Single Image Deraining via Recurrent Hierarchy Enhancement Network

Youzhao Yang, Hong Lu
{yzyang17,honglu}@fudan.edu.cn
Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University, Shanghai, China

ABSTRACT

Single image deraining is an important problem in many computer vision tasks since rain streaks can severely hamper and degrade the visibility of images. In this paper, we propose a novel network named **Re**current **H**ierarchy **E**nhancement **N**etwork (ReHEN) to remove rain streaks from rainy images stage by stage. Unlike previous deep convolutional network methods, we adopt a **H**ierarchy **E**nhancement **U**nit (HEU) to fully extract local hierarchical features and generate effective features. Then a **R**ecurrent **E**nhancement **U**nit (REU) is added to keep the useful information from HEU and benefit the rain removal in the later stages. To focus on different scales, shapes, and densities of rain streaks adaptively, **S**queeze-and-**E**xcitation (SE) block is applied in both HEU and REU to assign different scale factors to high-level features. Experiments on five synthetic datasets and a real-world rainy image set show that the proposed method outperforms the state-of-the-art methods considerably. The source code is available at https://github.com/nnUyi/ReHEN*.

CCS CONCEPTS

• **Computing methodologies → Image processing**.

KEYWORDS

single image deraining; recurrent network; hierarchy features; low-level vision

ACM Reference Format:
Youzhao Yang, Hong Lu. 2019. Single Image Deraining via Recurrent Hierarchy Enhancement Network. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351149

1 INTRODUCTION

Many outdoor computer vision systems like video surveillance, visual navigation, object tracking, as well as self-driving car will fail to work since rain streaks can severely degrade the visibility of images. How to remove rain streaks from rainy images becomes meaningful. In this paper, we propose a novel method to remove rain streaks from a single image.

*This work was supported in part by National Natural Science Foun-dation of China (No. U1509206).

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 '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351149

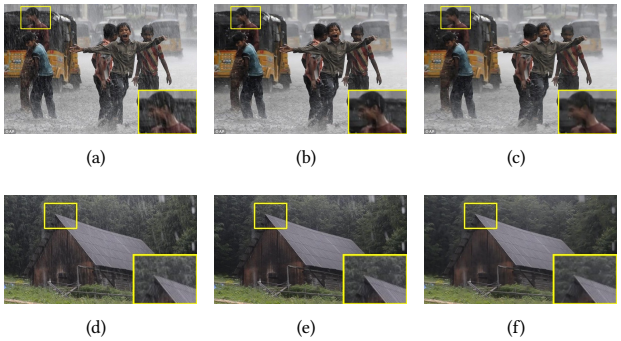


Figure 1: Derained results of real-world rainy images. (a) & (d) are rainy images, (b) & (e) result from Li et al. [21], and (c) & (f) result from the proposed method. Yellow rectangle box in each image shows the texture details. The proposed method achieves better visual quality.

Actually, deraining methods can be categorized into two parts: single image deraining methods and video-based deraining methods. Video-based deraining methods focus on fully utilizing temporal and spatial correlations between neighboring video frames. Instead, single image deraining methods pay attention to remove rain streaks from a rainy image. By regarding the rain removal problem as a signal separation problem, a rainy image can be decomposed into rain layer and background. Recently, learning based methods have been widely employed on single image deraining [7–9, 20, 21, 26, 33, 35, 36]. However, there still remains several limitations:

- Rainy image decomposition problem is ill-posed. However, most learning based deraining methods [7–9, 26, 35, 36] remove rain streaks in one stage directly. Thus relatively low performances are obtained.
- Existing stage-by-stage methods [20, 33] neglect the correlations between neighboring stages. Though [21] applies recurrent unit to preserve useful information in current stage and benefit the later stages, the direct flow of rain details from previous recurrent block to current recurrent block leads to the redundant background details.

In view of foregoing, we develop a deraining model that is capable to resolve these problems and remove rain steaks effectively. Differing from [20, 33], we consider the correlations between neighboring stages instead of removing rain streaks independently in

each stage. Though [21] treats rain removal stages correlatively, simply stacking recurrent units to remove rain streaks will cause redundant background details and achieve relatively low performances. So we adopt a hierarchy enhancement unit (HEU) to fully extract local hierarchical features and generate effective features first. Subsequently, recurrent enhancement unit (REU) is applied to memorize useful information in current stage and benefit the later stages. As shown in Fig.1, the derained results of the proposed method **(Fig. 1 (c) & (f))** obtains better visual quality compared with the derained results of [21] **(Fig. 1 (b) & (e))**.

Overall, this paper makes the following contributions:

- We propose a novel method named recurrent hierarchy enhancement network (ReHEN) for single image deraining which can remove rain streaks effectively.
- In ReHEN, we construct a recurrent hierarchy enhancement block (ReHEB). In ReHEB, hierarchy enhancement unit (HEU) is applied for effective feature extraction. And by incorporating recurrent enhancement unit (REU), information from previous stage has access to the later stages and guide them to remove rain streaks.
- The proposed method outperforms the state-of-the-art methods on five synthetic datasets and a real-world rainy image set.

2 RELATED WORK

To remove rain streaks from rainy images, many methods have been proposed to separate the rain layers from rainy images. All these methods can be categorized into two parts: video-based deraining methods as well as single image deraining methods. In the following parts, we will review these methods briefly.

Video-based deraining methods. Neighboring video frames are highly correlated. [2, 29] fully leverage temporal and spatial information of neighboring video frames to capture rain streaks effectively. Besides, [10] analyzes rain streaks by using average densities of rain between neighboring frames which is effective to static scene while method in [19] is robust to dynamic scene. Meanwhile, [4] proposes a method to remove rain streaks by describing a rain model of low-rank appearance representation. Furthermore, [17, 28] apply matrix decomposition to analyze rain streaks in videos. In [31], Wei et al. propose a patch-based mixture of Gaussian for rain removal. [15] utilizes directional gradient priors to remove rain steaks. Recently, [23] first visits the rain removal case including rain occlusions and proposes a learning based hybrid video rain model for rain removal. And [3] proposes a super-pixel segmentation based method to decompose the scene into rain layer and background efficiently.

Single image deraining methods. Differing from video-based deraining methods, rainy images are mutually independent. Thus it is more difficult to separate rain layers from rainy images. Traditionally, [16, 24, 37] decompose a rainy image into rain layer and background to remove rain streaks efficiently. In [22], a Gaussian mixture model is proposed to provide priors for both background and rain layer. Besides, [11] integrates analysis sparse representation and synthesis sparse representation and describes a joint convolutional analysis and synthesis sparse representation model to clean rain streaks. Moreover, Deng et al. employ a global sparse model to tackle deraining problem [6]. Recently, many learning based deraining methods are proposed to resolve the ill-posed inverse problem. Eigen et al. first employ a convolutional network to remove rain streaks from rainy images directly [7]. Differently, Fu et al. decompose a rainy image into low-frequency layer and high-frequency layer and map high-frequency layer to rain layer using a deep detail network [9]. In [32, 33], Yang et al. design a deep recurrent dilated network to jointly detect and remove rain streaks. And Zhang et al. construct a multi-stream dense network [35] to extract multi-scale features in low level and apply a classification network to enhance the perception in high level. Besides, Zhang et al. incorporate adversarial learning to prevent the degradation of background [36]. To preserve details of derained image, Pan et al. apply a dual convolutional neural network [26] to jointly estimate the structures and details. Methods mentioned above mostly remove rain streaks in one stage. [20] constructs a novel multi-stage convolutional network to remove rain streaks stage by stage, while [21] considers the correlations between neighboring stages and employs recurrent units to construct a recurrent squeeze-and-excitation context aggregation net for single image deraining.

3 PROPOSED METHOD

Since deraining is an ill-posed inverse problem, to tackle this problem, [33] proposes a model as follows:

$$I = \alpha(B + \sum_{t=1}^{K} R^t M) + (1 - \alpha)A \tag{1}$$

where I is the rainy image, B is the background, R^t is the rain layer, t is the index of the rain layer, K is the maximum number of rain layers, M is the rain streak mask, A is the global atmospheric light, and α is the atmospheric transmission.

Instead of using decomposition method with artifact priors (e.g. rain streaks mask) to remove rain streaks from rainy images, we simplify model in Eq. (1) and use a linear model as follows:

$$I = B + R \tag{2}$$

Our goal is to model the rain distribution and estimate rain layer R. By substracting R from rainy image I, we can obtain the background B. Owing to the ill-posed inverse problem of deraining, it is impractical to remove rain streaks in one stage. Thus we propose a novel recurrent hierarchy enhancement network (ReHEN) to model the rain distribution and remove rain streaks stage by stage (as shown in Fig. 2).

3.1 Network Structure

By decomposing rain removal procedure into multiple stages, we can formulate the procedure as follows:

$$\begin{aligned} R^t &= f_{ReHEN}(I^{t-1}) \\ I^t &= I^0 - R^t \\ 1 &\leq t \leq K \end{aligned} \tag{3}$$

where I^t indicates rainy residual image in t-th stage, R^t indicates rain layer estimated in t-th stage, K indicates the maximum number of the recurrent rain removal stages, and I^0 is the input rainy image.

As shown in Eq. (3), ReHEN maps rainy residual image I^t to obtain rain layer R^t in each stage by incorporating information

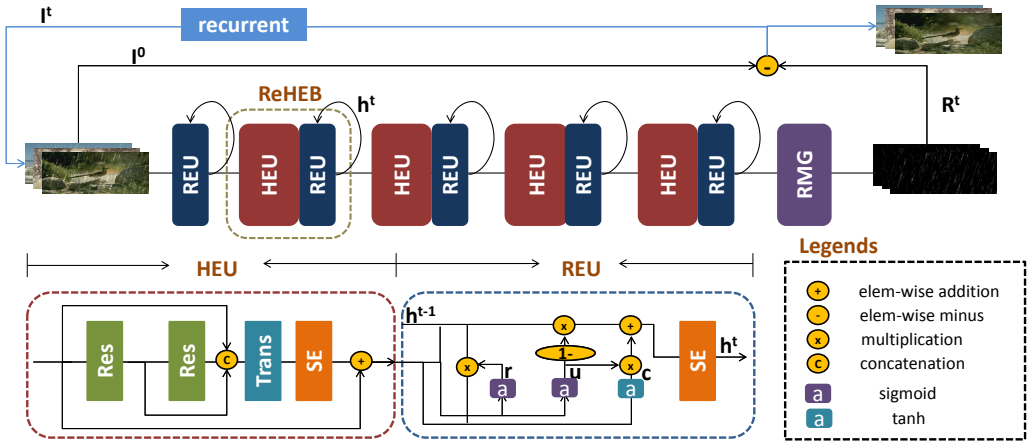


Figure 2: The whole structure of recurrent hierarchy enhancement network (ReHEN).

from previous stage. Notably, we predict the whole rain layer in each stage. By substracting the whole rain layer R^t predicted in t-th stage from input rainy image I^0, rainy residual image I^t is obtained as the input of the later stage.

Specifically, in ReHEN, we stack ReHEBs to extract effective features and incorporate useful information from neighboring stages. In the beginning of each stage, we use a REU to extract shallow features:

$$F_{SF}^t = f_{REU}(I^t) \tag{4}$$

where F_{SF}^t indicates shallow features extracted by REU in t-th stage.

Then, ReHEBs are stacked to extract further features. Each ReHEB contains a HEU and a REU. HEU is used to fully extract hierarchical features and generate effective high-level features while REU is applied to fully select and exploit useful information from HEU and neighboring stages.

$$F_{ReHEB,i}^t = f_{ReHEB,i}(F_{ReHEB,i-1}^t, h_{ReHEB,i}^{t-1}) \quad 1 \le i \le P \tag{5}$$

where P indicates the number of ReHEBs stacked in our network, $F_{ReHEB,i-1}^t$ indicates features extract by (i-1)-th ReHEB in t-th stage, $h_{ReHEB,i}^{t-1}$ indicates hidden status in i-th ReHEB deriving from (t-1)-th stage. Notably, $F_{ReHEB,0}^t = F_{SF}^t$ and $h_{ReHEB,i}^0 = None$. Furthermore, by unfolding Eq. (5), we can obtain the following formulation:

$$\begin{aligned} F_{HEU,i}^t &= f_{HEU,i}(F_{ReHEB,i-1}^t) \\ F_{REU,i}^t &= f_{REU,i}(F_{HEU,i}^t, h_{ReHEB,i}^{t-1}) \\ F_{ReHEB,i}^t &= h_{ReHEB,i}^t = F_{REU,i}^t \end{aligned} \tag{6}$$

Finally, rainy map generator (RMG) is applied to obtain rain layer efficiently. RMG maps the features from the last ReHEB into rain layer. A squeeze-and-excitation (SE) [13] block is applied in RMG to distribute different scale factors for each feature to focus on different rain details adaptively.

$$R^t = f_{RMG}(F_{ReHEB,P}^t) \tag{7}$$

where f_{RMG} is the combination operation of **conv-SE-conv**.

By substracting R^t , the derained result can be obtained.

$$I^t = I^0 - R^t \tag{8}$$

3.2 Hierarchy Enhancement Unit

Even though some methods [7–9, 20, 21, 26, 33, 35, 36] have achieved performance improvements on single image deraining, most of them tend to predict rain layer directly while neglecting the local hierarchical features in each convolutional layer. Actually, local hierarchical features contain different rain details. They can boost the deraining performances considerably. Based on the motivation, we construct a hierarchy enhancement unit (HEU) to fully extract local hierarchical features and generate more efficient features adaptively.

In HEU, residual blocks [12] are stacked to extract local hierarchical features.

$$F_{Res_j,i}^t = f_{Res_j,i}(F_{Res_{j-1},i}^t), 1 \le j \le Q \tag{9}$$

where $F_{Res_j,i}^t$ indicates features from j-th residual block of i-th ReHEB in t-th stage and $F_{Res_0,i}^t = F_{ReHEB,i-1}^t$.

To make full use of local hierarchical features, all the features extracted by each residual block are concatenated together and fed into a transition layer to learn more effective features.

$$F_{Trans,i}^t = f_{Trans,i}([F_{Res_0,i}^t, ..., F_{Res_j,i}^t, ..., F_{Res_Q,i}^t]) \tag{10}$$

where $[F_{Res_0,i}^t, ..., F_{Res_j,i}^t, ..., F_{Res_Q,i}^t]$ refers to the concatenation of the whole local hierarchical features extracted by residual blocks and $f_{Trans,i}(.)$ indicates **conv-bn** operation.

Though features obtained by transition layer contain different rain details, they are spatially correlated. Thus we apply a squeeze-and-excitation (SE) block to explicitly model inter-dependencies of features before they are fed into next transformation. In SE block, a squeeze operation is applied to exploit contextual information in the whole feature maps. Then an excitation operation is used to fully capture dependencies of features. As a result, scale factors for each feature are obtained as the weights of each feature map.

Moreover, we allow the flow of rain details from low level to high level to preserve rain details from low level.

$$\begin{aligned} s^t_{HEU,i} &= f_{SE,i}(F^t_{Trans,i}) \\ F^t_{HEU,i} &= g_{scale,i}(s^t_{HEU,i}, F^t_{Trans,i}) + F^t_{ReHEB,i-1} \end{aligned} \tag{11}$$

where $s^t_{HEU,i}$ indicates the scale factors and $g_{scale,i}(.)$ indicates the operation of multiplication between $F^t_{Trans,i}$ and $s^t_{HEU,i}$.

3.3 Recurrent Enhancement Unit

Recurrent structure used in [20, 33] simply considers the output image of current stage as input of next stage and neglects the correlations between the neighboring stages. In fact, all these input images $\{I^0, ..., I^t, ..., I^P\}$ in different stages are temporally correlated. Features extracted in each stage contain useful information which can guide rain removal efficiently in the later stages. Thus it is meaningful to investigate the correlations between features in different stages. We incorporate recurrent neural network [25] with memory unit to better make use of information in previous stages and benefit feature learning in the later stages.

Specifically, gated recurrent unit (GRU) [5] is used in this paper. Differing from long short-term memory (LSTM) [34], GRU lacks an output gate leading to fewer parameters than LSTM while maintaining similar performances. Moreover, convolutional GRU is applied in the proposed method. The convolutional GRU can be formulated as follows:

$$\begin{aligned} u^t_i &= \sigma(W_{u,i} \otimes F^t_{HEU,i} + U_{u,i} \otimes h^{t-1}_{ReHEB,i} + b_{u,i}) \\ r^t_i &= \sigma(W_{r,i} \otimes F^t_{HEU,i} + U_{r,i} \otimes h^{t-1}_{ReHEB,i} + b_{r,i}) \\ c^t_i &= \sigma_h(W_{c,i} \otimes F^t_{HEU,i} + U_{c,i} \otimes (r^t_i \odot h^{t-1}_{ReHEB,i}) + b_{c,i}) \\ h^t_{ReHEB,i} &= (1 - u^t_i) \odot h^{t-1}_{ReHEB,i} + u^t_i \odot c^t_i \end{aligned} \tag{12}$$

where u indicates update gate, r indicates reset gate, W and U indicate convolutional kernels with a size of 3×3, b indicates bias, σ indicates *sigmoid* function while σ_h indicates *tanh* function, and $\otimes$ and $\odot$ indicate convolutional operation and element-wise multiplication respectively.

Similarly, a squeeze-and-excitation (SE) block is applied to explicitly model inter-dependencies of features.

$$\begin{aligned} s^t_{REU,i} &= f_{SE,i}(h^t_{ReHEB,i}) \\ F^t_{REU,i} &= g_{scale,i}(s^t_{REU,i}, h^t_{ReHEB,i}) \end{aligned} \tag{13}$$

3.4 Loss Function

Intuitively, the derained images should be equal to corresponding rain-free images in pixel level. Thus we promote the constraint of derained images and rain-free images using mean square error (MSE) loss:

$$L_{MSE} = \frac{1}{K} \sum_{t=1}^{K} ||I^t - I_{gt}||_2 \tag{14}$$

where K indicates the maximum number of stages, and I_{gt} indicates the rain-free image.

However, a model trained with MSE loss tends to obtain a blurred reconstruction [18, 27]. Motivated by the observation that structural similarity (SSIM) [8] loss can preserve structure better, we also adopt SSIM loss.

$$L_{SSIM} = log(\frac{1.0}{f_{ssim}(I^K, I_{gt}) + \epsilon_1}) \tag{15}$$

where I^K indicates the derained image of the last stage, f_{ssim} indicates the function that calculates the SSIM value between I^K and I_{gt}, and ϵ_1 indicates a tiny constant which is applied to avoid zero in denominator.

Furthermore, we also leverage the peak signal-to-noise ratio (PSNR) [14] loss to keep the global similarity:

$$L_{PSNR} = \frac{1.0}{f_{psnr}(I^K, I_{gt}) + \epsilon_2} \tag{16}$$

where I^K indicates the derained image of the last stage and f_{psnr} indicates the function that calculates the PSNR value between I^K and I_{gt}. Similarly, ϵ_2 is added to avoid zero in denominator.

Combining all the three loss functions together, our final loss function is formulated as follows:

$$L = L_{MSE} + \lambda_1 L_{SSIM} + \lambda_2 L_{PSNR} \tag{17}$$

where λ_1 and λ_2 represent the weights of L_{SSIM} and L_{PSNR} respectively.

4 EXPERIMENTS

We conduct experiments on five synthetic datasets and a real-world rainy image set and compare the proposed method with the state-of-the-art methods: DSC [24], LP [22], JCAS [11], DDN [9], ID_CGAN [36], JORDER [33], DID-MDN [35], DualCNN [26], and RESCAN [21]. Deraining performances on the synthetic datasets are evaluated in terms of peak signal-to-noise (PSNR) [14] and structural similarity (SSIM) [30]. Since the ground truth of real-world rainy image is not available, performances of different methods on real-world rainy images are evaluated visually.

4.1 Datasets

As tabulated in Table 1, five synthetic datasets and a real-world rainy image set are chosen for comparison. Rain100L and Rain100H are derived from [33]. Rain800 is obtained from [36]. Rain1200 comes from [35] while Rain1400 is provided by [9]. Moreover, real-world rainy images are collected from [33, 36]. Since rain-free images of real-world rainy images are difficult to obtain, real-world rainy image set is used for testing only. Remarkably, we only use rain-free/rainy image pairs for training without other additional labels.

Table 1: Descriptions of synthetic and real-world datasets. Values in each column of Train-set and Test-set indicate the number of rain-free/rainy image pairs except the real-world set with rainy images only.

Datasets	Train-set	Test-set	Label
Rain100L	200	100	rain mask/rain map
Rain100H	1800	100	rain mask/rain map
Rain800	700	100	-
Rain1200	12000	1200	rain-density labels
Rain1400	12600	1400	-
Real-world set	-	67	-

4.2 Training Details

In our network, we set $P = 4$ and $Q = 2$ which indicate ReHEN contains 4 ReHEBs and ReHEB stacks 2 residual blocks respectively. Maximum number of stages is set as $K = 4$. Moreover, filters of all convolutional layers have a size of 3×3. The learning rate is initialized as 0.005 and decreased to 0.0005 after 50000 iterations. The weights of the loss function are set as $\lambda_1 = 0.001$ and $\lambda_2 = 0.1$. And constants in PSNR loss and SSIM loss are set as $\epsilon_1 = 1e-5$ and $\epsilon_2 = 1e-3$. During training, 32 rain-free/rainy patch pairs with a size of 64×64 are randomly generated from input image pairs per iteration. Adam optimizer is adopted to optimize ReHEN. β_1 and β_2 of Adam optimizer are set as 0.9 and 0.999 respectively. We implement the model with Tensorflow library [1] and train it on an NVIDIA Geforce GTX 1070 GPU for 60000 iterations.

4.3 Ablation Study

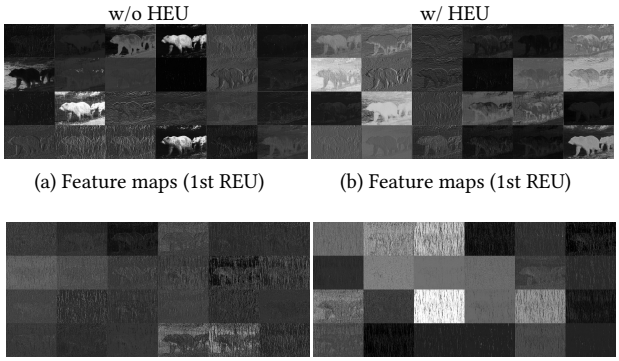
We conduct three ablation studies to demonstrate the efficiency of ReHEN. For simplification, Rain100L and Rain100H are used for evaluation. Ablation studies on module, loss, and stage are investigated as follows:

Table 2: Investigation of hierarchy enhancement unit (HEU), recurrent enhancement unit (REU), squeeze-and-excitation (SE) block, and multiple stages (MS) of rain removal. Note that MS-1 indicates one stage and MS-4 indicates four stages. Performances are evaluated on Rain100L dataset.

Modules	Different combinations of modules				
HEU (w/o SE)	√	×	√	√	√
REU (w/o SE)	×	√	√	√	√
SE	×	×	×	√	√
MS-1	√	√	√	√	×
MS-4	×	×	×	×	√
PSNR	28.82	34.36	35.31	36.49	**37.41**
SSIM	0.894	0.963	0.969	0.975	**0.980**

Ablation study on module. Table 2 shows the ablation investigation on the effects of hierarchy enhancement unit (HEU), recurrent enhancement unit (REU), squeeze-and-excitation (SE) block, and multiple stages (MS) of rain removal. For fair comparison, 4 variants of ReHEB are stacked in the five networks. We can see that simply stacking HEUs or REUs without SE block (denote as HEU (w/o SE) and REU (w/o SE)) and removing rain steaks in one stage achieves relatively low performances. Applying HEU (w/o SE) to fully extract local hierarchical features and generate high-level features and adopting REU (w/o SE) to select useful information adaptively can boost the rain removal performances considerably. Moreover, SE blocks are applied in both HEU and REU and assign different scale factors to each feature which can improve the performances tremendously. Since rain removal with one stage fails to fit the ill-posed inverse deraining problem to some extent, we tend to remove rain streaks stage by stage. By incorporating the correlations between neighboring stages, the network can remove rain streaks efficiently.

We further explore the improvements obtained by HEU visually. For simplification, we visualize the feature maps of the 1st REU and 4th REU of two different structures in Fig. 3. As can be observed, incorporating HEU and REU to construct the network can obtain richer features in the 1st REU (**see Fig. 3 (a) & (b)**). Besides, in the 4th REU, the structure with HEU can capture more rain details while discarding background (**see Fig. 3 (c) & (d)**).



(a) Feature maps (1st REU) (b) Feature maps (1st REU)

(c) Feature maps (4th REU) (d) Feature maps (4th REU)

Figure 3: Feature map visualization of the 1st REU and 4th REU between two different structures. 'w/o HEU' indicates using REU to construct ReHEN while 'w/ HEU' indicates using both HEU and REU to construct ReHEN. Performances are evaluated on Rain100L dataset.

Table 3: Investigation of loss function on Rain100L dataset. For simplification, ReHEN with one-stage rain removal structure is applied in the ablation study.

Loss	Combinations of losses			
MSE loss	√	√	√	√
SSIM loss	×	√	×	√
PSNR loss	×	×	√	√
PSNR	36.31	36.27	36.33	**36.49**
SSIM	0.972	0.973	0.974	**0.975**

Ablation study on loss. We further investigate the contribution of each loss function (MSE loss, SSIM loss, as well as PSNR loss). Notably, MSE loss is applied as the baseline of the whole experiments. Experimental results of different combinations of loss functions are tabulated in Table 3. For simplification, we conduct the experiments using ReHEN with one-stage rain removal structure. As can be observed in Table 3, SSIM loss can boost the performance of SSIM metric to some extent, since it can help to capture visual perception as human and keep the structural similarity. Besides, PSNR loss can enhance both PSNR metric and SSIM metric which indicates that PSNR loss can help to keep the global similarity between derained results and the corresponding rain-free images. As shown in Fig. 4, by combining MSE loss, SSIM loss, and PSNR loss, the proposed method can efficiently remove rain streaks and keep the texture details.

Ablation study on stage. Furthermore, we conduct an ablation study on the number of recurrent stages. As shown in Table 4 and Fig. 5, a recurrent rain removal scheme can enhance the deraining

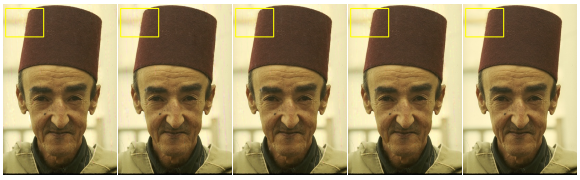


Figure 4: Derained results of each combination of loss functions. Results are evaluated on Rain100L dataset. As can be seen, loss function with MSE loss, SSIM loss, and PSNR loss obtains the best visual quality.

Table 4: Investigation of recurrent rain removal stages. Note that MS-N indicates recurrent rain removal with N stages. Performances are evaluated on Rain100L dataset.

Metrics	MS-1	MS-2	MS-3	MS-4
PSNR	36.49	37.08	37.31	**37.41**
SSIM	0.975	0.978	0.979	**0.980**

performances considerably. The gap between one stage and two stages is large which demonstrates that recurrent enhancement unit (REU) can preserve rain details from previous stage and guide the later stages to remove rain streaks efficiently. Moreover, the improvements between two stages and three stages or three stages and four stages are marginal which shows that rich rain details have been kept in each hidden status in the second stage (as discussed in Section 4.6). Furthermore, in Fig. 6, we can observe that derained images obtain better visual quality with the increased rain removal stages. To balance the performances and the efficiency, we remove rain streaks from rainy images in four stages.

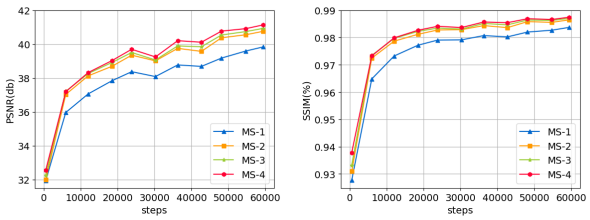


Figure 5: Training convergence analysis on PSNR/SSIM of ReHEN with different recurrent rain removal stages. Performances are evaluated on Rain100L dataset.

4.4 Results on Synthetic Datasets

Quantitative evaluation. We compare quantitative performances of the proposed method with the state-of-the-art methods on five synthetic datasets. Table 5 tabulates the quantitative results of PSNR and SSIM metrics evaluated on different methods. As can be observed, the proposed method achieves tremendous improvements over the state-of-the-art methods on PSNR and SSIM metrics

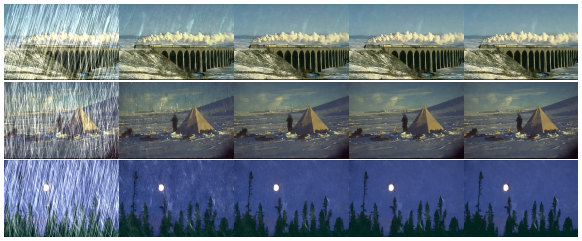


Figure 6: Derained results of three rainy images in each stage. Performances are evaluated on Rain100H dataset.

while requiring relatively few parameters. Specifically, the proposed method obtains significant improvements by PSNR metric of 0.77db, 1.52db, 2.85db, 2.69db, 2.56db and SSIM metric of 0.5%, 1.8%, 1.3%, 0.6%, 1.7% on Rain100L, Rain100H, Rain800, Rain1200, as well as Rain1400 respectively. On complex synthetic datasets like Rain100H, Rain800, and Rain1400, the proposed method outperforms existing state-of-the-art methods more than 1.5db on PSNR metric and 1.3% on SSIM metric. The quantitative results show that the proposed method can tackle the complex rainy conditions efficiently.

Running time comparison is also shown in Table 5. We choose learning based methods for comparison and test them on an NVIDIA Geforce GTX 1070 GPU. Since images in Rain1200 dataset have an equal size of 512×512, we choose Rain1200 dataset for running time comparison. As can be observed, the running time of the proposed method is competitive to ID_CGAN [36] as well as JORDER [33] while exceeding CNN [7], DDN [9], RESCAN [21], DID-MDN [35], as well as DualCNN [26]. Though the running time of DID-MDN [35] is similar to the proposed method, the performances of the proposed method surpass DID-MDN [35] considerably with the improvements of 4.69db on PSNR metric and 0.6% on SSIM metric on Rain1200 dataset.

Qualitative evaluation. To visually demonstrate the performances of the proposed method, Fig. 7 shows the derained results of synthetic rainy images. Since the limitation of space, we only illustrate derained results of three recent state-of-the-art methods. As shown in Fig. 7, DualCNN [26] fails to remove rain streaks from images with heavy rain while JORDER [33] tends to dim and blur the derained results. Comparing with RESCAN [21], the proposed method can remove rain streaks and keep the texture and structure details efficiently.

4.5 Results on Real-World Images

To assess the generalization of the proposed method, we also evaluate the proposed ReHEN on real-world rainy images. For fair comparison, all the methods evaluated on real-world rainy images use the pre-trained model trained on Rain800. We select three real-world rainy images containing rain streaks with different scales, shapes, and densities for comparison. As shown in Fig. 8, the proposed method achieves the best visual quality compared with the state-of-the-art methods. JORDER [33] and DID-MDN [35] tend to dim and blur the details of the derained results and RESCAN [21] fails to remove rain streaks from images in most cases. Though

Figure 7: Derained results of JORDER [33], DualCNN [26], RESCAN [21], and the proposed method on the synthetic datasets. Values below every image indicate SSIM value and PSNR value respectively. Metrics with bold font are the highest ones in this row. More derained results on synthetic rainy images are shown in the supplementary material.

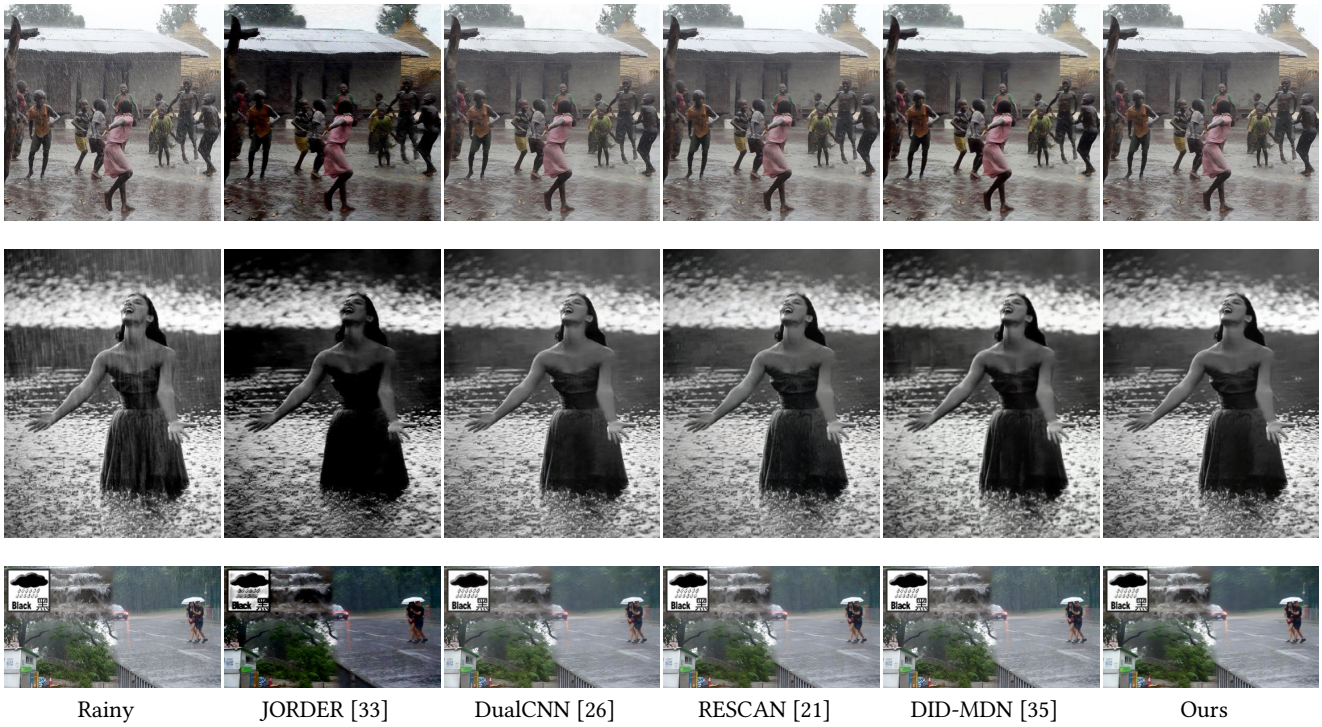


Figure 8: Derained results of JORDER [33], DualCNN [26], RESCAN [21], DID-MDN [35], and the proposed method on real-world rainy images. More derained results on real-world rainy images are shown in the supplementary material.

Table 5: Parameter complexity, running time, and average PSNR and SSIM values on five synthetic datasets. The value with red bold font indicates ranking the first place in this column while value with blue font is the second place. For fair comparison, we re-train the learning based methods on the synthetic datasets whose metrics are not reported by the authors. Since density labels in Rain100L, Rain800, and Rain1400 are not available, we fine-tune DID-MDN [35] without the classification network.

		Time (s)	Rain100L		Rain100H		Rain800		Rain1200		Rain1400	
Methods	Params	512×512	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rainy	-	-	25.52	0.825	12.13	0.349	21.16	0.652	21.15	0.778	23.69	0.757
DSC (ICCV'15) [24]	-	-	24.16	0.870	15.66	0.544	18.56	0.599	21.44	0.789	22.03	0.799
LP (CVPR'16) [22]	-	-	29.11	0.880	14.26	0.423	22.27	0.741	22.75	0.835	25.64	0.836
JCAS (ICCV'17) [11]	-	-	28.40	0.881	13.65	0.459	22.19	0.766	27.91	0.778	28.77	0.819
DDN (CVPR'17) [9]	57,369	0.547	32.04	0.938	24.95	0.781	21.16	0.732	27.33	0.898	27.61	0.901
ID_CGAN ('17) [36]	817,824	0.286	25.88	0.891	14.16	0.607	22.73	0.817	23.32	0.803	21.93	0.784
JORDER (CVPR'17) [33]	369,792	0.268	36.11	0.970	22.15	0.674	22.24	0.776	24.32	0.862	27.55	0.853
DID-MDN (CVPR'18) [35]	372,839	0.532	25.70	0.858	17.39	0.612	21.89	0.795	27.95	0.908	27.99	0.869
DualCNN (CVPR'18) [26]	687,008	20.19	26.87	0.860	14.23	0.468	24.11	0.821	23.38	0.787	24.98	0.838
RESCAN (ECCV'18) [21]	134,424	0.750	36.64	0.975	26.45	0.846	24.09	0.841	29.95	0.884	28.57	0.891
Ours	298,263	0.531	**37.41**	**0.980**	**27.97**	**0.864**	**26.96**	**0.854**	**32.64**	**0.914**	**31.33**	**0.918**

DualCNN [26] can remove rain streaks to some extent, there still remains some rain streaks in derained images. The proposed method can remove most rain streaks of real-world rainy images while maintaining texture and structure details.

4.6 Visualization Results of Hidden Statuses

To understand what information recurrent enhancement unit (REU) keeps in each stage, we inspect the hidden statuses of each REU in all rain removal stages. To better visualize the hidden statuses, we consider an operation T that transforms a 3D tensor to a 2D tensor defined over the channel dimension and visualize them with heatmap. The operation T can be formulated as follows:

$$T : F^{h\times w\times c} \rightarrow F^{h\times w} \tag{18}$$

Specifically, we consider the mean of the feature maps over the channel dimension which can be formulated as follows:

$$T(F) = \frac{1}{c}\sum_{i=1}^{c} F_i \tag{19}$$

where F_i indicates the i-th feature map and $\sum$ indicates the element-wise addition operation. The average feature map roughly represents the whole feature maps.

As illustrated in Fig. 9, in hidden status of the first REU, backgrounds of rainy images are obtained. Then in the following two hidden statuses, REU tends to extract smooth rain streak areas. Finally, in the last two hidden statuses, REU gradually reduces the smooth areas and obtains the clear rain details. With the increased rain removal stage, hidden statuses of each REU are more accurate in rain streaks and background extraction. Specifically, in the first REU, rain streaks are less and less while background becomes more clear. And in the fourth REU, background is discarded in the last stage while rain streaks become richer. Therefore, in stage t, taking rainy residual image I^{t-1} as input and incorporating rain details from I^{t-1} with the hidden statuses h^{t-1} of t-th stage, the proposed network can estimate R^t more accurately.

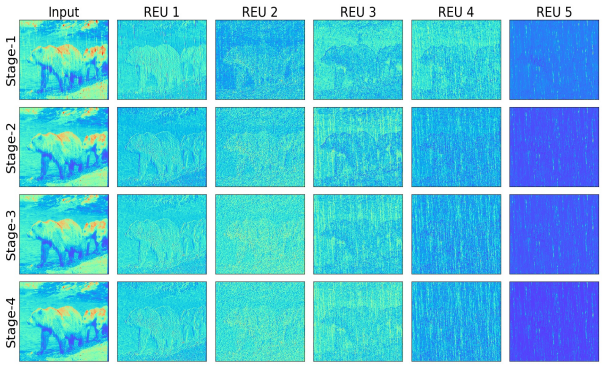


Figure 9: Visualization of the hidden statuses of all REUs in four stages. Images of the first column are input rainy residual images I^t of each stage while images of the other five columns are hidden statuses of each REU.

5 CONCLUSIONS

In this paper, we introduce a recurrent hierarchy enhancement network (ReHEN) for single image deraining. In ReHEN, a recurrent hierarchy enhancement block (ReHEB) is proposed as the sub-module of ReHEN. Specifically, we adopt hierarchy enhancement unit (HEU) to fully extract local hierarchical features and generate high-level features effectively. And a recurrent enhancement unit (REU) is applied to preserve the useful information from HEU and benefit the rain removal in the later stages. In HEU and REU, a squeeze-and-excitation (SE) block is added to explicitly model inter-dependencies of features. Experiments on synthetic datasets and real-world rainy images demonstrate that the proposed method outperforms the state-of-the-art methods on quantity and visual quality.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.

[2] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. 2010. Analysis of rain and snow in frequency space. *International Journal of Computer Vision* 86, 2-3 (2010), 256.

[3] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6286–6295.

[4] Yi-Lei Chen and Chiou-Ting Hsu. 2013. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1968–1975.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] Liang-Jian Deng, Ting-Zhu Huang, Xi-Le Zhao, and Tai-Xiang Jiang. 2018. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling* 59 (2018), 662–679.

[7] David Eigen, Dilip Krishnan, and Rob Fergus. 2013. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE International Conference on Computer Vision*. 633–640.

[8] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. 2018. Residual-Guide Network for Single Image Deraining. In *Proceedings of the 26th ACM International Conference on Multimedia*. 1751–1759.

[9] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. 2017. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1715–1723.

[10] Kshitiz Garg and Shree K Nayar. 2004. Detection and removal of rain from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 528–535.

[11] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. 2017. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1717–1725.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[13] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7132–7141.

[14] Quan Huynh-Thu and Mohammed Ghanbari. 2008. Scope of validity of PSNR in image/video quality assessment. *Electronics letters* 44, 13 (2008), 800–801.

[15] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. 2018. FastDeRain: A Novel Video Rain Streak Removal Method Using Directional Gradient Priors. *arXiv preprint arXiv:1803.07487* (2018).

[16] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. 2012. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing* 21, 4 (2012), 1742–1755.

[17] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. 2015. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing* 24, 9 (2015), 2658–2670.

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 105–114.

[19] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. 2018. Video Rain Streak Removal By Multiscale Convolutional Sparse Coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6644–6653.

[20] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. 2017. Single Image Deraining using Scale-Aware Multi-Stage Recurrent Network. *arXiv preprint arXiv:1712.06830* (2017).

[21] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. 2018. Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining. In *The European Conference on Computer Vision (ECCV)*. 262–277.

[22] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. 2016. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2736–2744.

[23] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. 2018. Erase or Fill? Deep Joint Recurrent Rain Removal and Reconstruction in Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3233–3242.

[24] Yu Luo, Yong Xu, and Hui Ji. 2015. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*. 3397–3405.

[25] Danilo P Mandic and Jonathon Chambers. 2001. *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc.

[26] Jinshan Pan, Sifei Liu, Deqing Sun, Jiawei Zhang, Yang Liu, Jimmy Ren, Zechao Li, Jinhui Tang, Huchuan Lu, Yu-Wing Tai, et al. 2018. Learning Dual Convolutional Neural Networks for Low-Level Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3070–3079.

[27] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2536–2544.

[28] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. 2017. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4210–4219.

[29] Varun Santhaseelan and Vijayan K Asari. 2015. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision* 112, 1 (2015), 71–89.

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

[31] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. 2017. Should we encode rain streaks in video as deterministic or stochastic. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2516–2525.

[32] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2016. Joint rain detection and removal via iterative region dependent multi-task learning. *CoRR, abs/1609.07769* 2 (2016), 3.

[33] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1357–1366.

[34] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014).

[35] He Zhang and Vishal M. Patel. 2018. Density-Aware Single Image De-Raining Using a Multi-Stream Dense Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 695–704.

[36] He Zhang, Vishwanath Sindagi, and Vishal M Patel. 2017. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957* (2017).

[37] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. 2017. Joint bilayer optimization for single-image rain streak removal. In *Proceedings of the IEEE International Conference on Computer Vision*. 2526–2534.