Supplementary Material – Sparse Sampling Transformer with Uncertainty-Driven Ranking for Unified Removal of Raindrops and Rain Streaks

Sixiang Chen[1,3*] Tian Ye[1,3*] Jinbin Bai[2] Erkang Chen[3]
Jun Shi[4] Lei Zhu[1,5†]

[1]The Hong Kong University of Science and Technology (Guangzhou) [2]National University of Singapore
[3]School of Ocean Information Engineering, Jimei University
[4]Xinjiang University [5]The Hong Kong University of Science and Technology
{*sixiangchen*, *owentianye*} @hkust-gz.edu.cn
jinbin.bai@u.nus.edu, ekchen@jmu.edu.cn, junshi2022@gmail.com, leizhu@ust.hk
Project page: https://github.com/Ephemeral182/UDR-S2Former_deraining

This is a supplementary material for Sparse Sampling Transformer with Uncertainty-Driven Ranking for Unified Removal of Raindrops and Rain Streaks.

We provide the following materials in this manuscript:

- Sec.1 Formulation of rain streaks and raindrops.
- Sec.2 more details on the UDR-S^{2}Former pipline.
- Sec.3 performance vs. run-time.
- Sec.4 performance vs. memory cost.
- Sec.5 additional ablation studies.
- Sec.6 more visual comparisons.
- Sec.7 future works.

1. Formulation

Rain streaks. Rain streaks are defined as the visible lines that accumulate on an image due to rain. To address this issue, a clean background scene $\mathcal{B}$ is defined and added to the accumulated rain streaks $\mathcal{S}$ to form the rain streak image $\mathcal{R}_s$:

$$\mathcal{R}_s = \mathcal{B} + \mathcal{S} \tag{1}$$

which can then be used to obtain the clean image $\mathcal{B}$ by removing the rain streaks $\mathcal{S}$.

Raindrops. Similarly, raindrops can distort images, but this distortion can be decomposed into two parts: a clean background $\mathcal{B}$ and blurry or obstruction effects of the raindrops $\mathcal{D}$ in small scattered regions, the raindrops model can be expressed as:

$$\mathcal{R}_d = (1 - \mathcal{M}_r) \odot \mathcal{B} + \mathcal{D}, \tag{2}$$

where $\mathcal{M}$ is a binary mask that identifies which pixels belong to the raindrop regions and which belong to the background. Raindrop removal aims to obtain a clear, rain-free image by eliminating the raindrop effect.

In the real world, these two kinds of degradations often occur together, and they cannot be simply combined, so it is vital to develop networks that can jointly remove them to deal with this situation.

2. The Proposed Architecture

2.1. Uncertainty Map in UDR-S^{2}Former

For the framework we designed, we need to output two variables, the uncertainty map and the expected rain-free image $\mathcal{B}_{pre}$. For the uncertain output, we are consistent with the previous work [8], using a simple Conv-ELU to generate the uncertainty map for the place where uncertainty is required.

2.2. Convolutional Block

Our UDR-S^{2}Former approach employs simple convolutional blocks to extract degraded image features during the Feature Extraction stage. As presented in Fig.1 (a), we aim to extract complex rain scene information in high-dimensional space for subsequent global modeling. Inspired by previous work [1], we first increase the channel dimension to twice that of the input channels to construct more discriminative features. Then use the **DWConv-LN-GELU-DWConv** design to capture sufficient rain information. Finally, we strengthen channel interaction by

*Equal contributions.
†Lei Zhu (leizhu@ust.hk) is the corresponding author.

Table 1: Comparison of speed and GFLOPs of previous SOTA deraining methods in different resolutions (§3).

	256×256 patch		320×320 patch		448×448 patch		512×512 patch		PSNR ↑
	#GFLOPs (G)	#Times (s)	#GFLOPs (G)	#Times (s)	#GFLOPs (G)	#Times (s)	#GFLOPs (G)	#Times (s)	
CCN [9]	245.85	0.031	384.13	0.046	752.90	0.090	983.38	0.115	34.79
MPRNet [15]	148.55	0.025	232.11	0.036	454.94	0.067	594.20	0.084	34.99
DGUNet [7]	199.74	0.044	312.09	0.061	611.71	0.115	798.95	0.147	35.34
Uformer [12]	19.69	0.021	30.76	0.034	60.30	0.078	78.76	0.084	34.99
Restormer [14]	140.99	0.086	220.30	0.133	Out of Memory	Out of Memory	Out of Memory	Out of Memory	36.08
IDT [13]	57.89	0.068	90.46	0.128	Out of Memory	Out of Memory	Out of Memory	Out of Memory	36.23
UDR-S^{2}Former	21.58	0.031	33.72	0.042	66.10	0.071	86.33	0.082	<u>36.91</u>

adding a channel attention mechanism [6] before reducing the number of channels. Experiments demonstrate that our approach is simple and effective, ensuring excellent information extraction performance while maintaining low inference time and computational complexity.

2.3. Transformer Block

After Feature Extraction, we perform deep-level global modeling using the vanilla transformer block, including multi-head self-attention [5] and multi-scale feedforward network [3]. For the obtained feature $\mathcal{F}_E \in \mathbb{R}^{H\times W\times C}$, we reshape it to $\mathcal{F}_E \in \mathbb{R}^{N\times C}$ ($N = H \times W$) and adopt learnable $\mathcal{W}_{\mathcal{Q}}^{C\times C}$, $\mathcal{W}_{\mathcal{K}}^{C\times C}$ and $\mathcal{W}_{\mathcal{V}}^{C\times C}$ to project the $\mathcal{F}_E$ into $\mathcal{Q}$ ($\mathcal{F}_E\mathcal{W}_{\mathcal{Q}}$), $\mathcal{K}$ ($\mathcal{F}_E\mathcal{W}_{\mathcal{K}}$) and $\mathcal{V}$ ($\mathcal{F}_E\mathcal{W}_{\mathcal{V}}$). To improve the local extraction ability, we also add a 3×3 depthwise convolution:

$$\textbf{Softmax}\left(\frac{\mathcal{Q}\mathcal{K}^{\mathrm{T}}}{\sqrt{C}}\right)\times\mathcal{V}+\textbf{DWConv}(\mathcal{F}_E), \tag{3}$$

where H, W, and C represent the height, width, and number of dimensions. Softmax(·) is the the Softmax operation. To carry out the above operations, we utilize multi-head self-attention following the approach outlined in [5].

For the feedforward stage, we utilize the multi-scale feedforward network [3] to enhance the modeling of complex and diverse degradations in rain scenes. To this end, the transformer block can be expressed as:

$$\begin{aligned}\mathcal{F}'_E &= \mathcal{F}_E + \textbf{MSA}(\textbf{LN}(\mathcal{F}_E)),\\ \hat{\mathcal{F}}_E &= \mathcal{F}'_E + \textbf{MFFN}(\textbf{LN}(\mathcal{F}'_E)),\end{aligned} \tag{4}$$

where $\hat{\mathcal{F}}_E$ denotes the feature processed by the global modeling via transformer block. **LN** is the LayerNorm, and the residual connection is employed followed by vanilla ViT [5].

2.4. Refinement Block

For image restoration, incorporating a refinement module in the final stage of the network can aid in detail-focused processing. Specifically, the proposed refinement block employs channel attention to refine features at different stages, ultimately achieving non-trivial restoration of details in the final feature map. The detail of such a design is shown in Fig.1 (b).

3. Computation Time Comparison

In this section, we showcase our speed advantages. We conduct all inference stages on an RTX3090 GPU to ensure a fair comparison. We use the $torch.cuda.synchronize()$ API function to obtain accurate feed-forward runing times. Our results in Table 1 demonstrate that UDR-S^{2}Former clearly outperforms the previous method CCN [4] for unified removal of rain streaks and raindrops, in terms of GFLOPs, inference time, and PSNR gain, particularly for high resolution. Furthermore, our proposed UDR-S^{2}Former achieves superior model complexity and runing time performance compared to other general image restoration methods. Even the state-of-the-art Restormer [14] architecture encountered a "CUDA out of memory" issue when processing high-resolution images. Additionally, UDR-S^{2}Former significantly surpasses IDT [13] in running time and GFLOPs, while maintaining an excellent balance between PSNR and speed (36.23PSNR → 36.91PSNR).

4. Memory Cost Comparison

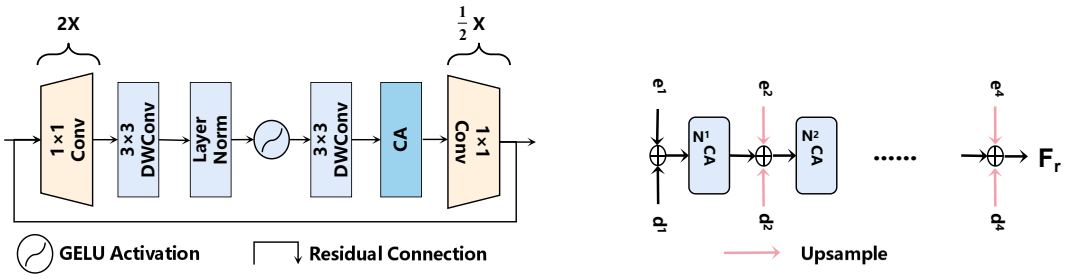
In addition to comparing speed, we also measure the memory cost of our proposed approach for rain removal using a single RTX3090 GPU. As demonstrated in Table 4, UDR-S^{2}Former incurs the lowest memory cost during inference, indicating the suitability of our method for practical applications.

5. Additional Ablation Studies

In order to fully research the proposed UDR-S^{2}Former, we present a more exhaustive set of ablation studies. The individual components are then elucidated in the subsequent sections.

5.1. Superiority of Convolutional Block

In this section, we research the effectiveness of the proposed convolutional block by experimenting with various configurations. Specifically, we investigate the importance



(a) Convolutional Block **(b) Refinement Block**

Figure 1: Detailed structure of proposed convolutional block and refinement block.

Table 2: Ablation studies on Convolutional Block (§2.2).

Setting	Model	#Param	#GFLOPs	PSNR	SSIM
i	1X	5.32M	19.66G	36.37	0.962
ii	3X	11.72M	22.94G	36.96	0.967
iii	One DWConv	8.45M	21.37G	36.68	0.964
iv	w/o CA	7.03M	21.56G	36.53	0.963
v	Ours	8.53M	21.58G	36.91	0.966

Table 3: Ablation studies on Transformer Block (§2.3).

Setting	Model	#Param	#GFLOPs	PSNR	SSIM
i	w/o DWConv	8.50M	21.58G	36.81	0.965
ii	MLP [5]	8.76M	21.72G	36.66	0.964
iii	ConvFFN [11]	8.62M	21.69G	36.64	0.964
iv	LeFF [12]	8.38M	21.52G	36.79	0.965
v	Ours	8.53M	21.58G	36.91	0.966

Table 4: Comparison of memory cost in inference process tested with 256×256 resolution (§4).

Method	Memory Cost	PSNR ↑
(CVPR'2021)CCN [4]	2874M	34.79
(CVPR'2021)MPRNet [15]	2527M	34.99
(CVPR'2022)DGUNet [7]	3618M	35.34
(CVPR'2022)Uformer [12]	3019M	34.99
(CVPR'2022)Restormer [14]	3696M	36.08
(TPAMI'2022)IDT [13]	3547M	36.23
UDR-S^{2}Former	2502M	36.91

of critical designs such as high-dimensional space amplification, the number of DWConv, and channel attention. Our findings, presented in Table 2, indicate that leveraging high-dimensional space can capture complicated rain degradation information effectively. Additionally, the DWConv layers significantly enhance local feature extraction without significantly increasing the number of parameters. Furthermore, channel attention aids in channel-wise modeling in high dimensions. It's important to note that while a more extensive scaling factor can improve network performance, it requires more time and computational resources for training. As such, we set the scaling factor to 2 in our paper.

5.2. Improvements of Transformer Block

Our study aims to assess the effects of different components within the transformer block. Specifically, we examine the impact of (i) excluding the DWConv operation from the transformer block (w/o DWConv), (ii) using a traditional MLP-based feed-forward network [5] instead of the presented MFFN (MLP), (iii) employing the ConvFFN [11] instead of the proposed MFFN (ConvFFN), (iv) comparing the performance of LeFF [12] and MFFN (LeFF), and (v) using the MFFN to demonstrate the superiority of our approach (Ours). Our experimental results, reported in Table 3, show that the DWConv operation can improve performance by complementing self-attention. Moreover, the MFFN has the most significant impact on the PSNR metric compared to other notable designs.

5.3. Gains of Refinement Block

We conduct ablation experiments on refinement block (RB) and present our advantages compared to the corresponding paradigm of refinement block in the Restormer [14] (Single-stage). Table 5 shows that compared with the Restormer, we can achieve better results by using multi-stage refinement to preserve more perfect details.

Table 5: Ablation studies on Refinement Block (§2.4).

Setting	Model	#Param	#GFLOPs	PSNR	SSIM
i	w/o RB	8.39M	12.53G	36.42	0.962
ii	Single-stage [14]	8.43M	14.56G	36.56	0.963
iii	Ours	8.53M	21.58G	36.91	0.966

5.4. Effectiveness of Loss Function

In our study, we investigate different loss functions to determine their effectiveness. We choose PSNRloss [2] instead of L1 loss as our reconstruction loss and demonstrate

Table 6: Comparison of different sets on loss functions (§5.4).

Setting	Module						Metric
	$\mathcal{L}_1$	$\mathcal{L}_{psnr}$	$\mathcal{L}_{perceptual}$ (1,3,5,9,13)	$\mathcal{L}_{perceptual}$ (1,3)	$\mathcal{L}_{UDL}$	intermediate$\mathcal{L}_{UDL}$	PSNR/SSIM
Baseline	✓				✓	✓	36.71 / 0.964
ii		✓			✓	✓	36.79 / 0.965
iii		✓	✓		✓	✓	<u>36.94</u> / <u>0.966</u>
iv		✓		✓			Nan / Nan
v		✓		✓	✓		36.69 / 0.964
vi (Ours)		✓		✓	✓	✓	36.91 / 0.966

that perceptual loss and uncertainty loss are superior for our training supervision. Table 6 shows that PSNRloss has an advantage over L1 loss, and perceptual loss is beneficial during training due to its supervision at the feature level. For the rain removal task, we only use the 1-th and 3-th shallow layers of the VGG [10] network, which saves GPU memory and achieves impressive results. Finally, we find that uncertainty loss helps the network generate uncertainty maps. Intermediate layer supervision is more robust to the uncertainty maps at each stage in our image reconstruction module.

6. More Visual Comparisons

We present more visual comparisons against state-of-the-art methods on synthetic and real datasets to demonstrate the excellent visual performance of UDR-S^{2}Former on removing raindrops and rain streaks.

6.1. Synthectic Datasets

UDR-S^{2}Former can effectively eliminate complex and challenging rain degradations as demonstrated in Fig.2 and Fig.3, due to its robust degradation relationship modeling. Additionally, the ability to restore fine details is also noticeable compared with previous state-of-the-art methods.

6.2. Real-world Dataset

To demonstrate the impressive performance of our model in real-world scenarios, we present extensive visual comparisons in Fig.4 and Fig.5. Our observations indicate that our UDR-S^{2}Former can improve image quality and effectively remove complex degradations, while other algorithms often struggle with complex rain degradations. Moreover, UDR-S^{2}Former outperforms state-of-the-art methods in effectively handling diverse degradations.

7. Future Work

In future, we plan to explore the potential of uncertainty for various tasks while enhancing our overall architecture to make it more efficient and less computationally complex.

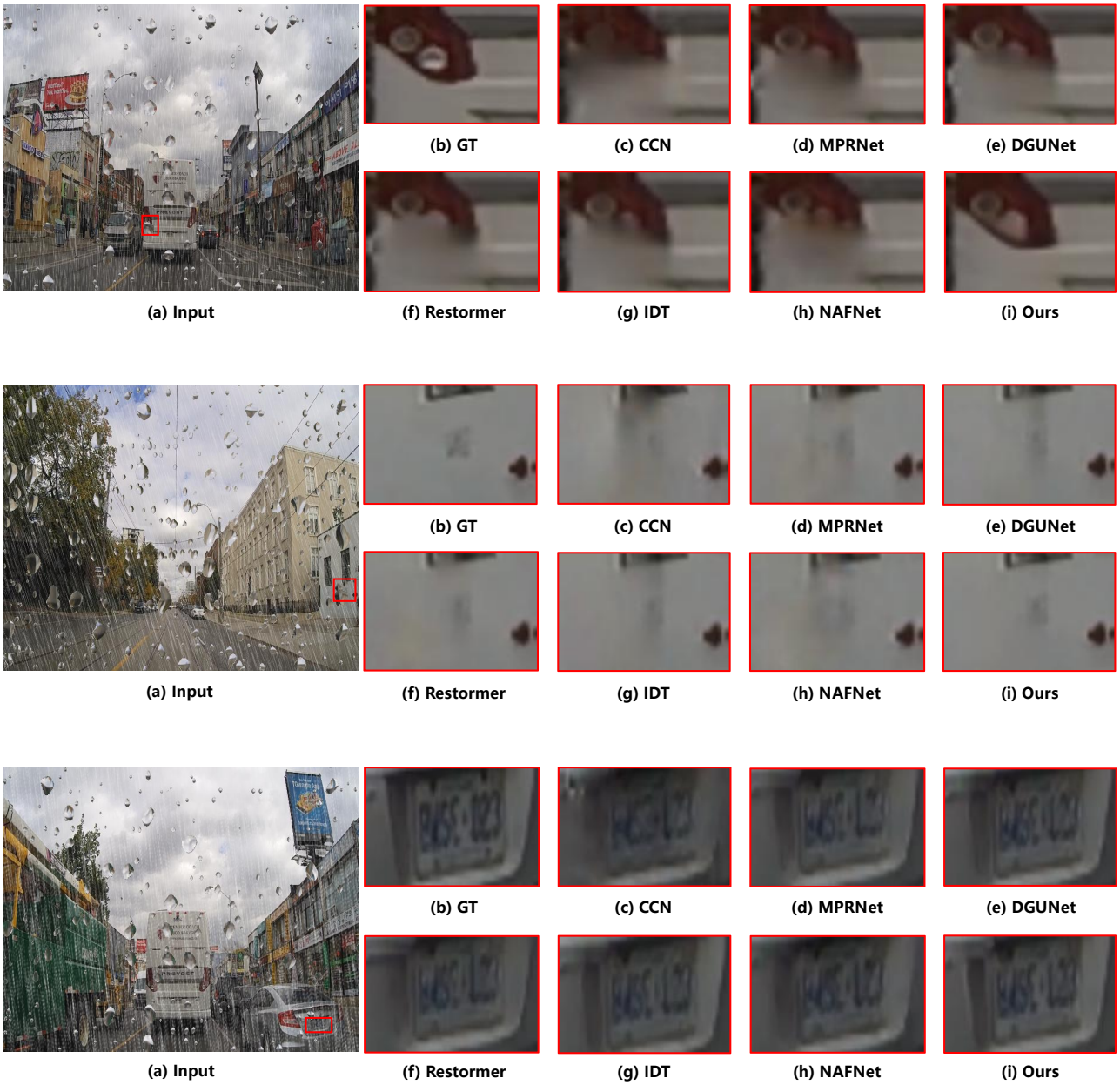


Figure 2: Visual comparisons for unified rain removal on synthetic dataset among CCN [9], MPRNet [15], DGUNet [7], Restormer [14], IDT [13], NAFNet [1] and our proposed UDR-S^{2}Former. Please zoom them for better watching.

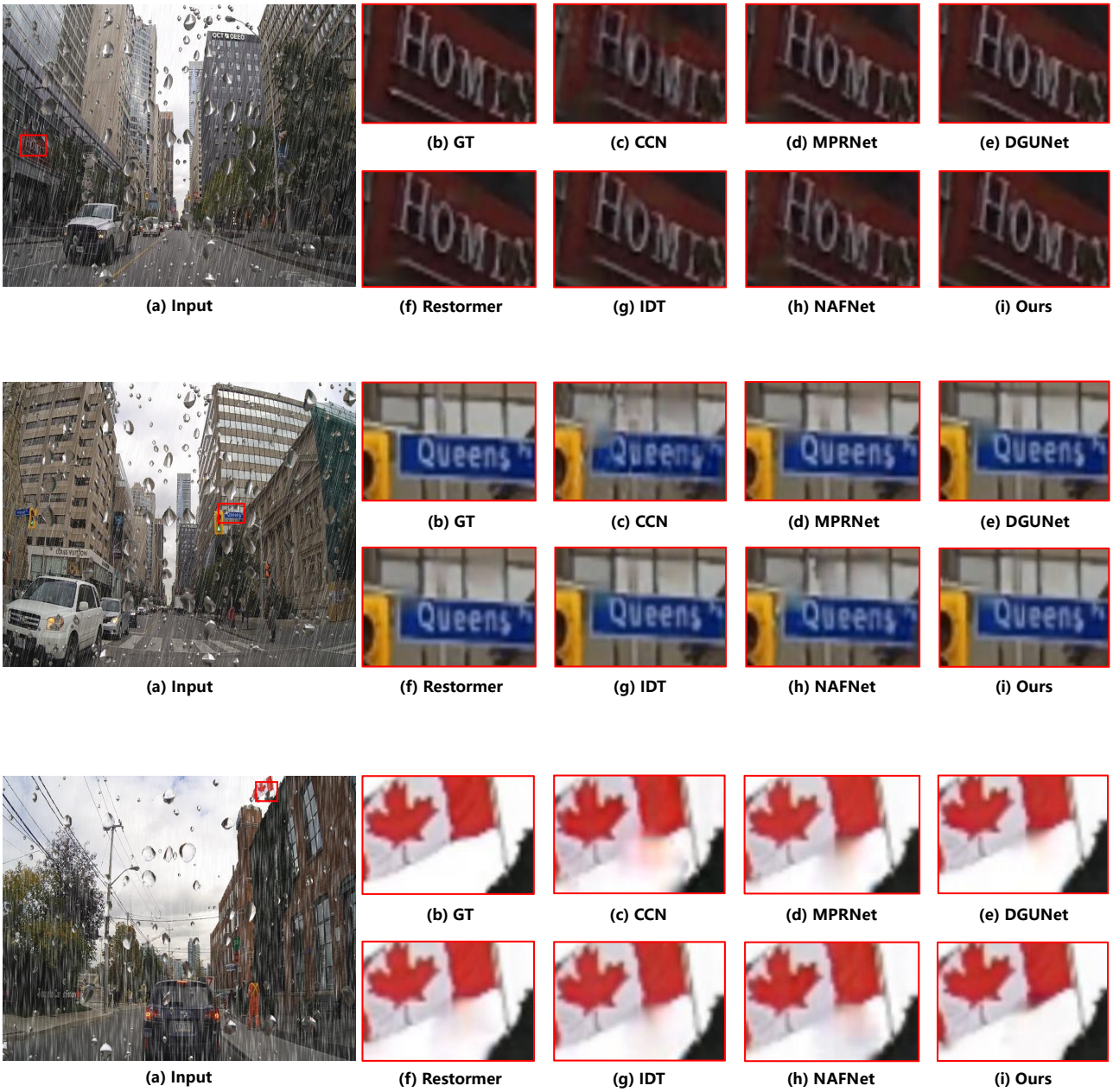


Figure 3: Visual comparisons for unified rain removal on synthetic dataset among CCN [9], MPRNet [15], DGUNet [7], Restormer [14], IDT [13], NAFNet [1] and our proposed UDR-S^{2}Former. Please zoom them for better watching.

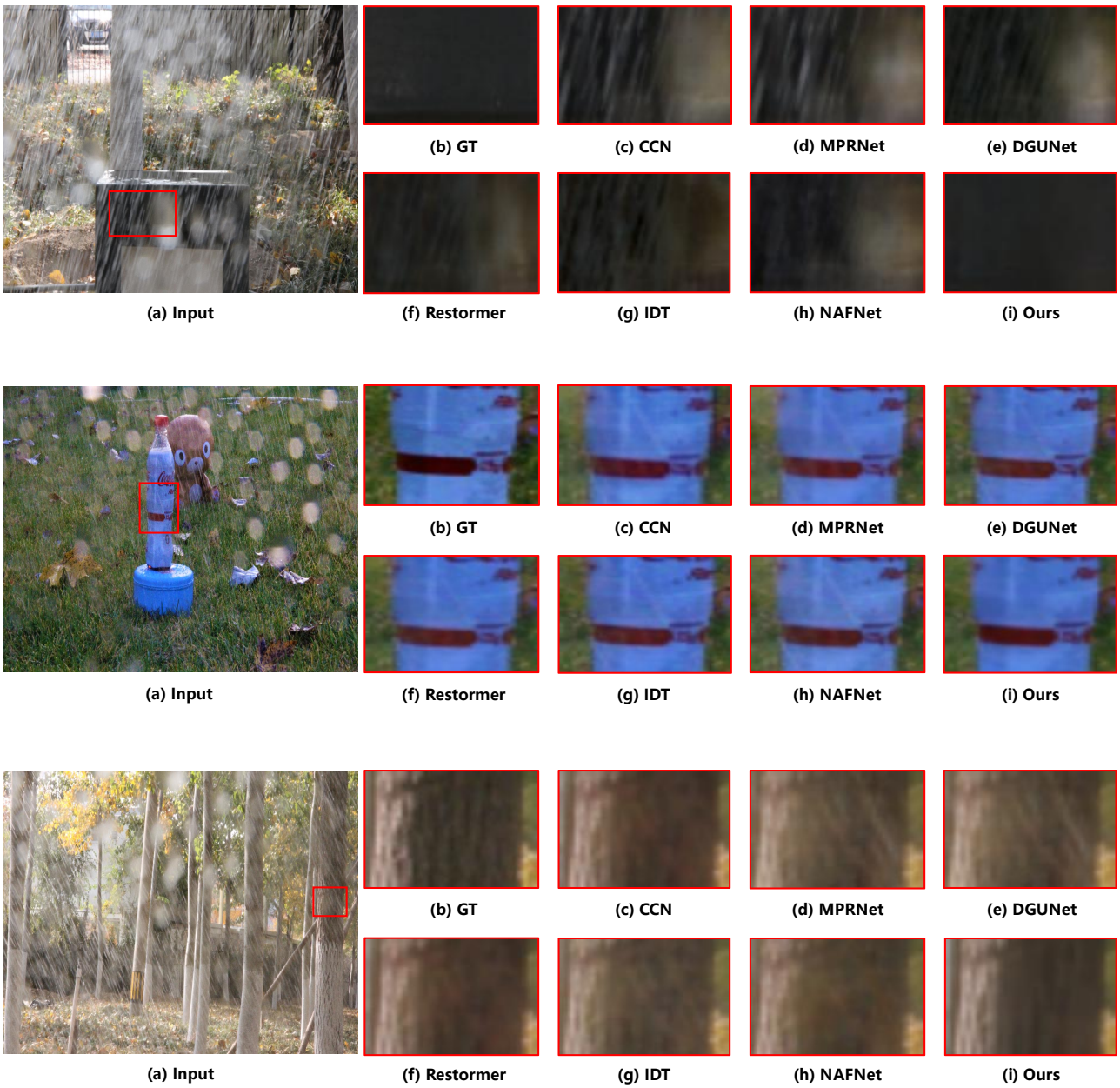


Figure 4: Visual comparisons for unified rain removal on real-world dataset among CCN [9], MPRNet [15], DGUNet [7], Restormer [14], IDT [13], NAFNet [1] and our proposed UDR-S^{2}Former. Please zoom them for better watching.

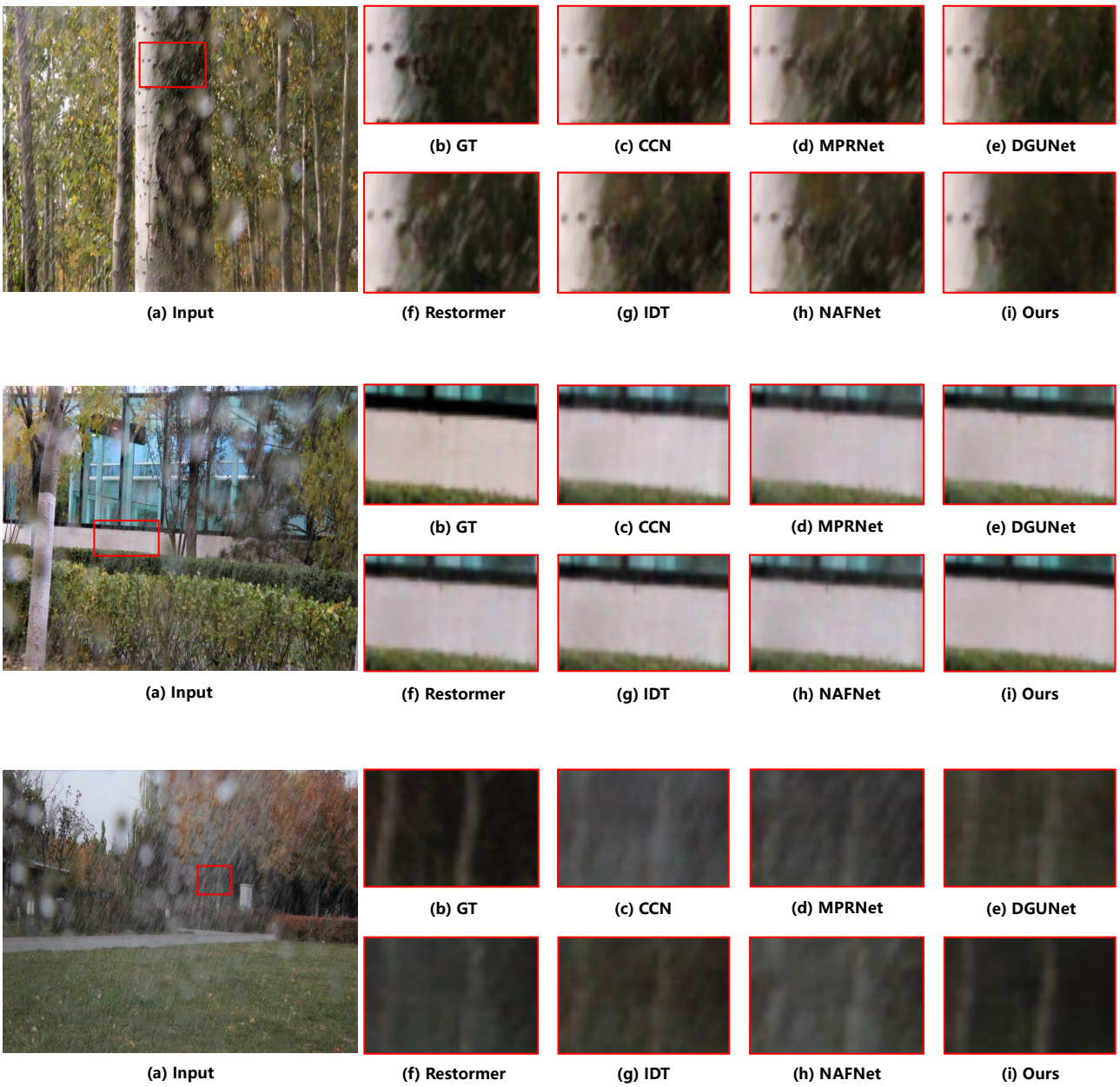


Figure 5: Visual comparisons for unified rain removal on real-world dataset among CCN [9], MPRNet [15], DGUNet [7], Restormer [14], IDT [13], NAFNet [1] and our proposed UDR-S^{2}Former. Please zoom them for better watching.

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pages 17–33. Springer, 2022. 1, 5, 6, 7, 8

[2] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–192, 2021. 3

[3] Sixiang Chen, Tian Ye, Yun Liu, and Erkang Chen. Dualformer: Hybrid self-attention transformer for efficient image restoration. *arXiv preprint arXiv:2210.01069*, 2022. 2

[4] Bodong Cheng, Juncheng Li, Ying Chen, Shuyi Zhang, and Tieyong Zeng. Snow mask guided adaptive residual network for image snow removal. *arXiv preprint arXiv:2207.04754*, 2022. 2, 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[7] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17399–17410, 2022. 2, 3, 5, 6, 7, 8

[8] Qian Ning, Weisheng Dong, Xin Li, Jinjian Wu, and Guangming Shi. Uncertainty-driven loss for single image superresolution. *Advances in Neural Information Processing Systems*, 34:16398–16409, 2021. 1

[9] Ruijie Quan, Xin Yu, Yuanzhi Liang, and Yi Yang. Removing raindrops and rain streaks in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9147–9156, 2021. 2, 5, 6, 7, 8

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[11] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 3

[12] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 2, 3

[13] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3, 5, 6, 7, 8

[14] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 2, 3, 5, 6, 7, 8

[15] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. 2, 3, 5, 6, 7, 8