

UConNet: Unsupervised Controllable Network for Image and Video Deraining

Junhao Zhuang

University of Electronic Science
and Technology of China
Chengdu, China
junhaozhuang@foxmail.com

Yisi Luo

University of Electronic Science
and Technology of China
Chengdu, China
yisiluo1221@foxmail.com

Xile Zhao*

University of Electronic Science
and Technology of China
Chengdu, China
xlzhao122003@163.com

Taixiang Jiang

Southwestern University of
Finance and Economics
Chengdu, China
taixiangjiang@gmail.com

Bichuan Guo

Tsinghua University
Beijing, China
guobichuan@gmail.com

ACM Reference Format:

Junhao Zhuang, Yisi Luo, Xile Zhao, Taixiang Jiang, and Bichuan Guo. 2022. UConNet: Unsupervised Controllable Network for Image and Video Deraining. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3503161.3547772>

0.1 Definition of the Derivative Operator ∇_θ

In this subsection, we give the definition of ∇_θ . We denote the i, j -th element of a matrix \mathbf{X} by X_{ij} . To clarify the definition of the derivative operator ∇_θ , we first introduce the vertical derivative operator ∇_y . Given a matrix $\mathbf{R} \in \mathbb{R}^{H \times W}$, the vertical derivative operator ∇_y is defined as $\nabla_y \mathbf{R} = \mathbf{H} \otimes \mathbf{R}$, where

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

is the vertical derivative convolutional kernel and \otimes is the convolutional operator defined as $(\mathbf{H} \otimes \mathbf{R})_{ij} = \sum_{r,s=1,2,3} \mathbf{H}_{rs} \mathbf{R}_{i-2+r, j-2+s}$ ¹. To capture rain streaks in different directions (other than just vertical), we introduce an affine operator $\tau_\theta(\cdot)$ (parameterized by θ) to modify the kernel \mathbf{H} , so that the new kernel $\tau_\theta(\mathbf{H}) \in \mathbb{R}^{3 \times 3}$ can extract the derivative value along the direction θ . The new kernel is defined by $(\tau_\theta(\mathbf{H}))_{ij} = I(\mathbf{H}, (i, j))$ ($i, j = 1, 2, 3$), where

$$\begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \quad (2)$$

*Corresponding author

¹Here, \mathbf{R} is padded with zeros to perform the convolution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '22, October 10–14, 2022, Lisboa, Portugal

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9203-7/22/10...\$15.00

<https://doi.org/10.1145/3503161.3547772>

and $I(\mathbf{H}, (\hat{i}, \hat{j}))$ returns the bilinear interpolation result of the matrix \mathbf{H} at the coordinate (\hat{i}, \hat{j}) ². Equipped with the new kernel $\tau_\theta(\mathbf{H})$, the derivative operator ∇_θ (parameterized by θ) is defined as $\nabla_\theta \mathbf{R} = (\tau_\theta(\mathbf{H})) \otimes \mathbf{R}$.

0.2 Details of the ADMM Algorithm

In this subsection, we clarify the ADMM algorithm we used to tackle the model

$$\min_{\mathbf{R}} \beta_1 \|\nabla_\theta \mathbf{R}\|_{\ell_1} + \beta_2 \|\mathbf{R}\|_{\ell_1} + \beta_3 \|\nabla_{\theta^T}(\mathbf{O} - \mathbf{R})\|_{\ell_1}. \quad (3)$$

Here, the rain direction θ is obtained via the rain direction estimation network. By introducing auxiliary variables $\{\mathbf{V}\}_{i=1}^3$, the problem is re-formulated as

$$\begin{aligned} \min_{\mathbf{R}} \beta_1 \|\mathbf{V}_1\|_{\ell_1} + \beta_2 \|\mathbf{V}_2\|_{\ell_1} + \beta_3 \|\mathbf{V}_3\|_{\ell_1} \\ \text{s.t. } \mathbf{V}_1 = \nabla_\theta \mathbf{R}, \mathbf{V}_2 = \mathbf{R}, \mathbf{V}_3 = \nabla_{\theta^T}(\mathbf{O} - \mathbf{R}). \end{aligned} \quad (4)$$

The corresponding augmented Lagrangian function is

$$\begin{aligned} L_\mu(\mathbf{R}, \mathbf{V}_i, \mathbf{D}_i) = & \beta_1 \|\mathbf{V}_1\|_{\ell_1} + \beta_2 \|\mathbf{V}_2\|_{\ell_1} + \beta_3 \|\mathbf{V}_3\|_{\ell_1} + \\ & \frac{\mu}{2} \|\nabla_\theta \mathbf{R} - \mathbf{V}_1 - \mathbf{D}_1\|_F^2 + \frac{\mu}{2} \|\mathbf{R} - \mathbf{V}_2 - \mathbf{D}_2\|_F^2 + \\ & \frac{\mu}{2} \|\nabla_{\theta^T}(\mathbf{O} - \mathbf{R}) - \mathbf{V}_3 - \mathbf{D}_3\|_F^2, \end{aligned} \quad (5)$$

where $\mathbf{D}_i = -\frac{\mu}{\Lambda_i}$ ($i = 1, 2, 3$) and $\{\Lambda_i\}_{i=1}^3$ are Lagrangian multipliers. μ is the penalty parameter. We can split the joint minimization problem into easier subproblems under the framework of ADMM.

V Sub-problems The \mathbf{V} sub-problems are

$$\begin{cases} \min_{\mathbf{V}_1} \frac{\mu}{2} \|\nabla_\theta \mathbf{R} - \mathbf{V}_1 - \mathbf{D}_1\|_F^2 + \beta_1 \|\mathbf{V}_1\|_{\ell_1} \\ \min_{\mathbf{V}_2} \frac{\mu}{2} \|\mathbf{R} - \mathbf{V}_2 - \mathbf{D}_2\|_F^2 + \beta_2 \|\mathbf{V}_2\|_{\ell_1} \\ \min_{\mathbf{V}_3} \frac{\mu}{2} \|\nabla_{\theta^T}(\mathbf{O} - \mathbf{R}) - \mathbf{V}_3 - \mathbf{D}_3\|_F^2 + \beta_3 \|\mathbf{V}_3\|_{\ell_1}, \end{cases} \quad (6)$$

²Here, \hat{i} and \hat{j} may not be integers. Thus, we use the bilinear interpolation to obtain the value of \mathbf{H} at the coordinate (\hat{i}, \hat{j}) .

which can be explicitly solved by

$$\begin{cases} \mathbf{V}_1 = \text{Soft}_{\frac{\beta_1}{\mu}}(\nabla_\theta \mathbf{R} - \mathbf{D}_1) \\ \mathbf{V}_2 = \text{Soft}_{\frac{\beta_2}{\mu}}(\mathbf{R} - \mathbf{D}_2) \\ \mathbf{V}_3 = \text{Soft}_{\frac{\beta_3}{\mu}}(\nabla_{\theta^T}(\mathbf{O} - \mathbf{R}) - \mathbf{D}_3). \end{cases} \quad (7)$$

Here, $\text{Soft}_\alpha(\cdot)$ denotes the soft-thresholding operator with the threshold α .

R Sub-problem The R sub-problem is

$$\min_{\mathbf{R}} \frac{\mu}{2} \|\nabla_\theta \mathbf{R} - \mathbf{V}_1 - \mathbf{D}_1\|_F^2 + \frac{\mu}{2} \|\mathbf{R} - \mathbf{V}_2 - \mathbf{D}_2\|_F^2 + \frac{\mu}{2} \|\nabla_{\theta^T}(\mathbf{O} - \mathbf{R}) - \mathbf{V}_3 - \mathbf{D}_3\|_F^2, \quad (8)$$

which can be explicitly solved by

$$\begin{aligned} \mathbf{R} = & F^{-1}((\overline{F(\tau_{\theta^T}(\mathbf{H}))} F(\tau_{\theta^T}(\mathbf{H})) F(\mathbf{O}) - \overline{F(\tau_{\theta^T}(\mathbf{H}))} F(\mathbf{V}_3 + \mathbf{D}_3) + \\ & \overline{F(\tau_\theta(\mathbf{H}))} F(\mathbf{V}_1 + \mathbf{D}_1) + F(\mathbf{V}_2 + \mathbf{D}_2)) / (\overline{F(\tau_{\theta^T}(\mathbf{H}))} F(\tau_{\theta^T}(\mathbf{H})) + \\ & 1 + \overline{F(\tau_\theta(\mathbf{H}))} F(\tau_\theta(\mathbf{H}))). \end{aligned} \quad (9)$$

Here, F denotes the Fourier transformation and F^{-1} denotes its inverse transformation. $\overline{\mathbf{X}}$ denotes the conjugate matrix of \mathbf{X} .

D Update The variables $\{\mathbf{D}_i\}_{i=1}^3$ are updated by

$$\begin{cases} \mathbf{D}_1 = \mathbf{D}_1 - \nabla_\theta \mathbf{R} + \mathbf{V}_1 \\ \mathbf{D}_2 = \mathbf{D}_2 - \mathbf{R} + \mathbf{V}_2 \\ \mathbf{D}_3 = \mathbf{D}_3 - \nabla_{\theta^T}(\mathbf{O} - \mathbf{R}) + \mathbf{V}_3. \end{cases} \quad (10)$$

0.3 Details of the Video Deraining Method

In this subsection, we introduce the details of our video deraining method UConNet-V. Compared with single image, videos have consistence along the temporal mode. To tackle the video deraining task, we only need to introduce an extra pre-trained optical flow network to help capture the temporal consistence. The overall flowchart of our video deraining method UConNet-V is illustrated in Fig. 1. Specifically, Given a rainy video, we denote its t -th frame by \mathbf{O}_t . The input of the network is \mathbf{O}_t and its adjacent rainy frames $\mathbf{O}_{t+1}, \mathbf{O}_{t+2}$ as well as the adjacent clean frames $\mathbf{B}_{t-1}, \mathbf{B}_{t-2}$ (Note that when we process the t -th frame, the clean frames \mathbf{B}_{t-1} and \mathbf{B}_{t-2} are already obtained). The output of the network is the rain map \mathbf{R}_t . As shown in Fig. 1, we align adjacent frames using the optical flow network [5] and concatenate the resulting aligned frames with the deraining result $\overline{\mathbf{B}}_t$ using the UConNet. Then, we take the median number and perform subtraction to obtain the estimated rain map \mathbf{R}_{E_t} . The following steps follow the standard UConNet introduced in the main paper, and some tiny differences between UConNet and UConNet-V can be easily checked in Fig. 1. The loss function for the t -th frame is

$$L = \beta_1 \|\nabla_\theta \mathbf{R}_t\|_{\ell_1} + \beta_2 \|\mathbf{R}_t\|_{\ell_1} + \beta_3 \|\nabla_{\theta^T}(\mathbf{O}_t - \mathbf{R}_t)\|_{\ell_1} + \beta_4 \|\mathbf{R}_{E_t} - \mathbf{R}_t\|_{\ell_1}, \quad (11)$$

where $\{\beta_i\}_{i=1}^4$ are sampled from uniform distributions. Then, the weightings $\{w_i\}_{i=1}^4$ are calculated by

$$w_i = \frac{(\beta_i/b_i)}{\sum_{j=1}^4 (\beta_j/b_j)} \quad (i = 1, 2, 3, 4). \quad (12)$$

Table 1: The results by UConNet on R100L dataset with estimated rain directions plus errors following the Gaussian distribution $N(0, \sigma)$, where σ denotes the standard deviation.

Metric	Rainy	$\theta = P_\eta(\mathbf{O})$	$\theta + N(0, 2)$	$\theta + N(0, 5)$	$\theta + N(0, 10)$
PSNR	25.832	31.28	31.22	30.75	29.64
SSIM	0.8300	0.9351	0.9337	0.9295	0.9081

In the loss function (11), the first three terms directly follow the loss function of UConNet. The fourth term enforces the output rain map \mathbf{R}_t to be similar to the estimated rain map \mathbf{R}_{E_t} . We collect 25 rainy videos in the NTURain dataset [2] as the training data, where each frame of these videos is viewed as a training sample. Other compared video deraining methods are also trained using the same datasets, but with additional clean videos as supervision. Our method is unsupervised and does not need clean videos.

At the inference stage, the four weightings $\{w_i\}_{i=1}^4$ are controllable. We extra learn a content-aware weightings recommendation network to suggest proper weightings for each frame of the rainy video. The controllability brings high generalization abilities for different rain scenarios and thus our UConNet-V achieves state-of-the-art performances for different datasets (see Fig. 6 and Table 4 in the main paper).

0.4 Image Deraining with Multiple Rain Directions

Although our UConNet was designed for image deraining with a single rain direction θ , it can be easily applied to image deraining with multiple rain directions by feeding the rainy image multiple times into the deraining network. At each time, we could select different rain directions θ to remove rain streaks in that direction (see Fig. 2). In this way, our UConNet can handle rainy images with multiple and different rain directions.

0.5 Robustness of UConNet to Rain Direction Estimation

In this section, we discuss the robustness of our UConNet to the rain direction estimation. Specifically, we perform Gaussian noise $N(0, \sigma)$, where σ denotes the standard deviation, on the estimated rain direction θ and feed it into the main UConNet for deraining. The quantitative results with different levels of noise are shown in Table 1. We can observe that our method is robust to the noise as the margin between the results of different methods in Table 1 is small, which ensures the applicability of our method in real scenarios with irregular rain streaks.

0.6 More Experimental Results

We display more visual results on real-world rainy images in Fig. 3–Fig. 5. The superiority of our method over compared methods is demonstrated, as our method can both remove the rain streaks and preserve the image details well.

REFERENCES

- [1] Chenghao Chen and Hao Li. 2021. Robust Representation Learning with Feedback for Single Image Deraining. In *CVPR*. 7738–7747.
- [2] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. 2018. Robust Video Content Alignment and Compensation for Rain Removal in a CNN Framework. In *CVPR*. 6286–6295.

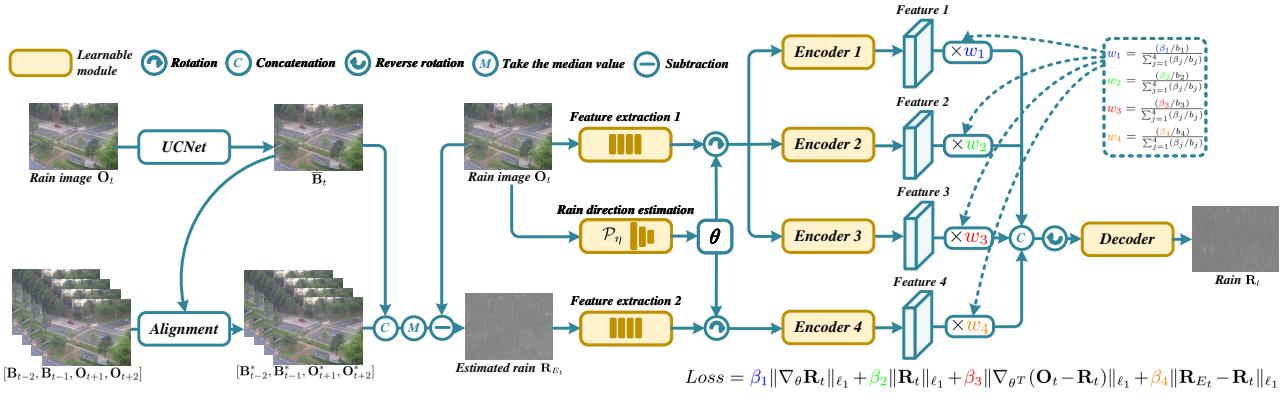


Figure 1: The flowchart of our UConNet-V for video deraining. Based on UConNet, we introduce an optical flow network [5] to align adjacent frames. The weightings $\{w_i\}_{i=1}^4$ are controllable at the inference stage to achieve high generalization abilities.



Figure 2: The deraining results for a rainy image with multiple rain directions. We feed the image twice into the deraining network, where each time we select different rain direction θ to remove the rain streaks.

- [3] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. 2018. Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining. In *ECCV*. 262–277.
- [4] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. 2019. Progressive Image Deraining Networks: A Better and Simpler Baseline. In *CVPR*. 3932–3941.
- [5] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*. 402–419.
- [6] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. 2019. Semi-Supervised Transfer Learning for Image Rain Removal. In *CVPR*. 3872–3881.
- [7] Changfeng Yu, Yi Chang, Yi Li, Xile Zhao, and Luxin Yan. 2021. Unsupervised Image Deraining: Optimization Model Driven Deep CNN. In *ACM MM*. 2634–2642.



Figure 3: The deraining results by different methods on real-world rainy images. Please increase screen brightness.

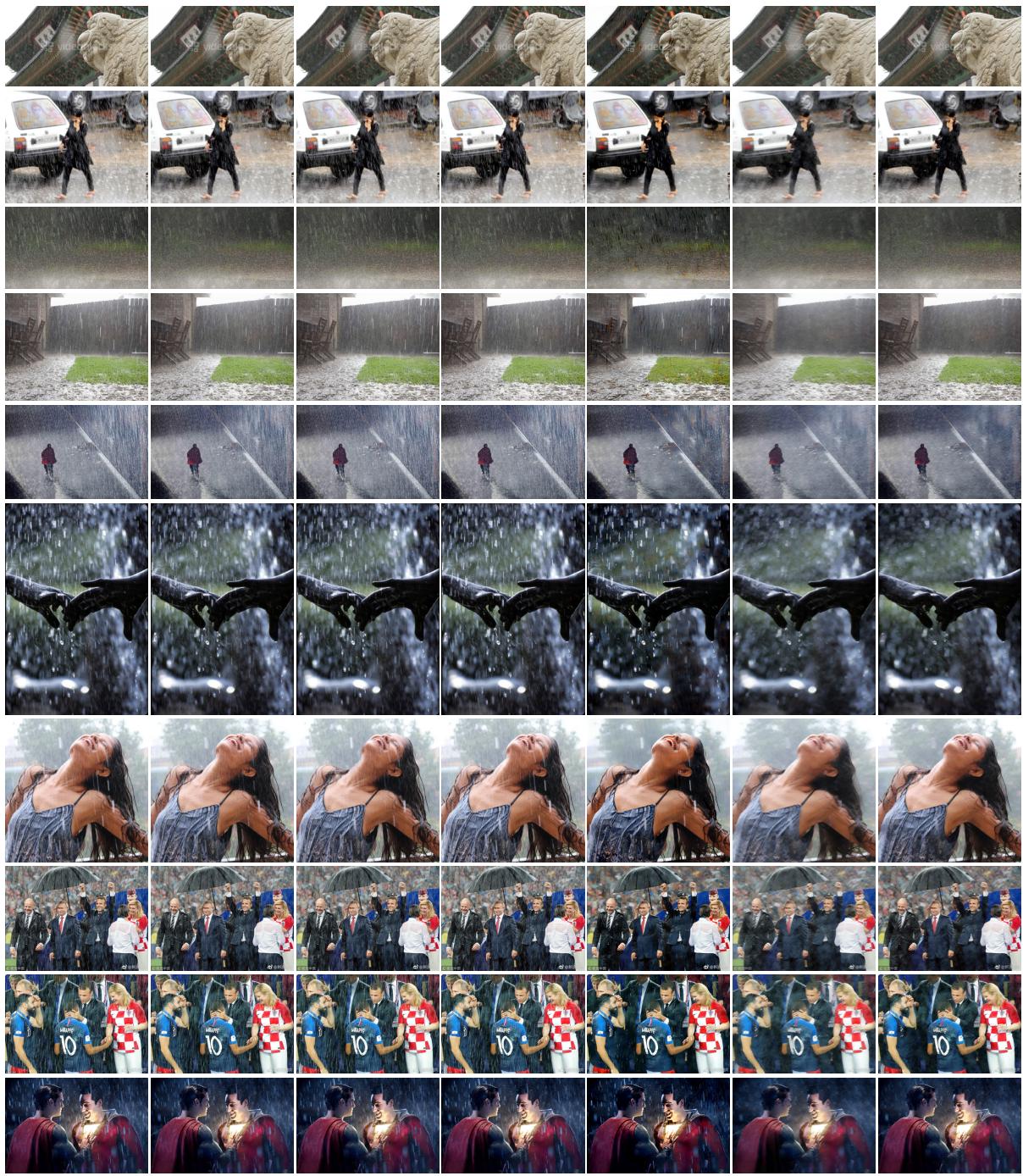


Figure 4: The deraining results by different methods on real-world rainy images. Please increase screen brightness.

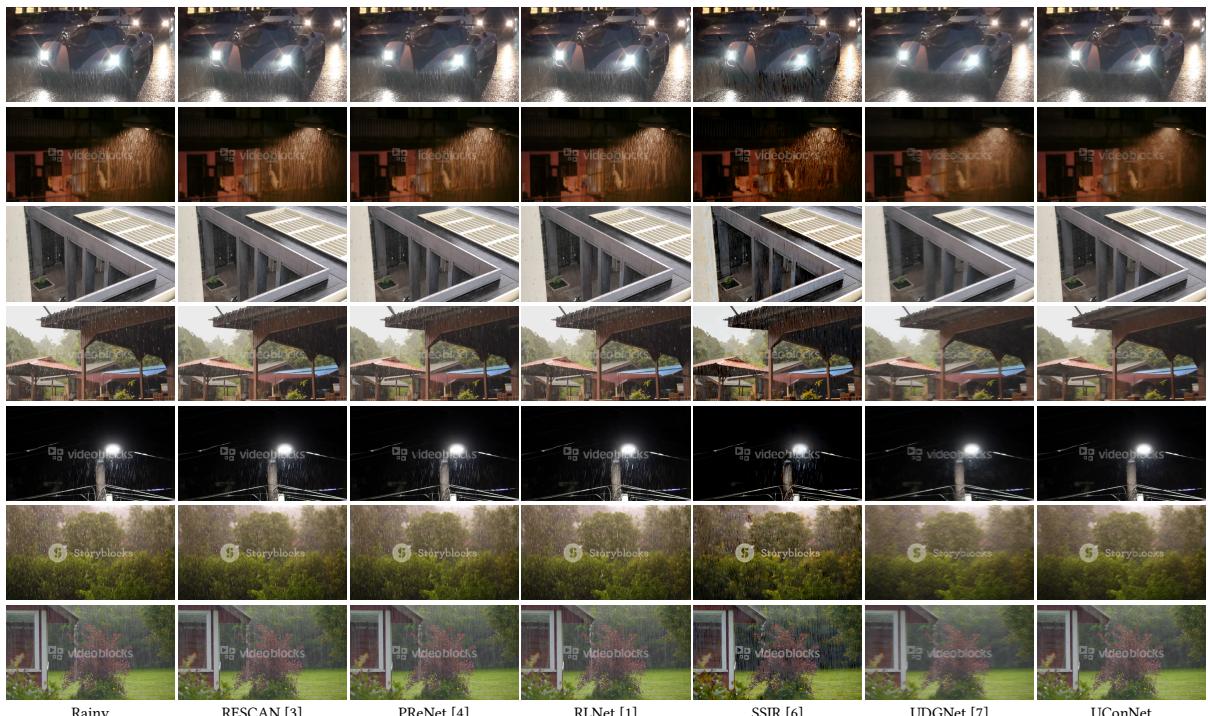


Figure 5: The deraining results by different methods on real-world rainy images. Please increase screen brightness.