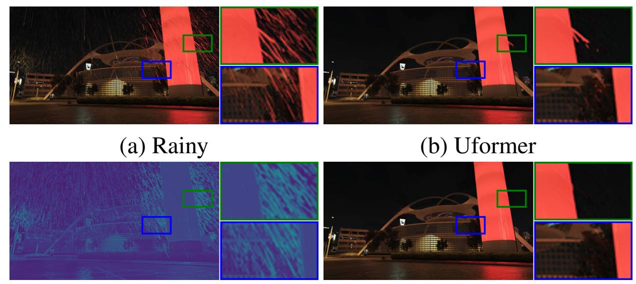
Learning Rain Location Prior for Nighttime Deraining

Fan Zhang[1] Shaodi You[2] Yu Li[3] Ying Fu[1*]
[1]Beijing Institute of Technology [2]University of Amsterdam
[3]International Digital Economy Academy

Abstract

Rain can significantly degrade image quality and visibility, making deraining a critical area of research in computer vision. Despite recent progress in learning-based deraining methods, there is a lack of focus on nighttime deraining due to the unique challenges posed by non-uniform local illuminations from artificial light sources. Rain streaks in these scenes have diverse appearances that are tightly related to their relative positions to light sources, making it difficult for existing deraining methods to effectively handle them. In this paper, we highlight the importance of rain streak location information in nighttime deraining. Specifically, we propose a Rain Location Prior (RLP) that is learned implicitly from rainy images using a recurrent residual model. This learned prior contains location information of rain streaks and, when injected into deraining models, can significantly improve their performance. To further improve the effectiveness of the learned prior, we also propose a Rain Prior Injection Module (RPIM) to modulate the prior before injection, increasing the importance of features within rain streak areas. Experimental results demonstrate that our approach outperforms existing state-of-the-art methods by about 1dB and effectively improves the performance of deraining models. We also evaluate our method on real night rainy images to show the capability to handle real scenes with fully synthetic data for training. Our method represents a significant step forward in the area of nighttime deraining and highlights the importance of location information in this challenging problem. The code is publicly available at https://github.com/zkawfanx/RLP.



(a) Rainy (b) Uformer
(c) RLP (d) Ours

Figure 1: (a) In night scenes, appearances of rain are highly dependent on locations relative to light sources. (b) Derain result of state-of-the-art method Uformer [34]. (c) Visualization of the proposed Rain Location Prior (RLP), which indicates the location of rain streaks and suppresses areas of light sources. (d) Derain result of our method.

1. Introduction

While rain degrades image quality and visibility, it can be particularly problematic at night, because the low light condition and complex illuminations make it more difficult to distinguish rain streaks from other image features. This not only hinders the performance of high-level vision algorithms, *e.g.*, object detection [13, 29] and tracking [2, 35], but also poses a safety risk for autonomous vehicles [6, 28] and other applications that rely on accurate visual information. Therefore, there is an increasing need for developing nighttime deraining methods.

Unfortunately, despite recent advances in deep learning [34, 39, 40], which have significantly improved the performance, current endeavors predominantly focus on daytime scenarios, leaving the realm of nighttime deraining still largely unexplored. The most frequently utilized datasets [16, 32] scarcely encompass night scenes, neither does the synthesizing procedure consider interactions between rain and light sources. This hinders the direct application of existing methods to nighttime deraining. Recently, some datasets [20, 42] have started to cover previously ignored night scenes, which may enable careful study on nighttime deraining. Since there is no prior method specifically targeting nighttime deraining, we attempt to handle real night rainy scenes with the help of the recently developed dataset.

Night scenes differ from day scenes in that there is no uniform global illumination; instead, darkness and non-uniform local illuminations dominate [12, 22, 25, 41, 45]. It results in the totally different appearance of night rain, which varies drastically at different spatial locations (*i.e.*

*Corresponding author: fuying@bit.edu.cn

only rain streaks near light sources are more discernible, as shown in Figure 2). Intuitively, it could facilitate the removal of rain streaks if precise locations were obtained in advance, which is often impractical in real scenes. Likewise, it is a common practice [14, 26] to train deep models to predict the locations of raindrop (*i.e*. raindrop mask) in the raindrop removal community. Therefore, identifying the rain location becomes more crucial than in day scenes.

In this paper, we propose a novel deraining method targeting night scenes. Firstly, we highlight the importance of location information of rain streaks for nighttime deraining, as the appearance of rain streaks is highly varying at different spatial locations at night. Thus, we propose a Rain Location Prior (RLP) which can be implicitly learned by recurrent residual models. It can reveal the location information of rain streaks in night scenes and be incorporated into deraining methods to improve their performance, including CNN-based and Transformer-based ones. Secondly, we propose a Rain Prior Injection Module (RPIM) to increase the importance of features within rain streak areas indicated by RLP. Deraining models can focus more on recovering lost information within rain streak areas and get a further increase in performance. Finally, we perform comprehensive experiments to demonstrate the effectiveness of our method. Our method outperforms existing state-of-the-art methods quantitatively and qualitatively on synthetic data. Furthermore, we demonstrate the capability of our method in handling real scenes. We also conduct ablation studies to evaluate the effectiveness of each component in our method. To summarize, the contributions of our paper are as follows:

- We propose a nighttime deraining method, which highlights the importance of location information of rain streaks for night scenes.
- We propose Rain Location Prior (RLP) and Rain Prior Injection Module (RPIM), which are the keys to revealing location information and boosting the performance of nighttime deraining.
- We demonstrate the state-of-the-art performance of our method on both synthetic and real night rainy scenes from rigorous experiments and ablation studies.

2. Related Work

In this section, we briefly review some existing deraining methods and image restoration backbones as well as existing deraining datasets.

Deraining methods. Recurrent and residual learning are common practices for deraining models. Yang *et al*. [36] proposed a multi-task network to jointly perform rain detection, estimation and removal in a recurrent manner. Fu *et al*. [8] proposed to predict the residue of rain streaks with high-frequency details as input. Li *et al*. [21] proposed Recurrent Squeeze-and-Excitation Context Aggregation Net (RESCAN) which is a recurrent network feeding output of the previous stage to the next one. Ren *et al*. [27] proposed PReNet with recurrent and residual design to balance performance and model complexity. Jiang *et al*. [16] proposed Multi-Scale Progressive Fusion Network (MSPFN) utilizing the multi-scale and recurrent strategy. Deng *et al*. [7] proposed to remove rain streaks and recover lost details with two parallel sub-networks. Yi *et al*. [37] proposed to protect structure information and guide network training with residue channel prior. Wang *et al*. [32] proposed SPatial Attentive Network (SPANet) which also utilizes residual blocks. Wang *et al*. [31] proposed Rain Convolutional Dictionary Network (RCDNet) to integrate the physical structure of residual rain streaks. Liang *et al*. [24] proposed a Deraining Recursive Transformer (DRT) with less parameters. In this work, we compare with some recent deraining models on night scenes.

(a) (b) (c)

Figure 2: Comparisons between synthetic and real rainy images. Synthetic images from (a) Rain100H [36], (b) GTAV-NightRain [42] and (c) real image from the Internet.

Image restoration backbones. Recently, powerful image restoration backbones [4, 5, 9] outperform previous methods on multiple tasks. Zamir *et al*. [40] proposed MPRNet making full use of encoder-decoder architecture and multi-stage strategies. They also proposed Restormer [39] to efficiently handle high-resolution images with the Transformer model. Wang *et al*. [34] also proposed a Transformer-based model following U-Net architecture and achieved promising performance on several tasks. While Tu *et al*. [30] proposed a multi-axis MLP-based architecture in a UNet-shaped hierarchical structure. Despite their robustness and capability, they fail to remove rain streaks in night scenes.

Deraining datasets. Most deraining datasets are synthesized by superposing the rain layer onto clean images. Li *et al*. [23] synthesized 12 rainy images following Garg and Shree [10, 11]. Yang *et al*. [36] proposed the Rain100L and Rain100H datasets. Fu *et al*. [8] synthesized 14000 rainy images using Photoshop. Zhang *et al*. [44] proposed the Rain800 dataset following their practice. Zhang and Patel [43] proposed the Rain1200 dataset with different density labels. Li *et al*. [18] proposed the NYU-Rain dataset and Hu *et al*. [15] proposed the RainCityscapes dataset based on depth information. Li *et al*. [19] proposed the MPID

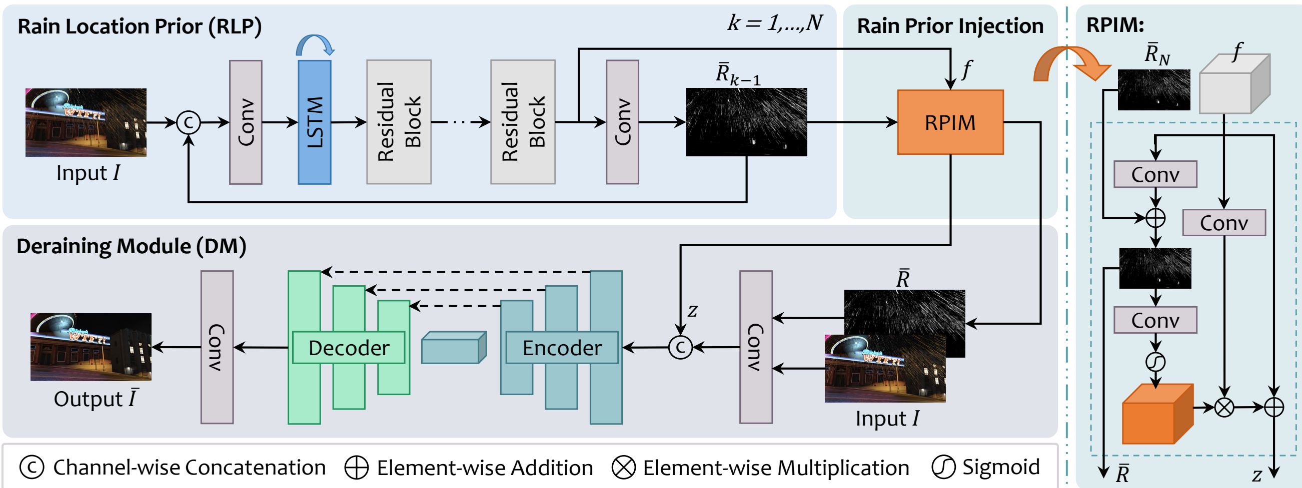


Figure 3: Illustration on the framework of our method for nighttime deraining, which consists of three parts. The Rain Location Prior (RLP) is the prior that we propose to learn from rainy images with recurrent residual model implicitly and indicates the location information of rain streaks. To further emphasize features within rain streaks areas, we propose the second part, namely Rain Prior Injection Module (RPIM), to modulate the RLP with attention mechanism. The final part is the deraining module, which consumes the original rainy input and the learned RLP to get boosted deraining performance.

dataset with different rain types as well as annotations for the detection task. Wang *et al*. [32] proposed the first real paired dataset, called SPA-data, using the semi-automatic method to generate clean images from rainy videos. Ba *et al*. [1] proposed GT-Rain dataset which contains real paired data. Recently, datasets turned their focus on night rainy scenes. Li *et al*. [20] proposed a real dataset with the ratio of night scenes to be around 50%. Zhang *et al*. [42] proposed a synthetic night rain dataset, namely GTAV-NightRain, using the game engine for rendering. The emergence of these datasets enables careful study on nighttime deraining.

3. Method

In this section, we first introduce the motivation. Then we describe the Rain Location Prior (RLP) and the Rain Prior Injection Module (RPIM) in detail. The framework of our method is illustrated in Figure 3.

3.1. Formulation and Motivation

The appearance and photometry of rain have been extensively studied for decades [10, 11]. However, most existing synthetic datasets [8, 23, 36, 43] adopt a simplified formulation of rain streaks, where the rainy image I is modeled as the sum of a rain streak layer R and a clean image B:

$$I = R + B. \tag{1}$$

A common strategy for rain removal is to model rain streaks with specific properties like direction, scale and space distribution and separate the rain layer from the background. This is usually achieved [8, 16, 21, 31, 36] by estimating the rain layer as an intermediate output before the final module and subtract it to restore the clean image.

However, the importance of location information embedded in the rain layer has been underestimated. As discussed in Section 1, since rain location is beneficial for rain removal and night rain appearance varies with spatial locations, obtaining the rain layer in advance is more necessary than in day scenes. Moreover, discriminating the rain layer from the night scene is more difficult due to the coupling of rain appearance and light sources, thus aforementioned practice (*i.e*., estimating the rain layer and subtracting it to get the final output using the same model) is inefficient.

In this work, therefore, we propose a novel framework for nighttime deraining, encompassing a model specializing the rain layer estimation, and a module that takes full advantage of the information of rain location. Specifically, we decouple the rain layer estimation and clean background reconstruction by introducing the Rain Location Prior (RLP) and Rain Prior Injection Module (RPIM). We learn to mine the information of rain location from nighttime rainy image and restore the clean scene respectively in an end-to-end manner. We further emphasize the features within rain regions by modulating them before rain revmoal and background reconstruction. In this way, useful information of rain location can be better extracted and exploited, boosting the performance of nighttime deraining.

3.2. Rain Location Prior (RLP)

As mentioned above, the rain layer is commonly assumed to be superposed onto the background, which facilitates the design ideas of progressive and residual learning for many deraining models [8, 16, 27, 32, 37]. These models typically regard the rain layer as a residual component to the clean image, and some of them learn to progressively model these high-frequency patterns with recurrent structures. These designs, in turn, can reveal location information of rain streaks in night scenes, where the appearance of rain highly depends on its spatial location. For deraining, it would be beneficial if accurate rain locations were provided in advance, which is impractical in real scenes. Therefore, we propose to use recurrent residual architecture to mine the prior knowledge of rain location before removing rain streaks. It can provide the location information to deraining models, thereby further boosting their performance, which highlights its importance for nighttime deraining.

In our proposed formulation, we introduce a module to learn and capture the prior knowledge of rain location, namely Rain Location Prior (RLP), before removing rain streaks. This module is composed of several residual blocks, which are effective in modeling rain streaks as a residual component. Meanwhile, it learns to progressively capture the rain through a recurrent design that utilizes the high-frequency characteristics of rain. When fed with the rainy input and an initial prior map, the module learns to extract deep features and recurrently update the prior map. The process of RLP can be formulated as follows:

$$\bar{R}_k = \mathcal{F}_{RLP}(I \odot \bar{R}_{k-1}; \theta), \quad k = 1, ..., N, \tag{2}$$

where I denotes the input and $\bar{R}_k$ is the RLP output at the k-th stage. $\bar{R}_0$ is the initial map and all values are set to 0.5. $\mathcal{F}_{RLP}(\circ; \theta)$ refers to the RLP module and θ is its parameters. $\odot$ denotes the channel-wise concatenation. N denotes the total number of recurrent stages, which is empirically set to 6 and 4 for CNN-based and Transformer-based models, respectively. Concretely, PReNet [27] is a lightweight deraining network with the residual and recurrent design for a balance between complexity and performance. Its simplicity in design tightly fits the role of the RLP module and is adopted in our experiment.

Note that we do not impose additional constraints on the RLP output, because we found that supervision signals from rain streak masks obtained by thresholding following the practice in raindrop removal [14, 26, 38] do not improve the performance. For simplicity, we do not apply regularization terms to the RLP output during training and encourage the module to learn it implicitly from training data. Corresponding experimental results are provided in the supplementary material.

3.3. Rain Prior Injection Module (RPIM)

In night scenes, the appearance of rain heavily depends on its spatial position relative to light sources. Therefore, deraining models should treat different areas with varying levels of importance. With the RLP indicating location information of rain streaks, we further emphasize the importance of these areas with the attention mechanism [40].

Specifically, we propose the Rain Prior Injection Module (RPIM). It takes the deep features extracted by the RLP module and further updates the prior map in a residual manner. The updated prior map is then used to generate the attention weights that increase the importance of rain streak areas and suppress other irrelevant areas. Finally, the features within rain streak areas are emphasized by the weights by element-wise multiplication of deep features and the attention weights. The RPIM can be described as follows:

$$\bar{R}, z = \mathcal{M}_{RPIM}(\bar{R}_N, f; \sigma), \tag{3}$$

where $\bar{R}_N$ and f are the final prior map and deep features extracted by residual blocks in the RLP module. $\mathcal{M}_{RPIM}(\circ; \sigma)$ denotes our Rain Prior Injection Module. $\bar{R}$ and z are the modulated output and feature tensor ready for injection into the deraining network, respectively.

3.4. Deraining Module (DM)

For pixel-wise image restoration tasks, U-shaped architectures with skip connections are commonly adopted as they have shown good capability in capturing contextual information without losing much spatial information [30, 34, 40]. CNN-based and Transformer-based models with such architecture are two main streams and many methods make further improvements based on them.

In Encoder-Decoder models, the input is typically projected into a feature space through a convolution layer, which is suitable for prior injection. After obtaining RLP and modulating the prior, we inject it into deraining models to further enhance deraining performance. We feed the rainy input and the RLP into the deraining module, which is a U-shaped model (either CNN-based or Transformer-based) for rain streak removal and background restoration. The process of this stage can be formulated as follows:

$$\bar{I} = \mathcal{F}_{DM}(I \odot \bar{R}, z; \Theta), \tag{4}$$

where $\bar{I}$ is the restored image by deraining module, $\bar{R}$ is the updated RLP and $\odot$ denotes the channel-wise concatenation. $\mathcal{F}_{DM}(\circ; \Theta)$ represents the deraining model, which focuses on the contextual information of neighboring pixels, and Θ is its parameters to be optimized. z refers to the modulated prior by RPIM and is injected into the deraining module by channel-wise concatenation with the projected feature after the first convolution layer.

For the training of the whole model, we apply the Charbonnier loss [3] on the final output:

$$\mathcal{L} = \sqrt{\left\|\bar{I} - Y\right\|^2 + \varepsilon^2}, \tag{5}$$

where $\bar{I}$ is the final output of the deraining model, Y denotes the clean ground truth and $\varepsilon = 10^{-3}$ is a constant.

4. Experiments

In this section, we first introduce the experimental setups. Then we provide quantitative and qualitative results on synthetic data and evaluations on real night rainy images, to show the superior performance of our method compared to existing deraining methods, under the nighttime deraining setting. Moreover, we conduct ablation studies to validate the effectiveness of the proposed modules. Finally, we conduct experiments to show the the generalization of our method on daytime datasets. Due to limited space, we provide more visual results in the supplementary material.

4.1. Experimental Setups

Datasets. Rain streaks in GTAV-NightRain dataset [42] are sparse with low difficulty, which is unfavorable for observing the difference since all methods behaves well. Thus we follow the instructions to render more difficult data (27dB *v.s.* 32dB in [42]) for experiments, by increasing the scale and density of rain streaks. We turn on and off the rain effect while keeping other settings unchanged for the same scene to render the rainy and clean image pairs. Please refer to [42] for more details on data collection. We collect 10 rainy images and 1 clean image for each scene and cover non-overlapping scenes for training and test sets. Finally, we get a total of 5,000 pairs of training data and 500 pairs of test data. Our collected data will be released with the code. Additionally, we conduct experiments on Rain13k [16, 40] and GT-Rain [1] datasets, to further show the generalization of our method to daytime data. We train our model with different modules on Rain13k and evaluate on its test sets, which are Test100 [44], Rain100H [36], Rain100L [36], Test2800 [8], Test1200 [43]. We also train our method on GT-Rain and compare it with other methods on its test set.

Implementation Details. Our method is implemented in PyTorch and trained with Adam optimizer [17] with default parameters. The initial learning rate is set to 2×10^{-4} and finally decreases to 1×10^{-6} following a cosine scheduler. The model is trained for 250 epochs with a batch size of 8 on 256×256 patches. Training is performed on a single NVIDIA RTX 3090 GPU. We adopt U-Net and Uformer (the Uformer-T variant) for CNN-based and Transformer-based architectures. All experiments are performed on a single NVIDIA RTX 3090 GPU.

Compared Methods. To show the effectiveness of our method, we make comparisons with PReNet [27], SPANet [32], DRDNet [7], RCDNet [31], SPDNet [37], MPRNet [40], GT-Rain [1], DRT [24], U-Net [26] and Uformer [34] (the Uformer-T variant) on both synthetic and real data. All methods are trained on synthetic data from scratch with publicly available codes provided by authors following their default settings. Among them, representative methods are selected for visual comparision.

Metrics. For quantitative evaluation, we adopt commonly used PSNR and SSIM [33] for comparison using the Y channel (in YCbCr color space) following [16].

Table 1: Quantitative results of compared methods on synthetic data. All the models are trained on our collected data following the default setting by authors. Higher metric is better for PSNR and SSIM. #GMACs is calculated on 256×256 patch.

Method	PSNR	SSIM	#Params	#GMACs
Input	27.10	0.8237	/	/
PReNet [27]	34.77	0.9622	0.17M	66.4G
SPANet [32]	31.10	0.9245	0.28M	36.3G
DRDNet [7]	32.20	0.9214	5.23M	689.8G
RCDNet [31]	34.17	0.9493	3.16M	195.1G
SPDNet [37]	33.90	0.9384	3.32M	96.6G
MPRNet [40]	36.63	0.9661	3.63M	548.7G
GT-Rain [1]	34.10	0.9572	2.29M	29.6G
DRT [24]	33.81	0.9381	1.18M	165.4G
U-Net [26]	36.63	0.9693	6.06M	45.3G
Uformer [34]	<u>37.45</u>	<u>0.9720</u>	5.20M	10.7G
Ours (U-Net)	37.28	0.9716	6.33M	117.8G
Ours (Uformer)	**38.44**	**0.9749**	5.52M	63.4G

4.2. Quantitative Results on Synthetic Data

As shown in Table 1, the performance of both SPANet [32] and DRDNet [7] is inferior to that of other competitors, even though there exists large difference between their model size and computational cost. While PReNet [27], RCDNet [31], SPDNet [37], GT-Rain [1] and DRT [24] perform better than the former two methods and their performance is comparable to others. Among all compared methods, MPRNet [40] and Uformer [34], as representative CNN-based and Transformer-based image restoration models, show superior performance to other competitors, which well validates their model capacity.

However, with the help of our proposed RLP and RPIM, the performance of both CNN-based and Transformer-based models can be boosted by a large margin. For CNN-based model U-Net [26], our method can get an increase of 0.65dB on PSNR and surpasses MPRNet [40] by 0.65dB. For the Transformer-based model Uformer [34], a large gain of 0.99dB on PSNR can be obtained and our proposed method outperforms all compared methods. These results can well validate the effectiveness of our proposed method.

4.3. Qualitative Results on Synthetic Data

As shown in Figure 4, SPANet [32] and DRDNet [7] can only remove a few rain streaks in night scenes and large amounts of rain streaks remain, which corresponds to inferior performance in quantitative results. While PReNet [27], RCDNet [31] and SPDNet [37] can remove more rain streaks than the former two competitors but there are still some rain streaks left there. MPRNet [40], U-Net [26] and Uformer [34] perform much better than these methods but

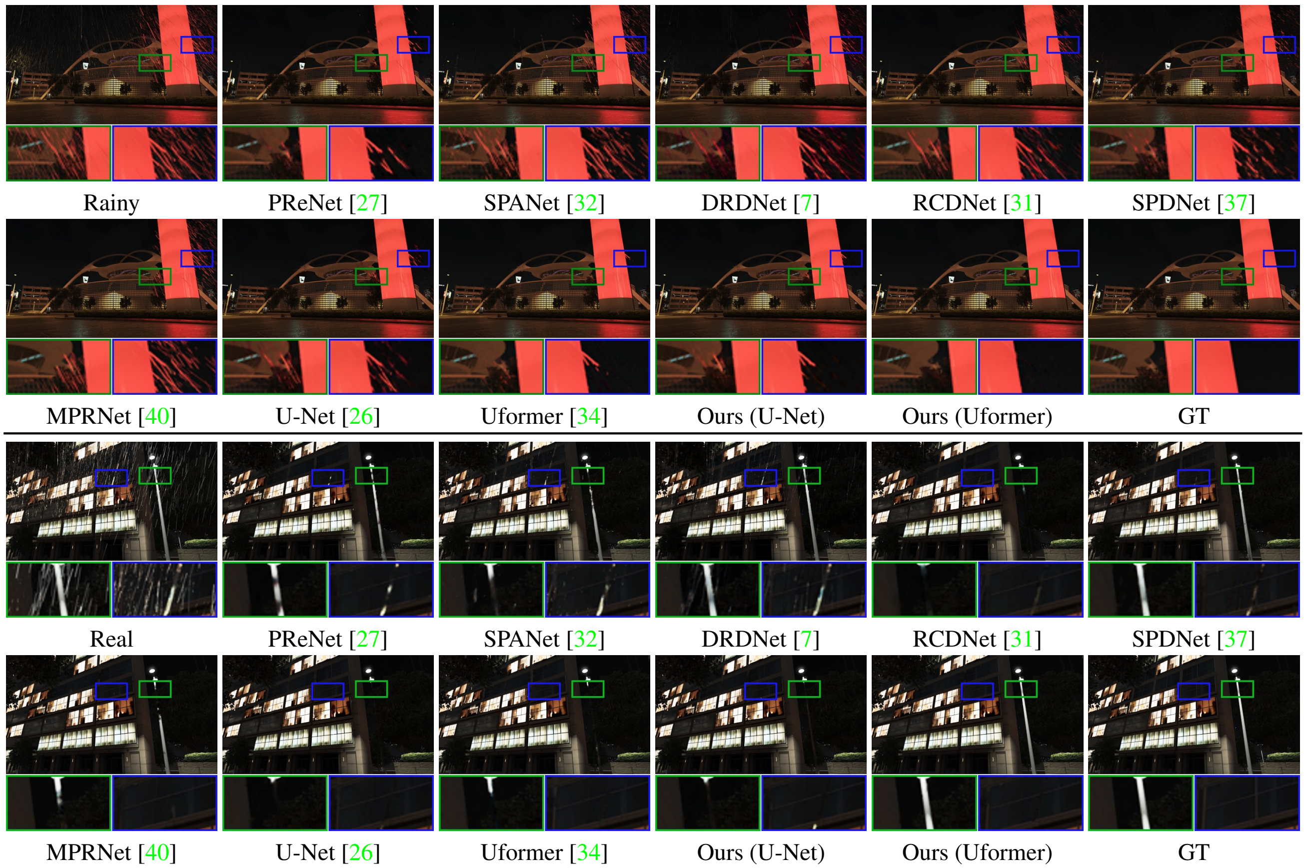


Figure 4: Qualitative results on synthetic data (gamma correction is applied for better visualization). SPANet and DRDNet can only remove a few rain streaks while PReNet, RCDNet and SPDNet behave better than the former two methods. MPRNet, U-Net and Uformer outperform other competitors but our proposed prior can boost the performance of U-Net and Uformer even further, in terms of removing more rain streaks (Uformer for the first scene) and preserving background details (Uformer for the second scene). These results demonstrate the effectiveness of our proposed method.

also face challenges for different scenes. Among all compared methods, Ours (Uformer) can get the best results.

In the first image, the illumination is dominated by the red neon cylinder and rain streaks look red. MPRNet [40] surprisingly fails the case and its result is similar to that of SPANet [32] and DRDNet [7]. For U-Net [26], Uformer [34] and Ours (U-Net), there are only a few rain streaks left on the image. However, Ours (Uformer) can fully remove the rain streaks, which demonstrates the effectiveness of our method. In the second image, bright windows consist of similar textures to rain streaks and the scene is relatively complex. We can find that almost all compared methods including Ours (U-Net) fail to preserve the texture of streetlight and it even totally disappears in some results. DRDNet [7] and SPDNet [37] keep the streetlight unchanged but there are some rain streaks left. Only Ours (Uformer) remove most rain streaks and preserve the texture of streetlight at the same time. We can infer that deraining models tend to recognize long thin textures with high pixel intensity as rain streaks in night scenes because rain streaks near light sources have higher pixel values. In such cases, location priors of rain streaks become more crucial for deraining models to accurately handle rain streaks in night scenes, which further confirms the effectiveness of our method.

4.4. Qualitative Results on Real Night Rainy Images

Here, we present the qualitative results on real data in Figure 5. Due to the domain gap between synthetic and real data, models trained with GTAV-NightRain still behave poorly in real night scenes [42].To further handle real night rainy images, we additionally add JPEG compression as a data augmentation in our training. Compared methods can partially remove rain streaks in real night rainy images but behave inconsistently for different rainy images. We can see that DRDNet [7], RCDNet [31] and SPDNet [37] perform better than PReNet [27], SPANet [32] and MPRNet [40] for the first image while U-Net [26] and Uformer [34] get better results. But they all fail to remove rain streaks in the second image. However, our method succeeds in handling both images. It can remove most rain streaks while keep-

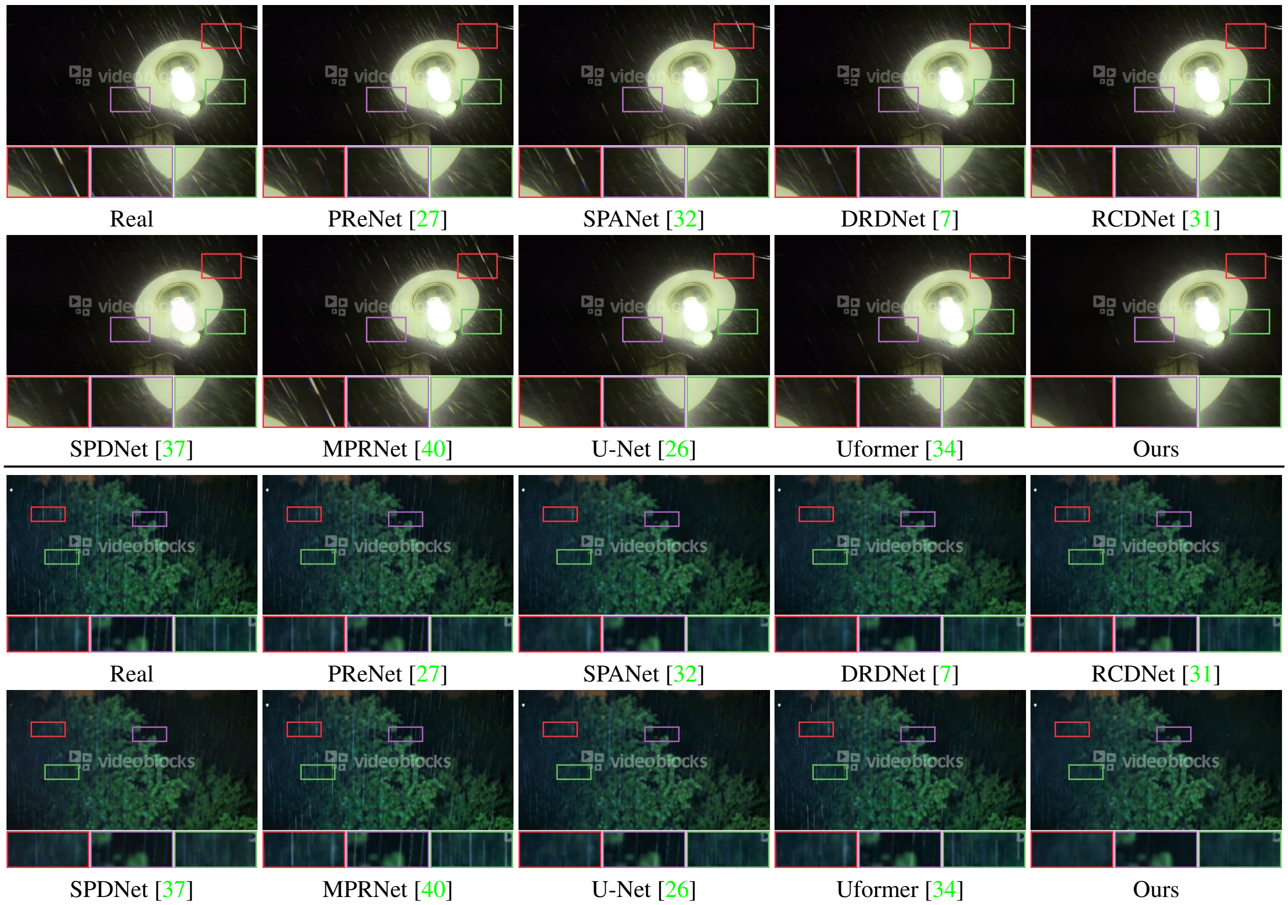


Figure 5: Qualitative results on real night rainy images. Our method gets the best visual results. Please zoom in for details.

ing backgrounds tidy, showing the effectiveness of our proposed method for removing rain streaks in real night scenes.

4.5. Qualitative Results of RLP

To further illustrate the effectiveness of our Rain Location Prior, we provide visualizations on both synthetic and real night rainy images in Figure 6. Our RLP module can learn to recognize rain streaks and emphasize these areas with higher weights. Notably, RLP can learn by itself to suppress areas of light sources, *i.e.*, these areas are much darker than neighboring areas of rain streaks. We can also notice that rain streaks in the sky can also be recognized even though they are not obvious in the rainy image.

As shown in the right half of Figure 6, our RLP module can also generalize well to real night rainy images with only synthetic data for training. Same to the synthetic case, light sources are suppressed in RLP and those areas are darker than areas of rain streaks. These results all show the effectiveness of RLP, which is beneficial for nighttime deraining.

4.6. Ablation Studies

In this section, we study the effect of each component of our method, including Rain Location Prior (RLP) and

Table 2: Ablation study on individual components of our proposed method & different model structures and sizes.

DM	RLP	RPIM	PSNR	Δ	SSIM	Δ
PReNet	✗	✗	34.77	-	0.9622	-
	✓	✗	35.00	**+0.23**	0.9649	**+0.0027**
	✓	✓	34.31	**-0.46**	0.9621	**-0.0001**
MPRNet	✗	✗	35.21	-	0.9627	-
	✓	✗	35.42	**+0.21**	0.9641	**+0.0014**
	✓	✓	35.67	**+0.46**	0.9633	**+0.0006**
U-Net	✗	✗	36.63	-	0.9693	-
	✓	✗	37.08	**+0.45**	0.9715	**+0.0022**
	✓	✓	37.28	**+0.65**	0.9716	**+0.0023**
Uformer-T	✗	✗	37.45	-	0.9720	-
	✓	✗	37.95	**+0.50**	0.9733	**+0.0013**
	✓	✓	38.44	**+0.99**	0.9749	**+0.0029**
Uformer-B	✗	✗	39.41	-	0.9763	-
	✓	✗	39.65	**+0.24**	0.9770	**+0.0007**
	✓	✓	39.77	**+0.36**	0.9773	**+0.0010**

Rain Prior Injection Module (RPIM). We choose U-Net [26] and Uformer [34] out of CNN-based and Transformer-based models as the deraining module in our method, with U-shaped Encoder-Decoder architecture. Besides, we examine the effects of RLP on PReNet [27] and MPRNet

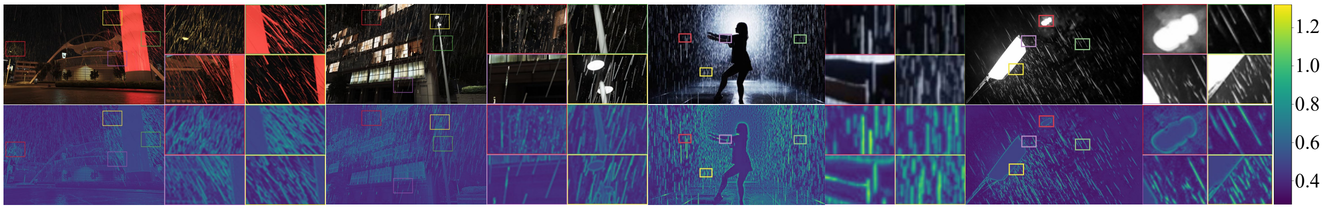


Figure 6: Visualizations of Rain Location Prior (RLP) on synthetic and real night rainy images. Please zoom in for details.

Table 3: Generalization to daytime datasets. Models are trained on Rain13k dataset [16] and evaluated on five test sets.

DM	RLP	RPIM	Test100 [44]		Rain100H [36]		Rain100L [36]		Test2800 [8]		Test1200 [43]		Avg.	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Input			22.55	0.6863	13.55	0.3784	26.90	0.8383	24.35	0.7829	23.63	0.7324	22.20	0.6837
U-Net	✗	✗	24.91	0.8584	24.21	0.8267	30.52	0.9256	31.31	0.9199	31.01	0.9026	28.39	0.8866
U-Net	✓	✗	25.04	0.8550	26.82	0.8467	30.56	0.9185	31.32	0.9176	30.71	0.8917	28.89	0.8859
U-Net	✓	✓	25.32	0.8665	26.06	0.8380	31.20	0.9324	31.32	0.9203	31.59	0.9089	**29.10**	**0.8932**
Uformer	✗	✗	28.01	0.8780	28.21	0.8516	32.16	0.9344	32.43	0.9287	31.66	0.9078	30.50	0.9001
Uformer	✓	✗	28.08	0.8775	28.37	0.8545	32.59	0.9385	32.49	0.9287	31.86	0.9093	30.68	0.9017
Uformer	✓	✓	28.50	0.8821	28.34	0.8576	32.67	0.9390	32.55	0.9295	31.67	0.9042	**30.74**	**0.9025**

[40] with other architecture designs. Finally, we apply our method to the Uformer-B variant to investigate RLP's effect on model with more parameters. Experiments are carried out on the same test set and results are listed in Table 2.

RLP. With the prior map learned by the RLP module, directly feeding it into the deraining module brings an increase of 0.23dB, 0.21dB, 0.45dB, 0.50dB and 0.24dB in PSNR for five different deraining modules. It shows the effectiveness of the proposed RLP which can reveal location information of rain streaks and improve the deraining performance. The benefit of accurate location is straightforward and it is also effective for PReNet and MPRNet with other model structures.

RPIM. In night scenes where location information plays a vital role in deraining, emphasizing the importance of RLP in rain streak areas is beneficial. As shown in Table 2, the performance of deraining modules can be further boosted with the help of RPIM. MPRNet [40], U-Net [26] and Uformer-T/-B [34] can get an additional increase of 0.25dB, 0.20dB and 0.49dB/0.12dB in PSNR. While PReNet [27] gets worse results and the repeated feature injections in its recurrent structure may leads to the performance drop.

Model Structure and Capability. With different models compared, we can find that (i) recurrent structure may suffer from repeated feature injection (*e.g.*, PReNet); (ii) models with vanilla Encoder-Decoder structure may benefit more from RLP (*e.g.*, U-Net *v.s.* MPRNet); (iii) moderate models with better capability may benefit more from RLP (Uformer *v.s.* U-Net); (iv) moderate models may prefer addtional knowledge from RLP than big models (*e.g.*, Uformer-B *v.s.* Uformer-T). The comparisons demonstrate the effectiveness of our proposed RLP and RPIM and further provide insights on further improvement.

4.7. Generalization to Daytime Datasets

Despite our goal of nighttime deraining, we also conduct experiments on Rain13k [16] and GT-Rain [1] to demonstrate the generalization of our method on day scenes. As listed in Table 3, our method is also beneficial for CNN-based and Transformer-based models on daytime deraining. Additionally, we can find that the improvement is smaller than that on the nighttime dataset, which suggests the greater importance of location information for nighttime deraining. We also test our method on the GT-Rain dataset [1] and results are provided in the supplementary material.

5. Conclusion

Nighttime rain removal is a challenging task due to complex illumination conditions. In night scenes, the location information of rain streaks becomes more critical for rain removal. In this paper, we propose the Rain Location Prior (RLP), which can be learned implicitly from rainy images using a recurrent residual model to reveal the location information of rain streaks. To further improve the deraining performance, we propose the Rain Prior Injection Module (RPIM) to modulate the learned RLP with the attention mechanism and emphasize the features within rain streak areas. Together, they help deraining modules focus more on recovering clean background rather than recognizing complex rain streaks. Despite the original goal of nighttime deraining, our method also works in day scenes. Experimental results show the effectiveness and generalization of our method on both synthetic data and real nighttime rainy images, which may be helpful for future research on nighttime deraining. We plan to explore more architectures for learning and extending RLP in the future.

Acknowledgments This work was supported by the National Natural Science Foundation of China (62171038, 62088101, 61931008, and 62006101), the R&D Program of Beijing Municipal Education Commission (KZ202211417048), and the Fundamental Research Funds for the Central Universities.

References

[1] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Celso M de Melo, Suya You, Stefano Soatto, Alex Wong, et al. Not just streaks: Towards ground truth for single image deraining. In *ECCV*, 2022. 3, 5, 8

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 1

[3] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 4

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022. 2

[5] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *CVPRW*, 2021. 2

[6] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE TPAMI*, 2022. 1

[7] Sen Deng, Mingqiang Wei, Jun Wang, Yidan Feng, Luming Liang, Haoran Xie, Fu Lee Wang, and Meng Wang. Detail-recovery image deraining via context aggregation networks. In *CVPR*, 2020. 2, 5, 6, 7

[8] 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*, 2017. 2, 3, 5, 8

[9] Hu Gao and Depeng Dang. Mixed hierarchy network for image restoration. *arXiv preprint arXiv:2302.09554*, 2023. 2

[10] Kshitiz Garg and Shree K Nayar. Photorealistic rendering of rain streaks. *ACM TOG*, 25(3):996–1002, 2006. 2, 3

[11] Kshitiz Garg and Shree K Nayar. Vision and rain. *IJCV*, 75(1):3–27, 2007. 2, 3

[12] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2):982–993, 2016. 1

[13] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *ICCV*, 2019. 1

[14] Zhixiang Hao, Shaodi You, Yu Li, Kunming Li, and Feng Lu. Learning from synthetic photorealistic raindrop for single image raindrop removal. In *ICCVW*, 2019. 2, 4

[15] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *CVPR*, 2019. 2

[16] 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*, 2020. 1, 2, 3, 5, 8

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *CVPR*, 2019. 2

[19] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *CVPR*, 2019. 2

[20] Wei Li, Qiming Zhang, Jing Zhang, Zhen Huang, Xinmei Tian, and Dacheng Tao. Toward real-world single image deraining: A new benchmark and beyond. *arXiv preprint arXiv:2206.05514*, 2022. 1, 3

[21] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 2018. 2, 3

[22] Yu Li, Robby T Tan, and Michael S Brown. Nighttime haze removal with glow and multiple light colors. In *ICCV*, 2015. 1

[23] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016. 2, 3

[24] Yuanchu Liang, Saeed Anwar, and Yang Liu. Drt: A lightweight single image deraining recursive transformer. In *CVPRW*, 2022. 2, 5

[25] Feifan Lv, Yu Li, and Feng Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset. *IJCV*, 129(7):2175–2193, 2021. 1

[26] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, 2018. 2, 4, 5, 6, 7, 8

[27] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, 2019. 2, 3, 4, 5, 6, 7, 8

[28] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. *arXiv preprint arXiv:2207.14024*, 2022. 1

[29] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul De Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *IJCV*, 129(2):341–360, 2021. 1

[30] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *CVPR*, 2022. 2, 4

[31] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, 2020. 2, 3, 5, 6, 7

[32] 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*, 2019. 1, 2, 3, 5, 6, 7

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[34] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022. 1, 2, 4, 5, 6, 7, 8

[35] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020. 1

[36] 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*, 2017. 2, 3, 5, 8

[37] Qiaosi Yi, Juncheng Li, Qinyan Dai, Faming Fang, Guixu Zhang, and Tieyong Zeng. Structure-preserving deraining with residue channel prior guidance. In *ICCV*, 2021. 2, 3, 5, 6, 7

[38] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Raindrop detection and removal from long range trajectories. In *ACCV*, 2014. 4

[39] 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*, 2022. 1, 2

[40] 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*, 2021. 1, 2, 4, 5, 6, 7, 8

[41] Fan Zhang, Yu Li, Shaodi You, and Ying Fu. Learning temporal consistency for low light video enhancement from single images. In *CVPR*, 2021. 1

[42] Fan Zhang, Shaodi You, Yu Li, and Ying Fu. Gtav-nightrain: Photometric realistic large-scale dataset for night-time rain streak removal. *arXiv preprint arXiv:2210.04708*, 2022. 1, 2, 3, 5, 6

[43] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018. 2, 3, 5, 8

[44] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE TCSVT*, 30(11):3943–3956, 2019. 2, 5, 8

[45] Jing Zhang, Yang Cao, and Zengfu Wang. Nighttime haze removal based on a new imaging model. In *ICIP*, 2014. 1