Learning A Sparse Transformer Network for Effective Image Deraining

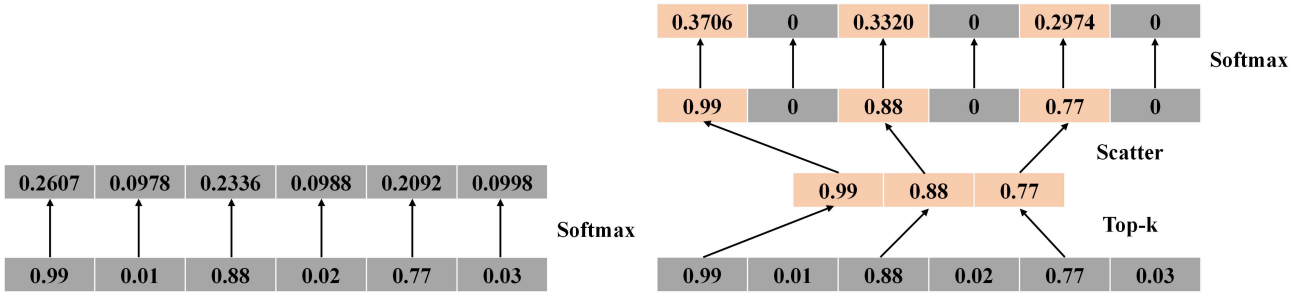
Xiang Chen[1] Hao Li[1] Mingqiang Li[2] Jinshan Pan[1]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology
[2]Information Science Academy, China Electronics Technology Group Corporation

Overview

In this document, we first show a comparison example of the standard self-attention and our proposed top-*k* self-attention in Section 1. Then, we conduct a user study in Section 2. Next, we analyse the generalization ability and model complexity of other tasks in Section 3-4. In Section 5, we show more comparisons. In Section 6, we investigate the impact of deraining performance on downstream vision tasks. Finally, we show more visual comparisons in Section 7.

1. Standard v.s Top-k Self-attention

Following [6], Figure 1 shows an example of normalizing a row vector of the similarity matrix in the standard self-attention (a) and the top-*k* self-attention (b), respectively. In our developed TKSA, we mask out the unnecessary elements assigned with lower attention weights in the attention matrix. Thus, such dynamic selection makes the attention weight matrix from *dense* to *sparse*. We further visualize feature maps of the middle layer of the model created by $Q \times K$ in Figure 2. As can be seen, our method can generate a sparse attention map and produce a much clearer recovery result.



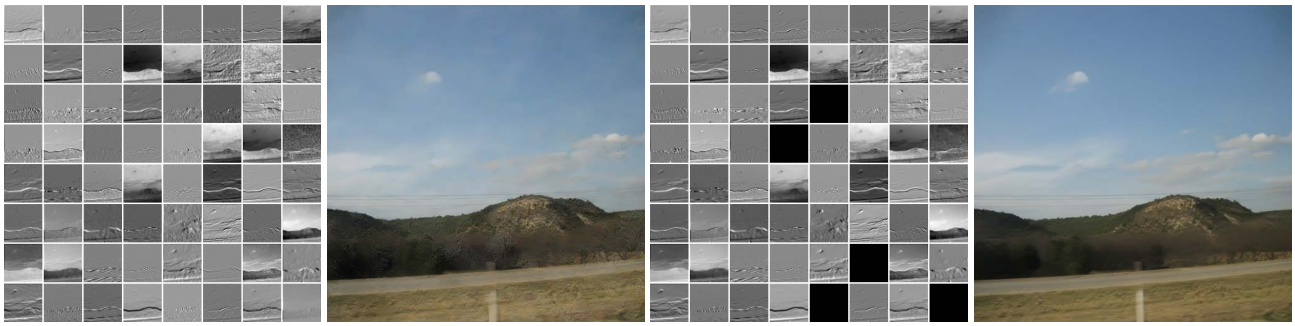
(a) native softmax in the standard self-attention

(b) top-k selection in our developed TKSA

Figure 1. An example of normalizing a row vector of the similarity matrix in the standard self-attention (a) and the top-*k* self-attention (b), respectively. The TKSA keeps the largest K similarity scores between the queries and the keys for the self-attention computing, thereby facilitating better feature aggregation.

2. User Study

We conduct a user study to evaluate the results of different methods. A user-study database is based on the real-world deraining results from the Internet-Data dataset. Users can choose the image with the best deraining performance from a group of images. We make the methods anonymous and randomly sort the images in each group to ensure fairness. We distribute the questionnaire to a wide range of online users without constraints, and finally obtain answers from a total of 186 human evaluators. Figure 3 shows the averaged selection percentage for each method. Our method performs better according to most of human evaluators.



(a) Feature map w/o Top-k (b) Recovery result w/o Top-k (c) Feature map w Top-k (d) Recovery result w Top-k

Figure 2. Feature visualization results of the middle layer of the model created by $Q \times K$ (a, c). Recovery results w or w/o Top-k (b, d). Our method can generate a sparse attention map and produce a clearer recovery result.

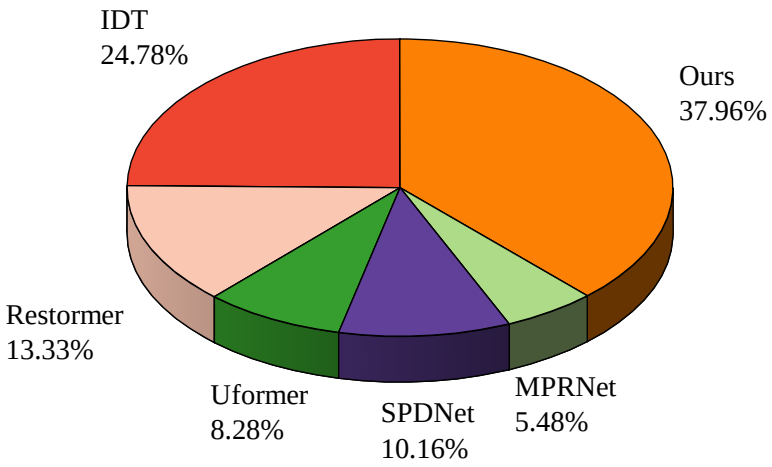


Figure 3. Averaged selection percentage of user study.

3. Generalization In Other Tasks

We extend our framework to the image denoising task. Table 1 reports results of different methods on the SIDD dataset [1]. Our method achieves competitive performance, suggesting generation of our approach in other related restoration tasks.

Table 1. Quantitative comparison (PSNR / SSIM) on the SIDD dataset.

Methods	MPRNet [20]	Uformer [15]	Restormer [19]	IDT [16]	DRSformer
PSNR / SSIM	39.71 / 0.958	39.77 / 0.959	40.02 / 0.960	39.79 / 0.959	39.98 / 0.960

4. Model Complexity

Table 2 shows the complexity comparisons. Since we introduce top-k calculation and cross-scale fusion into our model, our method has limitations in the model efficiency. We will apply the pruning or distillation scheme to maintain the original deraining performance while achieving credible model compression.

Table 2. Comparison of complexity on a 256×256 image.

Methods	MSPFN [7]	IPT [2]	Uformer [15]	Restormer [19]	IDT [16]	DRSformer
FLOPs (G)	595.5	/	45.9	174.7	61.9	242.9
Parameters (M)	13.4	115.5	50.8	26.1	16.4	33.7

5. More Comparisons

As the method [11] does not provide code, we refer to the results of their paper. Table 3 reports quantitative results trained on the Rain100L and Rain100H. Our method still achieves the highest PSNR and SSIM values.

Table 3. Quantitative comparison (PSNR / SSIM) with [9, 11].

Methods	ETDNet [11]	DRT [9]	DRSformer (Ours)
Rain100L	41.09 / 0.986	37.61 / 0.948	**42.49 / 0.990**
Rain100H	32.35 / 0.929	29.47 / 0.846	**33.79 / 0.937**

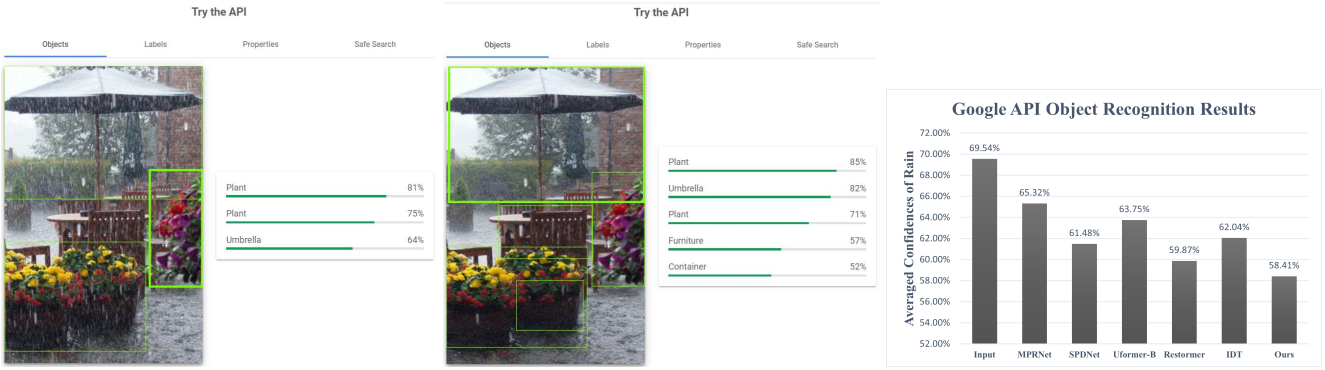
In addition, we also evaluate our method using another dataset, *i.e,* RainDS-Real [12], which collects more challenging data pairs corrupted by raindrops and rain streaks. As we mainly focus on removing rain streaks, we only adopt a subset of RainDS-Real, named Real-RS100, containing 150 real image pairs and 100 test images. Table 4 and Figure 7 show the quantitative and qualitative results on the RainDS-Real dataset respectively.

Table 4. Quantitative comparison (PSNR / SSIM) on the RainDS-Real dataset.

Methods	SPDNet [18]	Uformer [15]	Restormer [19]	IDT [16]	DRSformer
PSNR / SSIM	26.47 / 0.7263	26.83 / 0.7285	27.09 / 0.7463	27.12 / 0.7388	**27.24 / 0.7476**

6. Impact on Downstream Vision Tasks

To investigate the impact of deraining performance on downstream vision tasks, *e.g.*, object recognition, we use Google Vision API tool to evaluate corresponding rain-free outputs. Comparing Figure 4 (a) and (b), the recognition accuracies are improvement by using our deraining results, indicating that DRSformer can better facilitate subsequent detection performance. Similar to [3, 18], Figure 4 (c) compares the averaged confidences of different deraining models in recognizing rainy weather on 20 real rainy images. When confidence is lower, the rain is more light, which shows that the better deraining performance. As one can see, our net can eliminate more rain streaks, resulting in the largest decline, which further verify its effectiveness.



(a) (b) (c)

Figure 4. Comparison results tested on the Google Vision API. (a-b) Object recognition results for the input rainy image and our derained image; (c) The averaged confidences in recognizing rain. Note that lower scores indicate better performance.

7. More Experimental Results

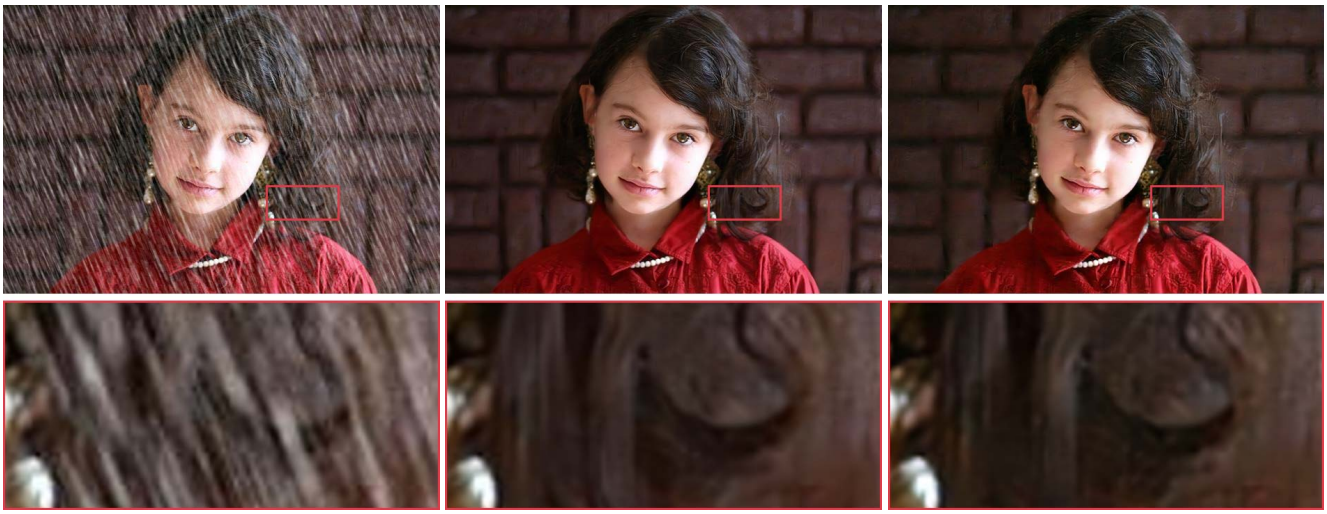
In this section, we show more experimental results to demonstrate the effectiveness of the proposed method. Figures 5-7 show the visual comparison results on the synthetic dataset, including Rain200H [17], DID-Data [21] and DDN-Data [4]. Compared to other methods, our DRSformer can generate high-quality deraining results with more accurate detail and texture recovery. Furthermore, Figures 8 also show the visual comparison results on the real-world dataset, Real-RS100 [12]. Our method can successfully remove most rain streaks and own visual pleasant recovery results.

(a) Rainy Input (b) MPRNet [20] (c) DualGCN [5]

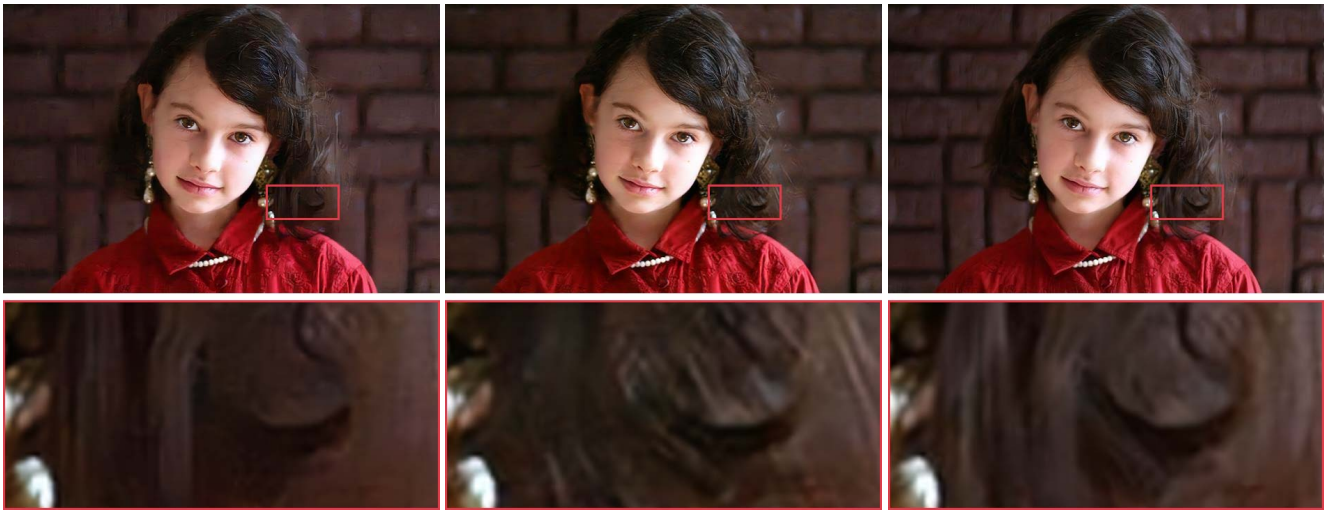
(d) SPDNet [18] (e) Uformer [15] (f) Restormer [19]

(g) IDT [16] (h) DRSformer (Ours) (i) Ground Truth

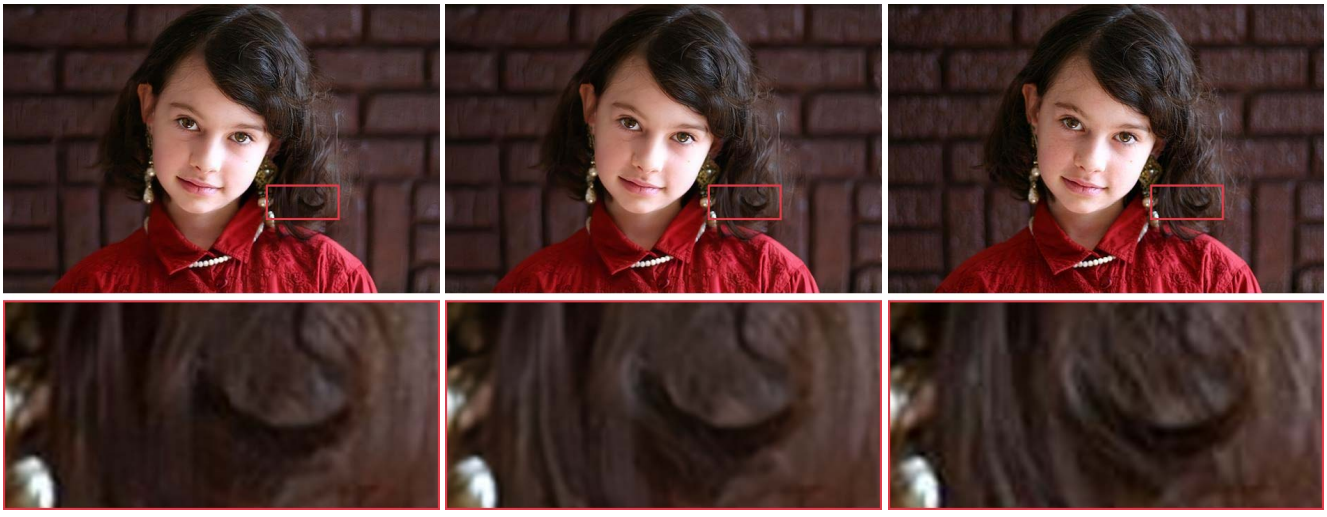
Figure 5. Visual comparison results on the Rain200H dataset [17]. The proposed method generates high-quality deraining results with more accurate detail and texture recovery.



(a) Rainy Input (b) MPRNet [20] (c) DualGCN [5]



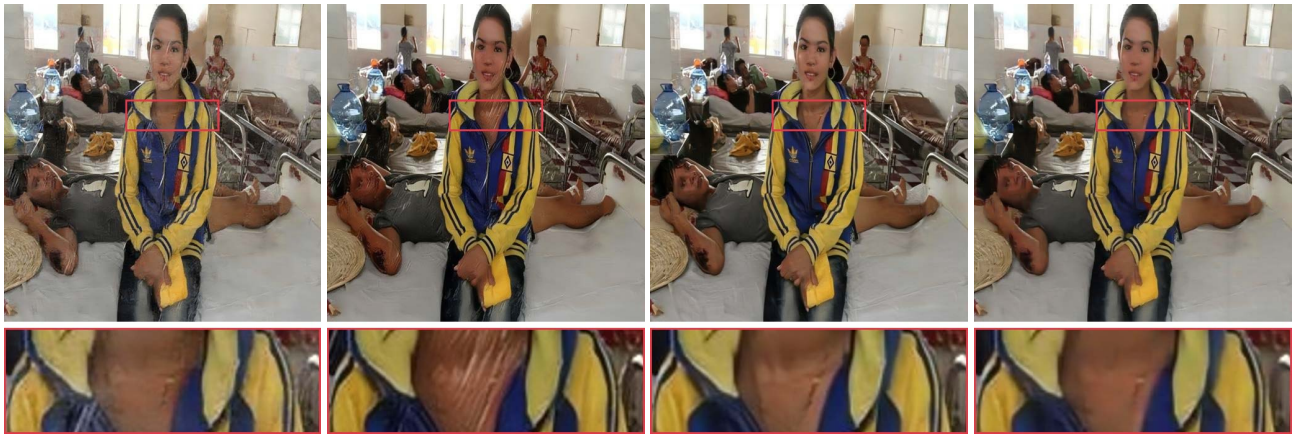
(d) SPDNet [18] (e) Uformer [15] (f) Restormer [19]



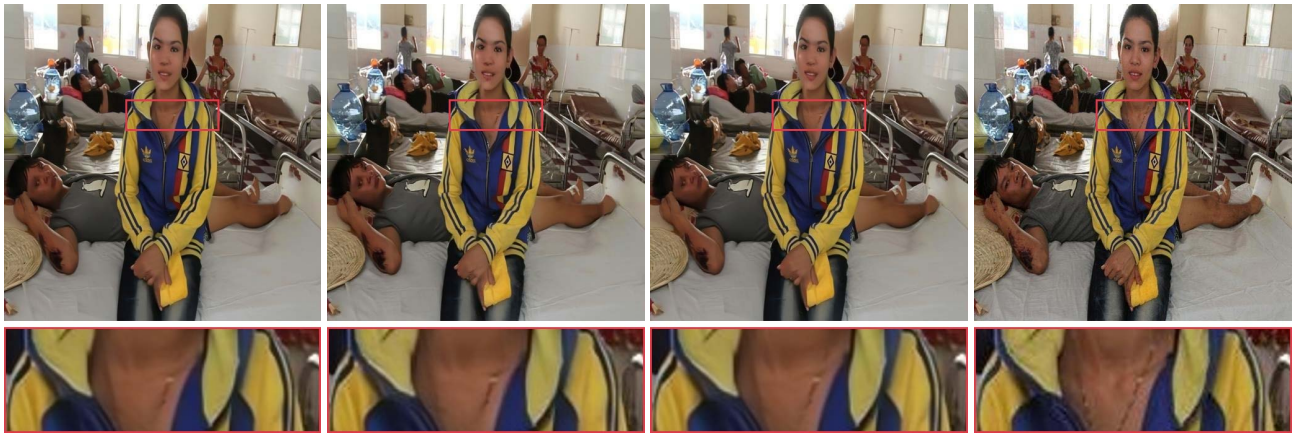
(g) IDT [16] (h) DRSformer (Ours) (i) Ground Truth

Figure 6. Visual comparison results on the DDN-Data dataset [4]. The proposed method generates high-quality deraining results with more accurate detail and texture recovery.

(a) Rainy Input (b) DSC [10] (c) DDN [4] (d) RESCAN [8]

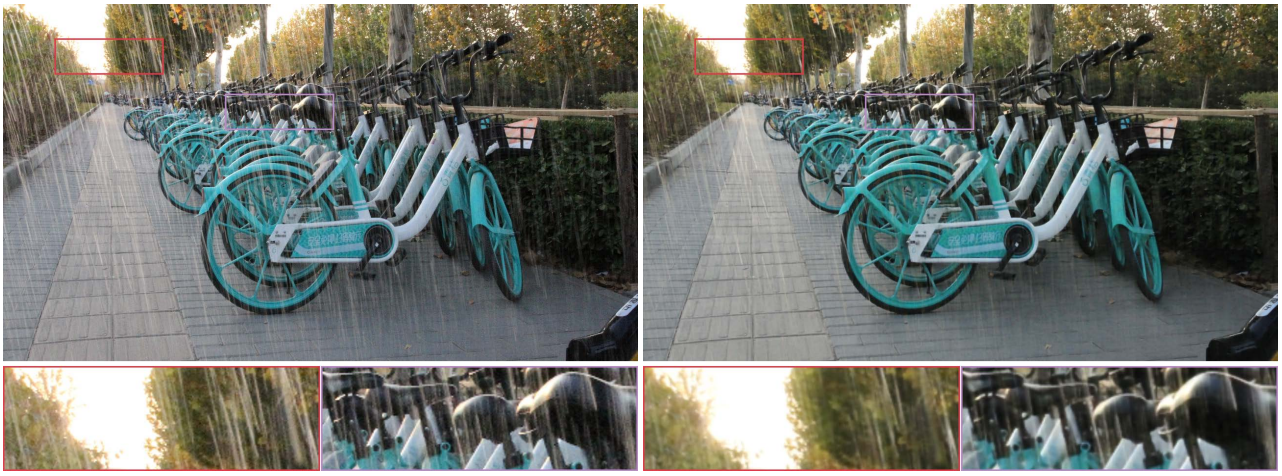


(e) PReNet [13] (f) MPRNet [20] (g) DualGCN [5] (h) SPDNet [18]



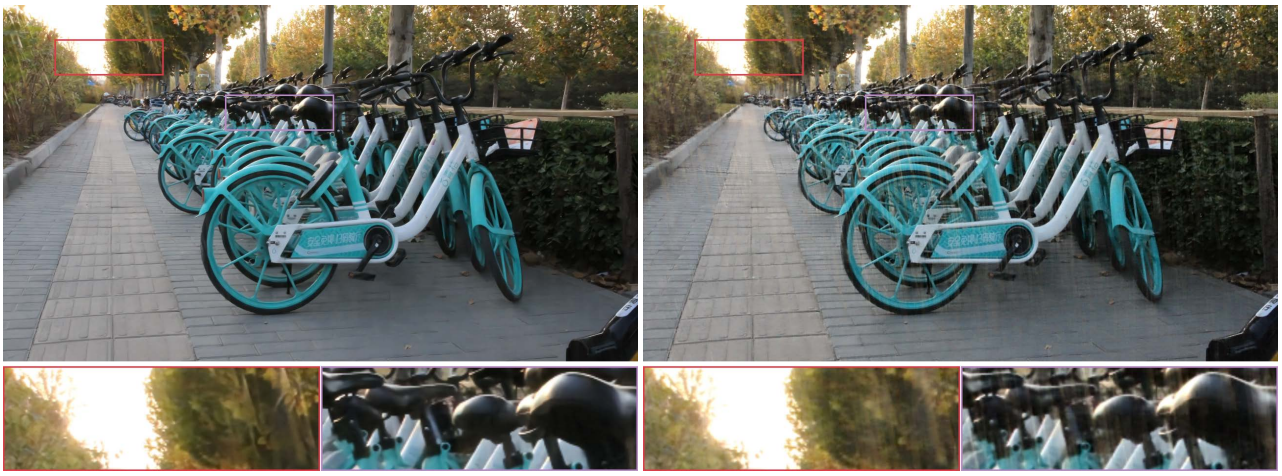
(i) Restormer [19] (j) IDT [16] (k) DRSformer (Ours) (l) Ground Truth

Figure 7. Visual comparison results on the DID-Data dataset [21]. The proposed method generates high-quality deraining results with more accurate detail and texture recovery.



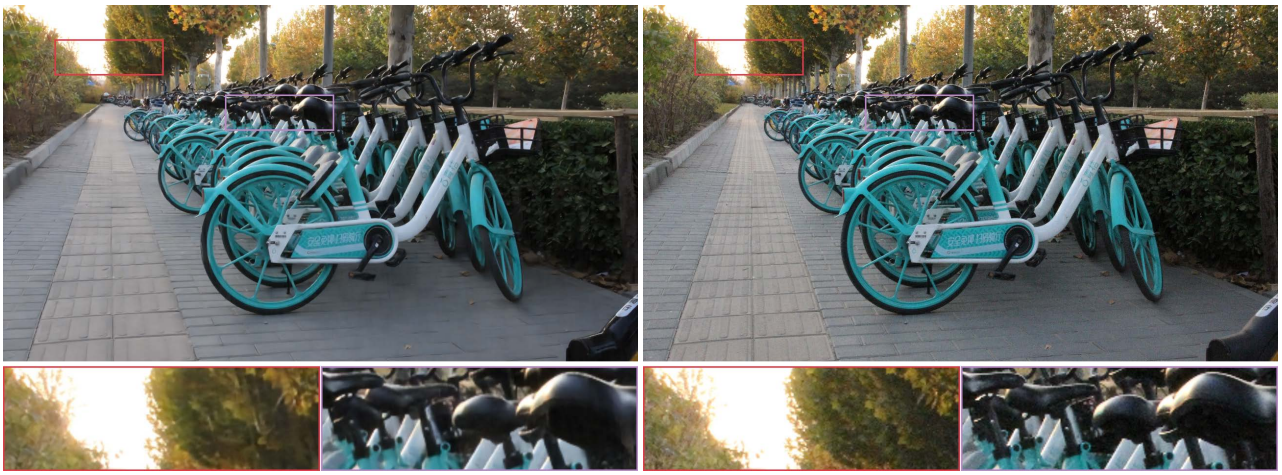
(a) Rainy Input

(b) SPDNet [18]



(c) Restormer [19]

(d) IDT [16]



(e) DRSformer (Ours)

(f) Ground Truth

Figure 8. Visual comparison results on the Real-RS100 dataset [12]. The proposed method removes most rain perturbation and generates much clearer recovery results.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, pages 1692–1700, 2018. 2

[2] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pages 12299–12310, 2021. 2

[3] Xiang Chen, Jinshan Pan, Kui Jiang, Yufeng Li, Yufeng Huang, Caihua Kong, Longgang Dai, and Zhentao Fan. Unpaired deep image deraining using dual contrastive learning. In *CVPR*, pages 2017–2026, 2022. 3

[4] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 3855–3863, 2017. 3, 5, 6

[5] Xueyang Fu, Qi Qi, Zheng-Jun Zha, Yurui Zhu, and Xinghao Ding. Rain streak removal via dual graph convolutional network. In *AAAI*, pages 1352–1360, 2021. 4, 5, 6

[6] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*, 2022. 1

[7] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, pages 8346–8355, 2020. 2

[8] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 254–269, 2018. 6

[9] Yuanchu Liang, Saeed Anwar, and Yang Liu. Drt: A lightweight single image deraining recursive transformer. In *CVPRW*, pages 589–598, 2022. 3

[10] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015. 6

[11] Qin Qin, Jingke Yan, Qin Wang, Xin Wang, Minyao Li, and Yuqing Wang. Etdnet: An efficient transformer deraining model. *IEEE Access*, 9:119881–119893, 2021. 3

[12] Ruijie Quan, Xin Yu, Yuanzhi Liang, and Yi Yang. Removing raindrops and rain streaks in one go. In *CVPR*, pages 9147–9156, 2021. 3, 7

[13] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, pages 3937–3946, 2019. 6

[14] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, pages 12270–12279, 2019.

[15] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, pages 17683–17693, 2022. 2, 3, 4, 5

[16] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE TPAMI*, 2022. 2, 3, 4, 5, 6, 7

[17] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, pages 1357–1366, 2017. 3, 4

[18] Qiaosi Yi, Juncheng Li, Qinyan Dai, Faming Fang, Guixu Zhang, and Tieyong Zeng. Structure-preserving deraining with residue channel prior guidance. In *ICCV*, pages 4238–4247, 2021. 3, 4, 5, 6, 7

[19] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 2, 3, 4, 5, 6, 7

[20] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, pages 14821–14831, 2021. 2, 4, 5, 6

[21] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, pages 695–704, 2018. 3, 6