benchmark Analysis and Beyond (Supplementary Material)

Yi Yu[1,2] Wenhan Yang[1*] Yap-Peng Tan[1] Alex C. Kot[1]
[1]School of Electrical and Electronic Engineering, Nanyang Technological University
[2]ROSE Lab, Interdisciplinary Graduate Programme, Nanyang Technological University
yuyi0010@e.ntu.edu.sg {wenhan.yang,eyptan,eackot}@ntu.edu.sg

Abstract

In this supplementary material, we provide more details on the module analysis and robust model training described in the main submission. Besides, we show more visualization results against adversarial attacks at different levels of attacks and with various losses/targets. In the end, we utilize more advanced attack scenarios to demonstrate the robustness of our proposed method. Code is available in https://github.com/yuyi-sd/Robust_Rain_Removal.

1. More Implementation Details

1.1. Attack Framework

1) Attack Metrics.

- *Restoration* (*Human Vision*): ℓ_2 Euclidean distance:

$$\delta = \underset{\delta, \|\delta\|_\infty \leq \epsilon}{\arg\max} \|f(X+\delta|\theta) - f(X|\theta)\|_2 . \tag{1}$$

  Note that $f(X|\theta)$ is in the vector form.

- *Downstream CV Tasks* (*Machine Vision*): Learned Perceptual Image Patch Similarity (LPIPS) [14]:

$$\delta = \underset{\delta, \|\delta\|_\infty \leq \epsilon}{\arg\max} \ell_{pips}(f(X+\delta|\theta), f(X|\theta)). \tag{2}$$

  Here, we use the default LPIPS setting (using a pretrained VGG network) to evaluate the distance between images. Higher means the compared images are more different, lower means the compared images are more similar for ℓ_{pips}.

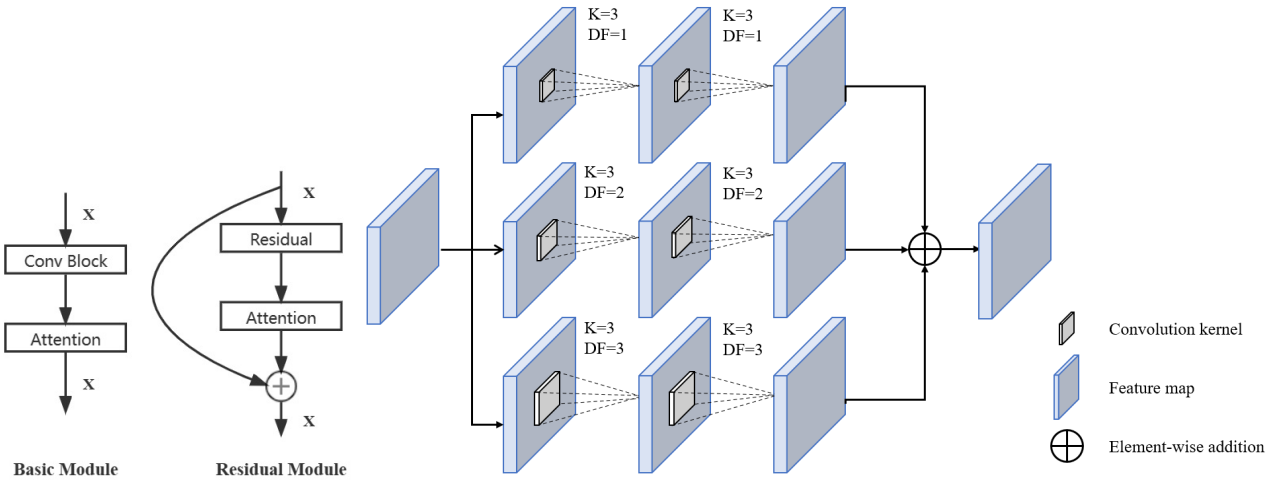
1.2. Evaluation Metrics

1) Task-driven Metrics Calculation. For the task-driven metrics, we use SSeg [15][1] for semantic segmentation and Pedestron [2][2] for pedestrian detection, and both approaches are using the respective pretrained model on Cityscape dataset. Note that when evaluating the deraining outputs by down-stream tasks, we use the corresponding segmentation/detection output of the ground truth no-rain image as the true label.

1.3. Module Analysis

1) Attention Module. Figure 1 (a) illustrates the integrated location of the attention module, and the attention module is integrated at all stages to better augment the performance of the whole architecture.

*Corresponding author.
[1]https://github.com/YeLyuUT/SSeg
[2]https://github.com/hasanirtiza/Pedestron



(a) Attention Module integration. (b) Convolution Block with diverse dilation factors.

Figure 1. Module.

2) Receptive Field. The basic block with a multi-dilated convolution structure is shown in Figure 1 (b).
3) Adversarial Loss. For the rain removal model, the adversarial loss can be obtained as:

$$L_{\text{adv}} = \lambda \max_{\delta, \|\delta\|_\infty \leq \epsilon} \|f(X+\delta|\theta) - f(X|\theta)\|_2 . \tag{3}$$

To optimize the inner-max function, we use PGD[6] with iteration $T = 5$ and $\epsilon = 4/255$ to obtain the perturbations $\delta_\epsilon(X)$. Thus, the adversarial loss can be formulated as:

$$L_{\text{adv}} = \lambda \|f(X+\delta_\epsilon(X)|\theta) - f(X|\theta)\|_2 . \tag{4}$$

Note that $f(X|\theta)$ is in the vector form. As the inner-max optimization costs a lot of time during training stages, we first train the model with fidelity loss, and then finetune the model with the adversarial loss for a few epochs.

Because large λ will compromise the performance for input without perturbations, we adjust the weight to make the final clean performance around $28.0(\pm 0.2)$ for Rain100H and $28.5(\pm 0.2)$ for RainCityscape in terms of PSNR.

1.4. Ensemble Modules into Robust Model

1) Network Design. We change the attention module in MPRNet (SE with multiplication) into SE with addition, set the number of recurrent blocks as 3, and incorporate convolution path with diverse dilations $\{1, 2, 3\}$.
2) Network Training. We first train the model with Charbonnier Loss [4] as fidelity loss:

$$L_{\text{fid}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{3}\sum_{j=1}^{3}\rho(Y^{(i)} - f_j(X^{(i)}|\theta)), \tag{5}$$

where $\rho(x) = \sum_{l=1}^{L}\sqrt{x_l^2 + \epsilon^2}$ (a differentiable variant of ℓ_1 norm, $\epsilon = 10^{-3}$), $f_j(X^{(i)}|\theta))$ is the output at the j-th recurrent block, $X^{(i)}$ is the input rainy image, and $Y^{(i)}$ is the groundtruth image. We then finetune the model with joint loss:

$$L_{\text{joint}} = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{1}{3}\sum_{j=1}^{3}\rho(Y^{(i)} - f_j(X^{(i)}|\theta)) + \lambda\left\|f_3(X^{(i)} + \delta_\epsilon(X^{(i)})|\theta) - f_3(X^{(i)}|\theta)\right\|_2\right\}, \tag{6}$$

We use PyTorch [7] to implement our model based on a NVIDIA GeForce RTX 3090 GPU. We use Adam optimizer [3] with batch size 8 for training with fidelity loss and batch size 4 for finetuning with joint loss, and the patch size is 160×160. For training with fidelity loss, the initial learning rate is 2×10^{-4} and divided by 5 every 30 epochs, and the total epochs are 120. For finetuning with joint loss, the initial learning rate is 4×10^{-5} and divided by 5 every 10 epochs, and the total epochs is 30. For both datasets, the training parameters are the same.

2. More Experimental Results

In this section, we demonstrate more experimental results on the test data of Rain100H [11], RainCityscape [1], and some real images [10]. The comparison methods include: JORDER-E [11][3], RCDNet [9][4], MPRNet [13][5], PReNet [8][6], UMRL [12][7], and RESCAN [5][8].

2.1. More Results on Rain100H

Rain100H. Rain100H dataset covers five types of rain streak directions and consists of 1800 paired rain/non-rain images of size 480×320 for training and 100 paired images of the same size for testing. PSNR and SSIM of the input testing set are 12.05 and 0.3623.

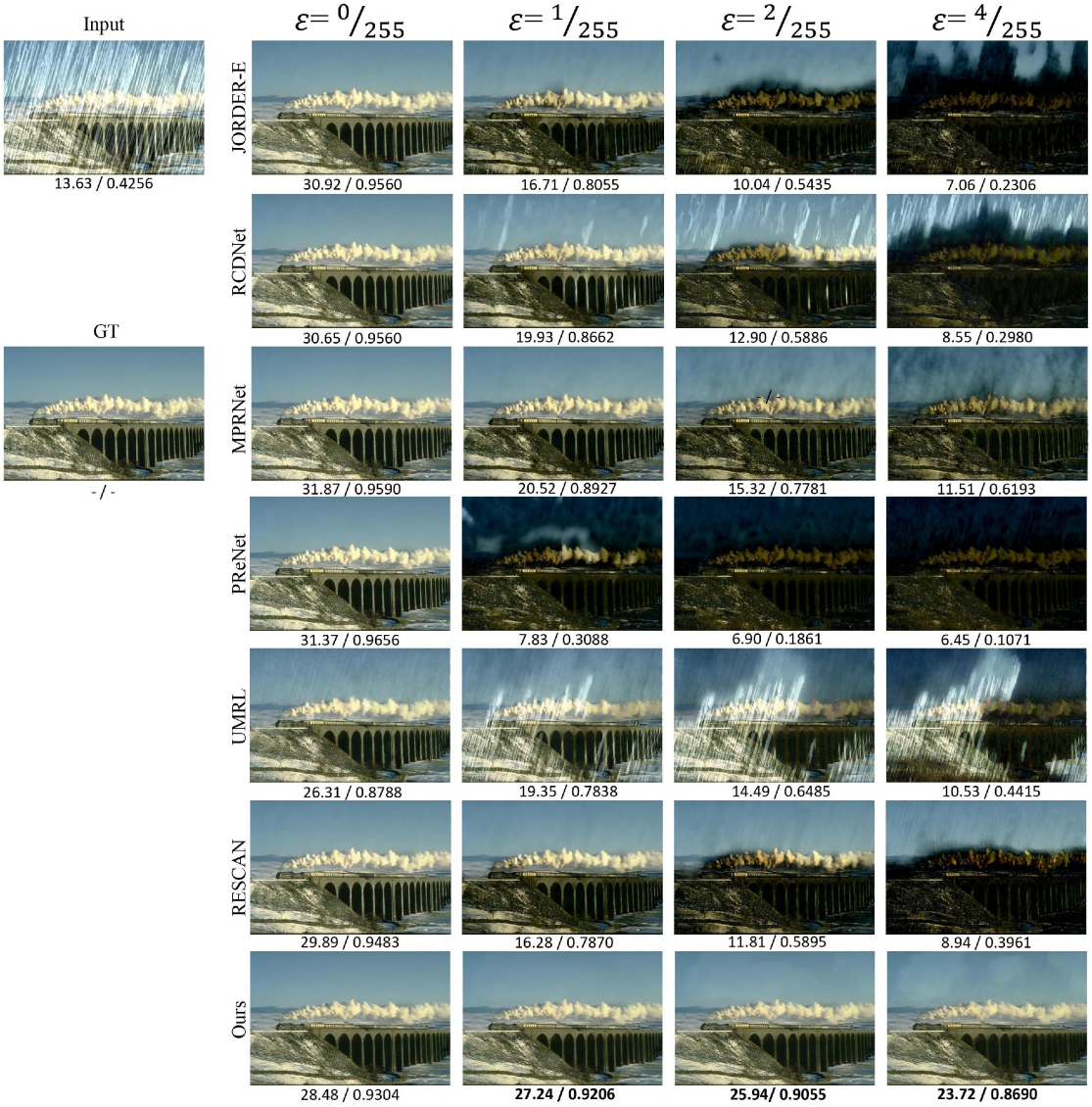


Figure 2. Visual comparison of the deraining outputs of Rain100H [11] test data at various perturbation levels based on **LMSE** attack. Note that $\epsilon = 0/255$ denotes the performance of output to input without perturbations. PSNR/SSIM is listed behind each result. Best view by zooming in.

[3] https://github.com/flyywh/JORDER-E-Deep-Image-Deraining-TPAMI-2019-Journal
[4] https://github.com/hongwang01/RCDNet
[5] https://github.com/swz30/MPRNet
[6] https://github.com/csdwren/PReNet
[7] https://github.com/rajeevyasarla/UMRL--using-Cycle-Spinning
[8] https://github.com/XiaLiPKU/RESCAN

Figure 2 illustrates the deraining outputs of all six comparison methods and our method against LMSE attack with perturbation bound $\epsilon = \{0/255, 1/255, 2/255, 4/255\}$ on a rainy image from Rain100H. As displayed, our proposed method can well handle adversarial attacks with perturbation bound $\epsilon = \{1/255, 2/255\}$, and achieves almost equivalent performance on the input without perturbations compared with other methods. When the perturbation bound $\epsilon = 4/255$, the corruption becomes obvious in the smooth regions, *e.g.* sky.

Figure 3 shows the deraining outputs against LPIPS attack with perturbation bound $\epsilon = 2/255$. Figure 4 and Figure 5 depict the rain removal results against LMSE attack with perturbation bound $\epsilon = 2/255$. From these results, we can easily conclude that our proposed method is quite robust to the perturbed input that aims to degrade the output image.

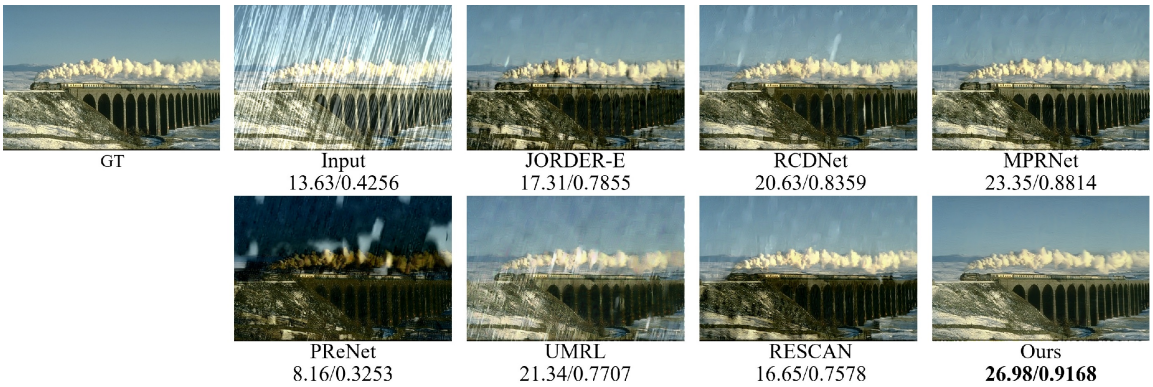


Figure 3. Visual comparison of the deraining outputs of Rain100H [11] with perturbation bound $\epsilon = 2/255$ based on **LPIPS** attack. PSNR/SSIM is listed behind each result. Best view by zooming in.

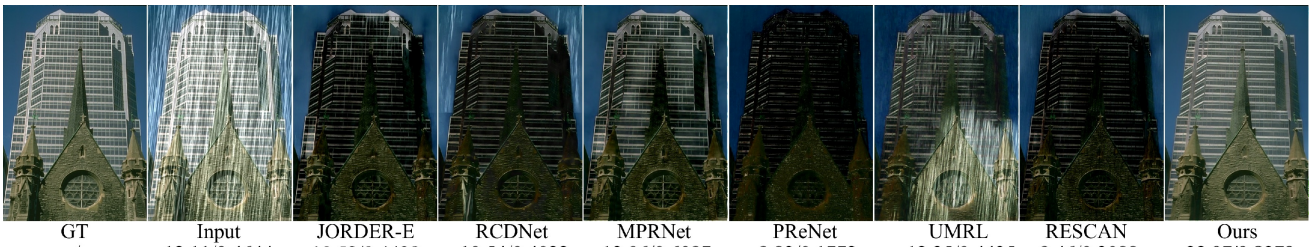


Figure 4. Visual comparison of the deraining outputs of Rain100H [11] with perturbation bound $\epsilon = 2/255$ based on **LMSE** attack. PSNR/SSIM is listed behind each result. Best view by zooming in.

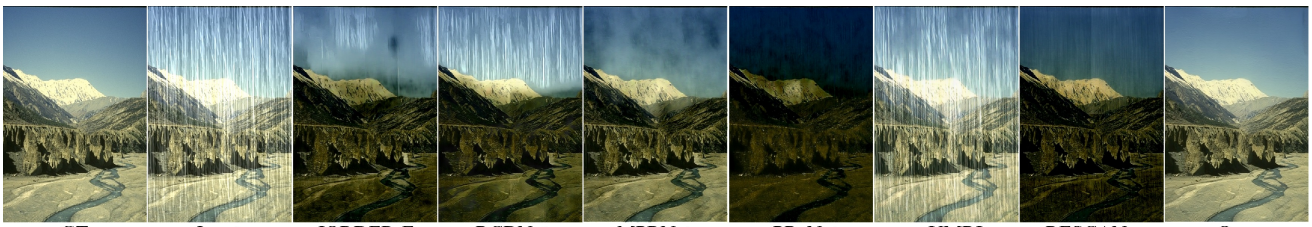


Figure 5. Visual comparison of the deraining outputs of Rain100H [11] with perturbation bound $\epsilon = 2/255$ based on **LMSE** attack. PSNR/SSIM is listed behind each result. Best view by zooming in.

2.2. More Results on RainCityscape

RainCityscape. Halder *et al.* [1] present a physically-based rain rendering pipeline for adding rain on clear weather images in a realistic way, and augment Cityscapes dataset. The augmented dataset consists subsets of four rain intensities $\{25mm, 50mm, 100mm, 200mm\}$, and each subset has 2975 paired images of size 1024×512. We select the subset of rain intensities 100mm, resize the images to 512×256 for efficient computing, and randomly split the dataset into 2875 paired images for training and 100 paired images for testing. PSNR and SSIM of the input testing set are 19.71 and 0.7087. With a well-trained semantic segmentation approach SSeg [15], we can further illustrate the deraining outputs by segmentation overlaps.

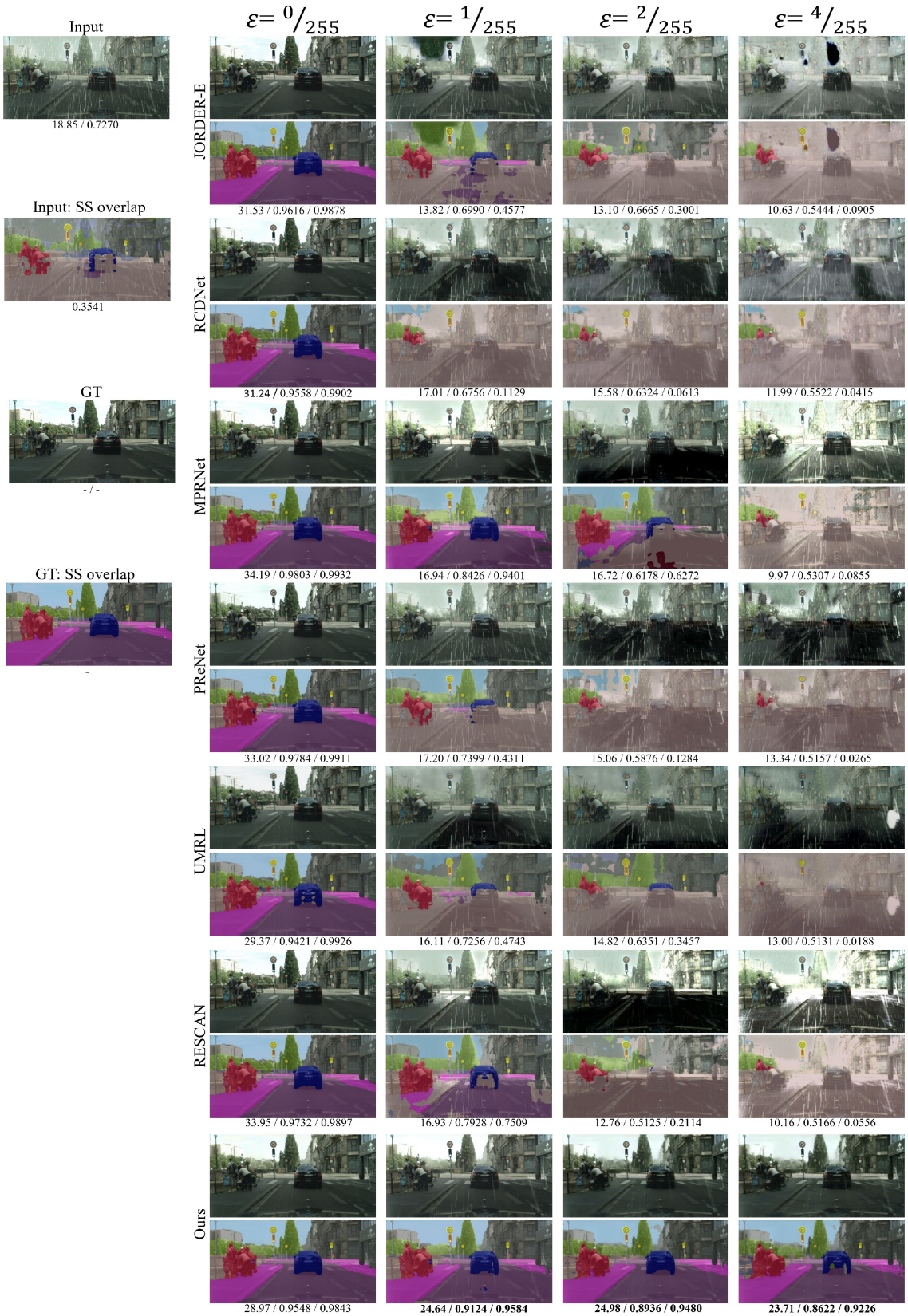


Figure 6. Visual comparison of the deraining outputs of RainCityscape [1] test data at various adversarial perturbation levels based on **LMSE** attack. Note that $\epsilon = 0/255$ denotes the performance of output to input without perturbations. PSNR/SSIM/mIoU is listed behind each result. Best view by zooming in.

From Figure 6, we can observe that comparison methods can't survive the adversarial attacks, and the mIoU of deraining outputs for most methods is under 0.3 with perturbation $\epsilon = 2/255$, which can affect autonomous driving much. In contrast, our proposed method is quite robust to LMSE attacks in terms of PSNR, SSIM and down-stream task evaluation.

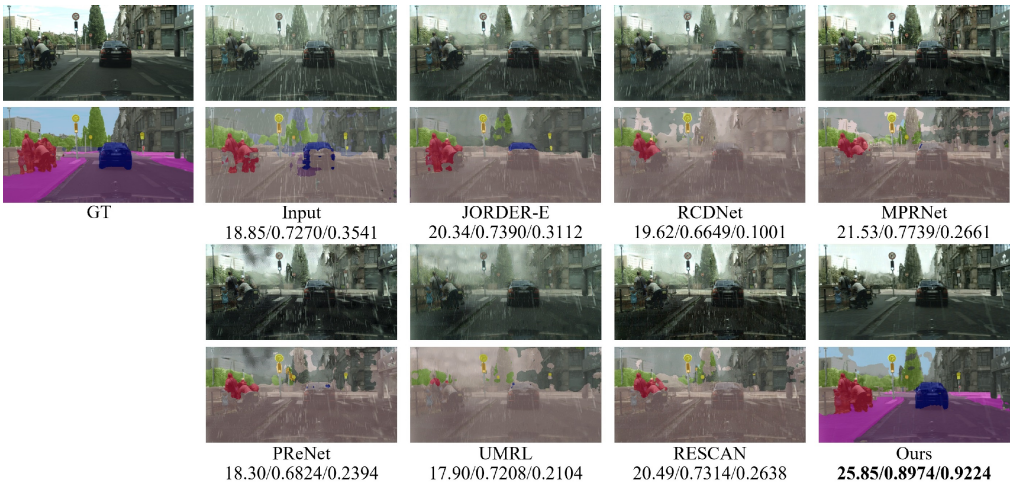


Figure 7. Visual comparison of the deraining outputs of RainCityscape [1] with perturbation bound $\epsilon = 2/255$ based on **LPIPS** attack. PSNR/SSIM/mIoU is listed behind each result.

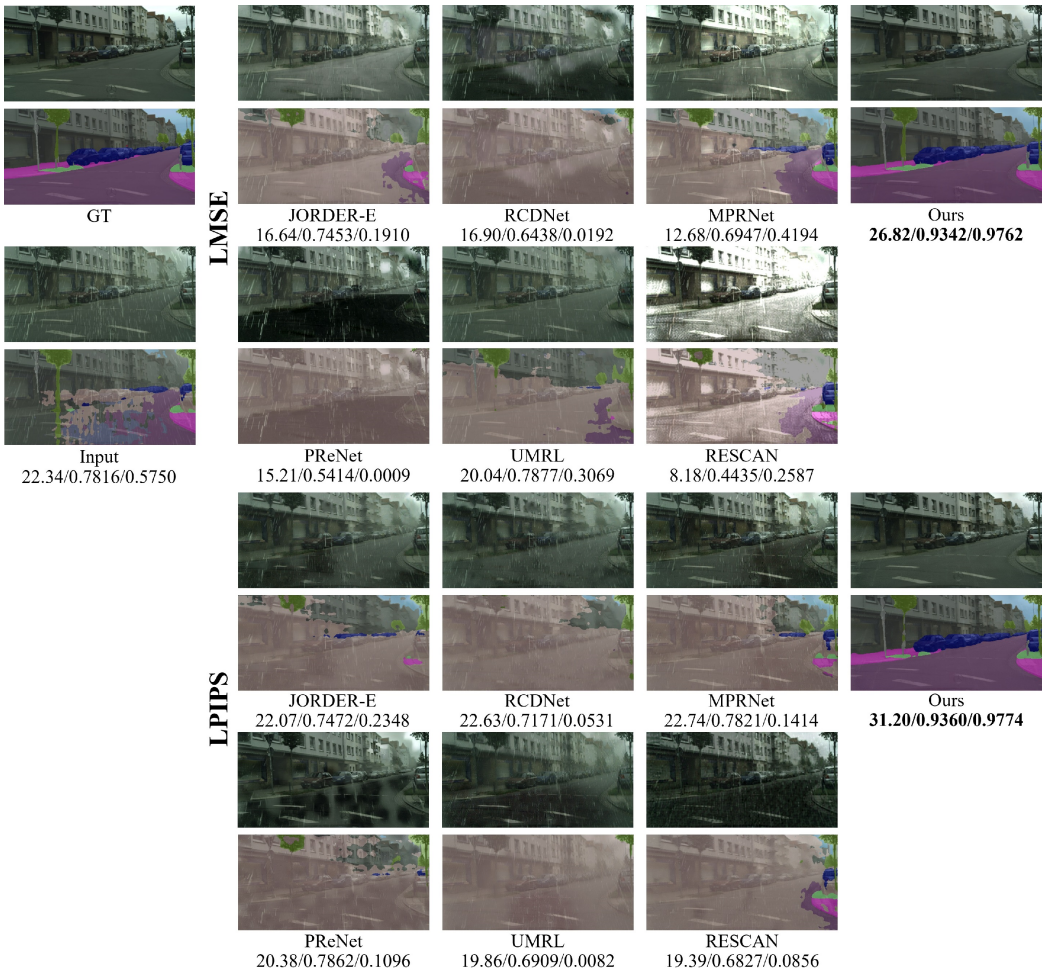


Figure 8. Visual comparison of the deraining outputs of RainCityscape [1] with perturbation bound $\epsilon = 2/255$ based on **LMSE** attack and **LPIPS**. Figures in the first two rows denote the results against LMSE attack, and figures in the first two rows denote the results against LPIPS attack. PSNR/SSIM/mIoU is listed behind each result.

Figure 7 show the results against LPIPS attack, and Figure 8 and Figure 9 offer some other examples with both LMSE and LPIPS attack. From these figures, we can observe that LMSE attack brings a larger performance drop on PSNR and SSIM, while LPIPS attack brings a larger drop on mIoU except for PReNet. It is not surprising as LPIPS attack aims to deviate the feature distance of pretrained VGG network rather than the ℓ_2 Euclidean distance.

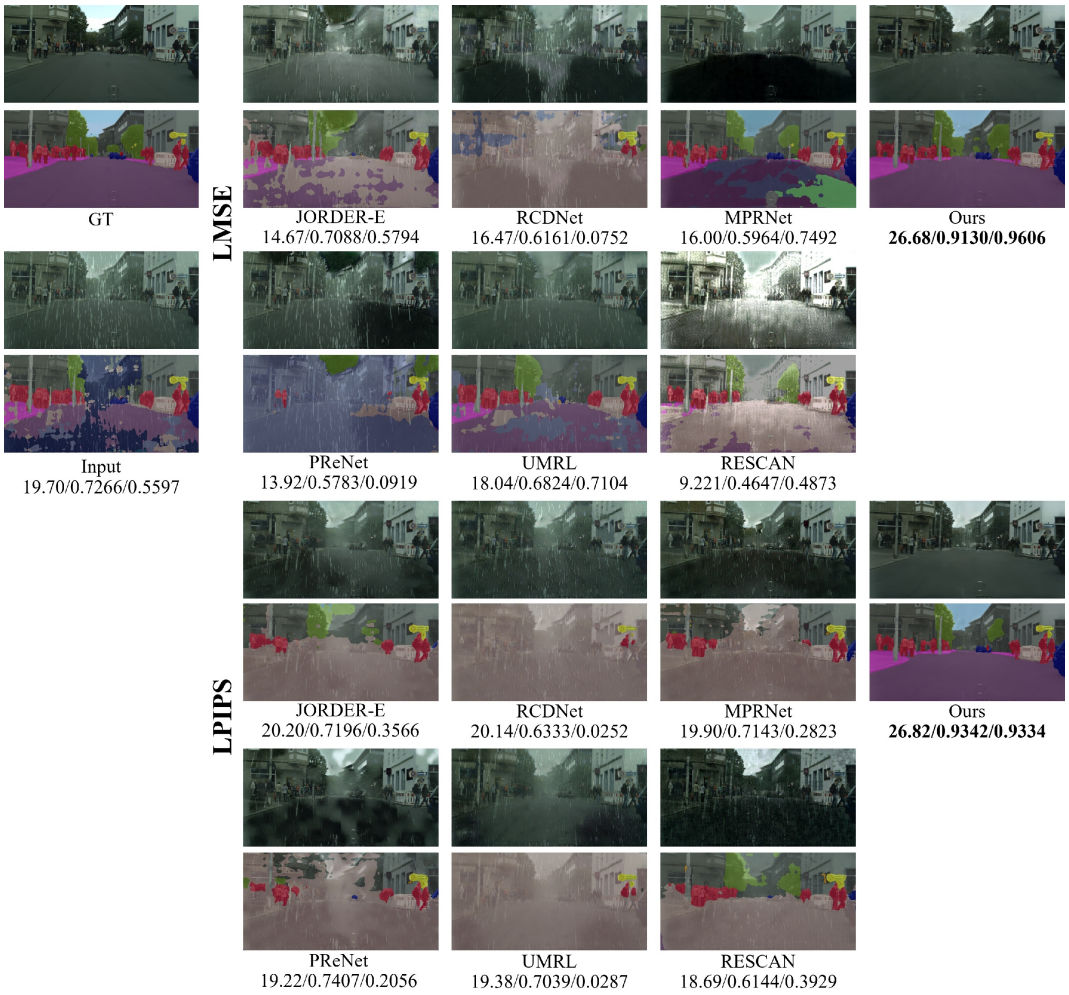


Figure 9. Visual comparison of the deraining outputs of RainCityscape [1] with perturbation bound $\epsilon = 2/255$ based on **LMSE** attack and **LPIPS**. Figures in the first two rows denote the results against LMSE attack, and figures in the first two rows denote the results against LPIPS attack. PSNR/SSIM/mIoU is listed behind each result.

2.3. More Results on Real Images

Real Images. We then analyze the performance of all competed methods on the Internet-Data [10] including 147 rainy images without ground truth. The Internet-Data is collected from the Internet and includes many hard samples with complicated rain streaks. Note that all competing methods evaluated are trained on Rain100H [11].

From Figure 10, our proposed method produces almost the same results for input with perturbation bound ranging from $0/255$ to $4/255$, while deraining outputs of compared methods are degraded heavily even with perturbation bound $\epsilon = 1/255$. Figure 11 shows the results against LPIPS attack, and Figure 12 and Figure 13 provide other examples.

Figure 10. Visual comparison of the deraining outputs of real images [10] at various adversarial perturbation levels based on **LMSE** attack. Note that $\epsilon = 0/255$ denotes the performance of output to input without perturbations.

Although the Internet-Data does not offer ground truth, we can still quantitatively measure the adversarial robustness by calculating the mean-Adversarial-Performance between the origin output and perturbed output:

$$mAP_D = \frac{1}{n(E)} \sum_{\epsilon \in E} \frac{1}{n(\mathcal{D}_t)} \sum_{X \in \mathcal{D}_t} P(f(X), f(X + \delta_{D,\epsilon}(X))), \tag{7}$$

where $E = \{1/255, 2/255, 4/255, 8/255\}$ is the set of ϵ to be evaluated, $\mathcal{D}_t$ is the Internet-Data, $n(\cdot)$ counts the numbers, and $\delta_{D,\epsilon}(X)$ is obtained perturbations by adversarial attacks with attack metric D. P is performance measurement in terms of PSNR and SSIM. Table 1 compares the adversarial robustness on the Internet-Data of all competing methods quantitatively. It is easy to see that our proposed method is not only robust on the corresponding test set of Rain100H, but is also resistant to adversarial perturbation on the cross-domain dataset.

Table 1. Adversarial Robustness (mean-Adversarial-Performance) of competing deraining methods on Internet-Data.

Methods	JORDER-E	RCDNet	MPRNet	PReNet	UMRL	RESCAN	Ours
Restoration (Human Vision): MSE Loss							
PSNR	9.18	12.85	13.53	8.46	18.05	14.05	**29.88**
SSIM	0.2136	0.4232	0.5556	0.1332	0.6595	0.4314	**0.8780**
Downstream CV Tasks (Machine Vision): LPIPS Loss							
PSNR	14.19	20.19	22.04	9.38	27.22	22.22	**37.59**
SSIM	0.4971	0.6968	0.7823	0.2644	0.8232	0.6939	**0.9270**

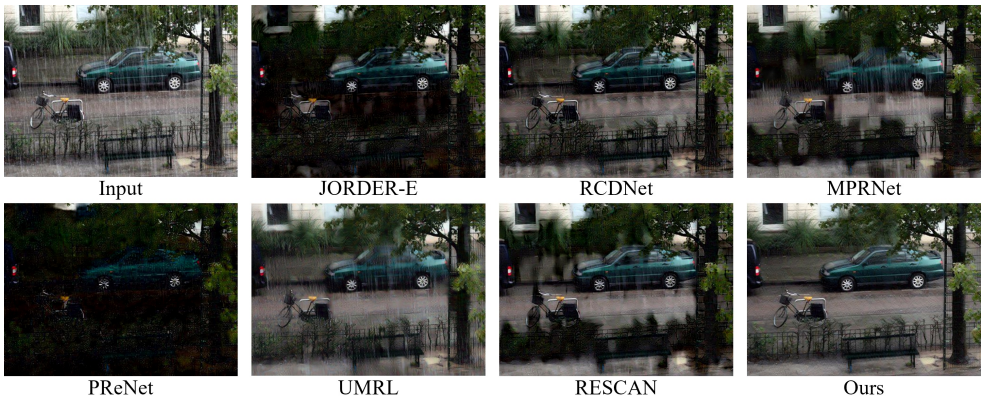


Figure 11. Visual comparison of the deraining outputs of real images [10] with perturbation bound $\epsilon = 2/255$ based on **LPIPS** attack.

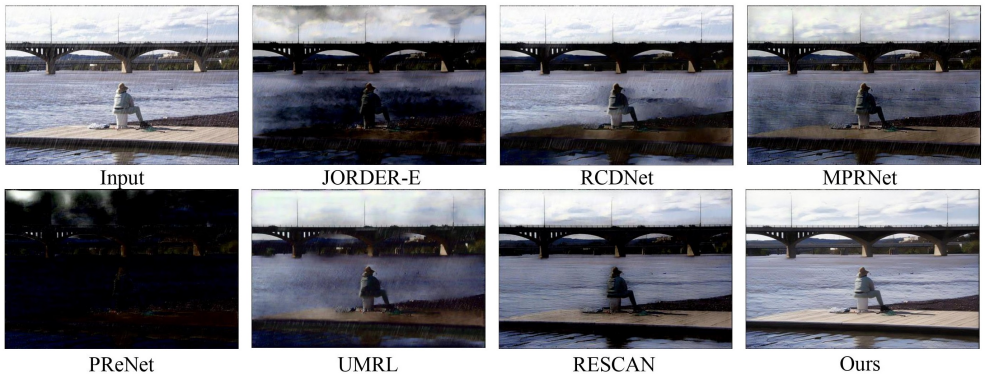


Figure 12. Visual comparison of the deraining outputs of real images [10] with perturbation bound $\epsilon = 2/255$ based on **LMSE** attack.

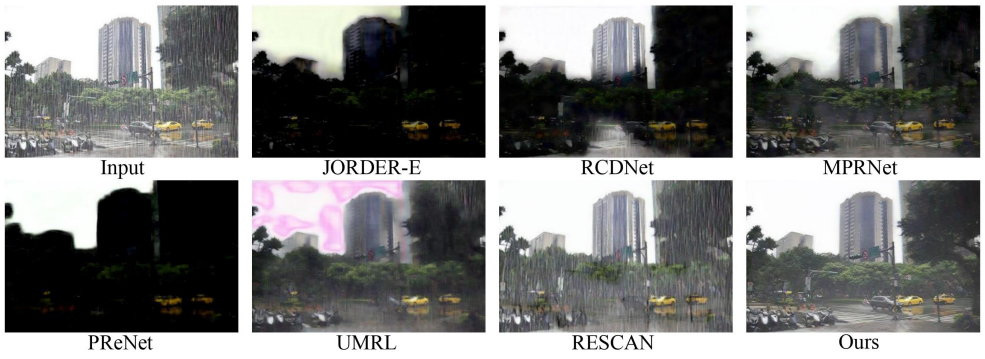


Figure 13. Visual comparison of the deraining outputs of real images [10] with perturbation bound $\epsilon = 2/255$ based on **LMSE** attack.

2.4. More Results of Advanced Attack Scenarios

1) Object-sensitive Attack. Figure 14 illustrate the object-sensitive attack on PReNet [8] and our proposed method with several rainy images from RainCityscape [1]. As the attack only leverages the perceptual loss [14] and has no prior information about the down-stream model, we select the case with a high success rate, such as: 1) Car to road; 2) Person to road; 3) Car to person. We can observe that our proposed method can handle this attack.

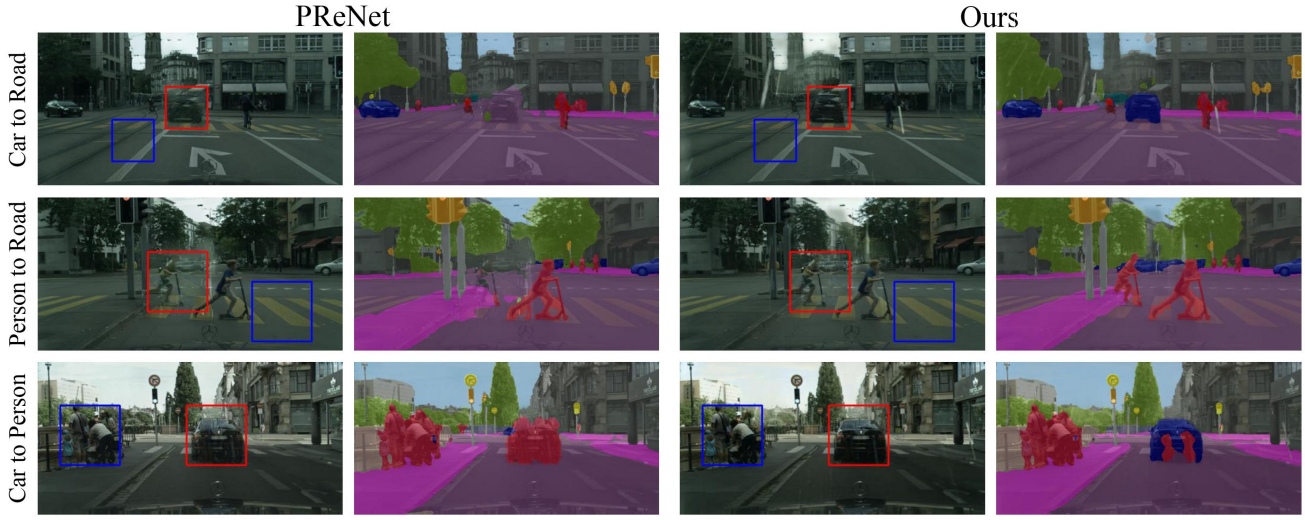


Figure 14. Deraining Images of **PReNet** and **Ours** against Object-sensitive Attack and its corresponding semantic segmentation overlap in RainCityscape [1]. Note that the adversarial perturbation bound $\epsilon = 4/255$. The patches with red boxes are the source objects to attack, and patches with blue boxes are the target objects.

2) Partial Attack (Rain Region Attack). We also consider the case that perturbations are added to the rain region, which can be further undetectable, and the rain region is estimated by thresholding on the difference of the input rainy image and the original output. As shown in Figure 15, our method is much more robust than MPRNet under such an attack.

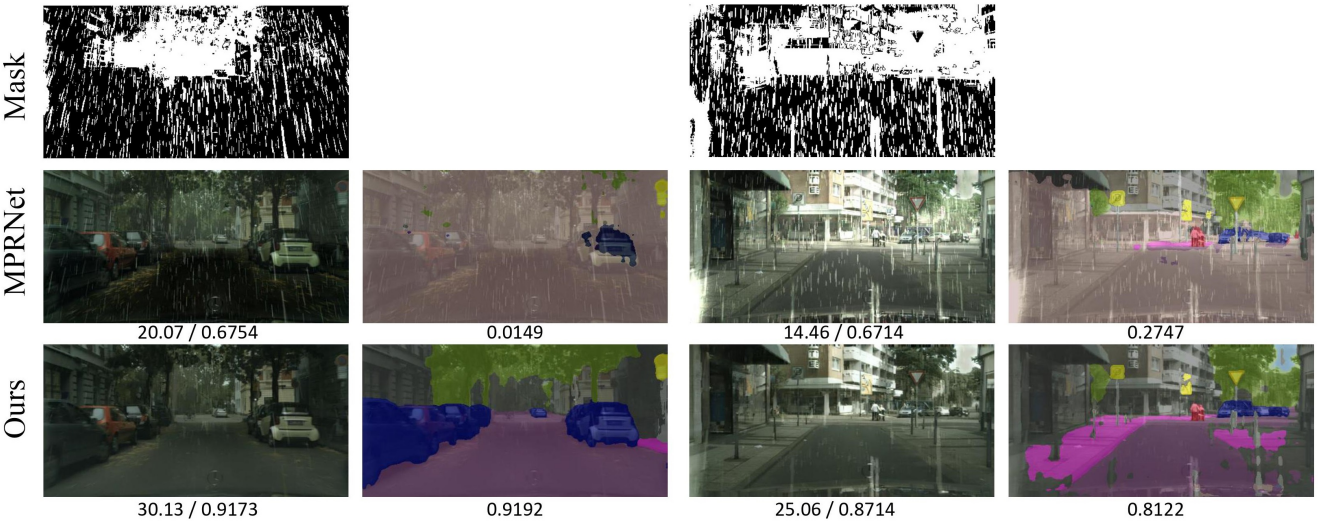


Figure 15. Deraining Images of **MPRNet** and **Ours** against Rain Region Attack (based on LMSE attack) and its corresponding semantic segmentation overlap in RainCityscape [1]. Note that the adversarial perturbation bound $\epsilon = 4/255$. PSNR/SSIM/mIoU is listed behind each result.

3) Unnoticeable Attack for Down-stream Tasks. Figure 15 shows the Unnoticeable Attack for Down-stream Tasks. We can easily observe that all outputs against this attack have quite similar quality with the ground truth in terms of PSNR and SSIM, while the down-stream segmentation performance of MPRNet degrades heavily. Our method is much more robust against this attack.

Figure 16. Deraining Images of **MPRNet** and **Ours** against Unnoticeable Attack for Down-stream Tasks and its corresponding semantic segmentation overlap in RainCityscape [1]. Note that the adversarial perturbation bound $\epsilon = 4/255$. PSNR/SSIM/mIoU is listed behind each result.

4) Input-close Attack. We also provide several examples against Input-close Attack in Figure 17. It is not surprising to find that the attacked outputs of MPRNet are quite close to the input rainy images, while our method is indeed robust to this attack.

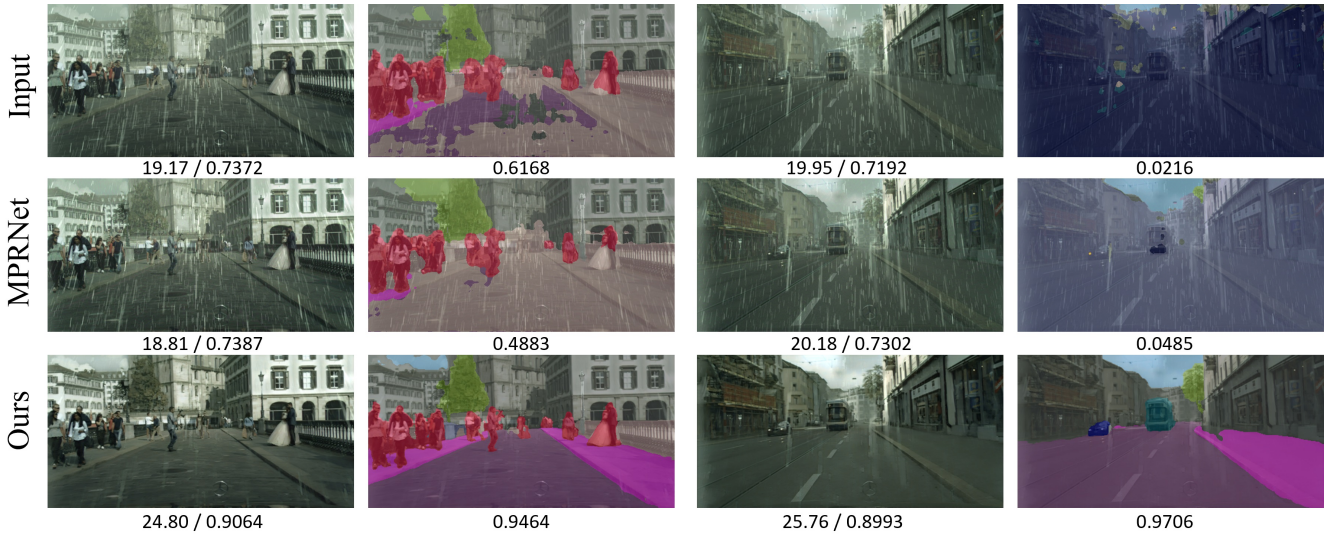


Figure 17. Deraining Images of **MPRNet** and **Ours** against Input-close Attack and its corresponding semantic segmentation overlap in RainCityscape [1]. Note that the adversarial perturbation bound $\epsilon = 4/255$. PSNR/SSIM/mIoU is listed behind each result.

References

[1] S. S. Halder, J. Lalonde, and R. de Charette. Physics-based rendering for improving robustness to rain. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 10202–10211, 2019. 3, 4, 5, 6, 7, 10, 11

[2] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao. Generalizable pedestrian detection: The elephant in the room. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 11328–11337, June 2021. 1

[3] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[4] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(11):2599–2613, 2018. 2

[5] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Proc. IEEE European Conf. Computer Vision*, volume 11211, pages 262–277, 2018. 3

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. Int'l Conf. Learning Representations*, 2018. 2

[7] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 2

[8] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng. Progressive image deraining networks: A better and simpler baseline. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 3937–3946, 2019. 3, 10

[9] H. Wang, Q. Xie, Q. Zhao, and D. Meng. A model-driven deep neural network for single image rain removal. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 3100–3109, 2020. 3

[10] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu. Semi-supervised transfer learning for image rain removal. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 3872–3881, 2019. 3, 8, 9

[11] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 1685–1694, 2017. 3, 4, 8

[12] R. Yasarla and V. M. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning CNN for single image de-raining. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 8405–8414, 2019. 3

[13] S. W. Zamir, A. Arora, S. H. Khan, M. Hayat, F. S. Khan, M. Yang, and L. Shao. Multi-stage progressive image restoration. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. 3

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018. 1, 10

[15] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. D. Newsam, A. Tao, and B. Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 8856–8865, 2019. 1, 4