Non-locally Enhanced Encoder-Decoder Network for Single Image De-raining

Guanbin Li
Sun Yat-sen University
liguanbin@mail.sysu.edu.cn

Xiang He
Sun Yat-sen University
hexiang7@mail2.sysu.edu.cn

Wei Zhang
Sun Yat-sen University
zhangwei.hi@gmail.com

Huiyou Chang
Sun Yat-sen University
isschy@mail.sysu.edu.cn

Le Dong*
University of Electronic Science and Technology of China
ledong@uestc.edu.cn

Liang Lin
Sun Yat-sen University
linliang@ieee.org

ABSTRACT

Single image rain streaks removal has recently witnessed substantial progress due to the development of deep convolutional neural networks. However, existing deep learning based methods either focus on the entrance and exit of the network by decomposing the input image into high and low frequency information and employing residual learning to reduce the mapping range, or focus on the introduction of cascaded learning scheme to decompose the task of rain streaks removal into multi-stages. These methods treat the convolutional neural network as an encapsulated end-to-end mapping module without deepening into the rationality and superiority of neural network design. In this paper, we delve into an effective end-to-end neural network structure for stronger feature expression and spatial correlation learning. Specifically, we propose a non-locally enhanced encoder-decoder network framework, which consists of a pooling indices embedded encoder-decoder network to efficiently learn increasingly abstract feature representation for more accurate rain streaks modeling while perfectly preserving the image detail. The proposed encoder-decoder framework is composed of a series of non-locally enhanced dense blocks that are designed to not only fully exploit hierarchical features from all the convolutional layers but also well capture the long-distance dependencies and structural information. Extensive experiments on synthetic and real datasets demonstrate that the proposed method can effectively remove rain-streaks on rainy image of various densities while well preserving the image details, which achieves significant improvements over the recent state-of-the-art methods.

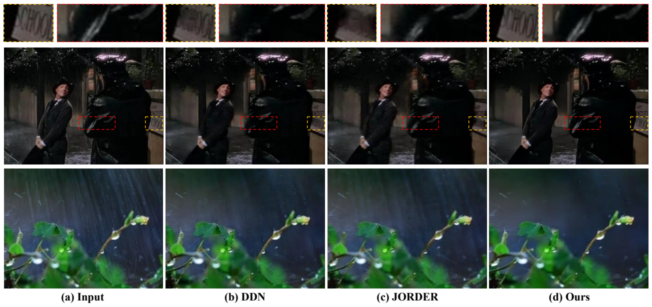


Figure 1: Sample examples of single image de-raining results. (a) Input images with rain-streaks. (b) Results of DDN [6] (c) Results of JORDER [35] (d) Our results. The first row is the enlargement of the selected regions of the second row, which shows the advantage of our proposed NLEDN in detail preserving. The third row demonstrates the promising result of our NLEDN in removing long rain streaks.

KEYWORDS

image de-raining; non-local mean calculation; dense network

ACM Reference Format:
Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. 2018. Non-locally Enhanced Encoder-Decoder Network for Single Image De-raining . In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240636

1 INTRODUCTION

Images with rain streaks are often captured by outdoor surveillance equipments, which may significantly degrade the performance of some existing computer vision systems and may also result in a pool visual experience for some multimedia applications. Automatic rain streaks removal has thus become a crucial research task in the field of computer vision and multimedia, and has been successfully applied in the fields of driverless technology [16, 31] and content based image editing [10, 28, 34, 39].

The research on visual de-raining can be traced back to the last decade. Most of the early research focused on the removal of rain streaks in video sequences captured with static cameras [2, 7, 8, 15, 27, 39]. They mostly attempted to solve the problem by exploiting

*Corresponding author is Le Dong. This work was supported by the National Natural Science Foundation of China under Grant 61702565 and Grant 61772114, the Science and Technology Planning Project of Guangdong Province under Grant 2017B010116001, Guangdong Natural Science Foundation Project for Research Teams under Grant 2017A030312006, and was also sponsored by CCF-Tencent Open Research Fund.

MM '18, October 22–26, 2018, Seoul, Republic of Korea
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240636

the temporal correlation in the luminance domain between successive frames [2, 7, 39]. Due to the lack of temporal information, de-raining on single image is more ill-posed, however, it has received widespread research attention due to its greater practicality and challenge [10, 13, 14, 35, 37]. Traditional methods on single image de-raining explore certain prior information on physical characteristics of rain streaks and model it as a signal separation problem [4, 13, 25, 30], or directly regard it as an image filtering problem and solve by resorting to nonlocal mean smoothing [14]. However, since these models are based on handcrafted low-level feature and fixed a priori rain streaks assumptions, they can only cope with rain drops of specific shapes, scales and density, and can easily lead to the destruction of image details which are similar to rain streaks.

In recent years, due to the powerful feature representation and end-to-end data inference capabilities, deep convolutional neural networks have been widely applied to single image de-raining and have achieved significant performance improvement. These methods generally model the problem as a pixel-wise image regression process which directly learns to map an input rainy image to its clean version or a negative residual map in an end-to-end mode through a series of convolution, pooling, and non-linear operations, etc. Although considerable progress has been made in comparison with traditional methods, existing deep models still suffer from several limitations. Firstly, most of the deep CNN based models emulate the experience of low-level image processing such as image denoising, super-resolution and filtering, design shallow neural network structure, and maintain a constant feature map resolution during network propagation. As the size of the network receptive field is limited, the pixel value inference of each spatial location only relies on small local surrounding regions, it is usually arduous to remove longer rain streaks (e.g. third row of Fig. 1). Moreover, due to the ignorance of long-distance spatial context modeling, these models often have difficulty in accurately filling raindrop-removed image content while detecting heavy rain streaks, resulting in an often overly blurred result, especially on texture-rich edges (e.g. first row in Fig. 1). Although various deep CNN based solutions have been proposed, existing efforts either focus on the entrance of the networks by decomposing the input image into high and low frequency information [6] or design cascaded learning schemes to decompose the task of rain removal into multi-stages [35, 37]. A contextualized dilated network is proposed in [35] to aggregate context information from three scales of receptive filed for more effective rain streak feature learning. All of these methods use convolutional neural network as an encapsulated end-to-end mapping module without deepening into the rationality and superiority of neural network design towards more effective rain streaks removal.

Inspired by the adaptive nonlocal means filter [14] for efficient single-image rain streaks removal, we proposed to incorporate nonlocal operation [32] to the design of our end-to-end de-raining network framework. The non-local operation computes the feature response at a spatial position as a weighted sum of the features at a specific range of positions in the considered feature maps [32]. Specifically, we propose a non-locally enhanced encoder-decoder network framework for single-image de-raining. The core architecture of our trainable de-raining engine is a concatenation of an encoder network and a corresponding decoder network. It is designed to be a symmetrical structure and both the encoder and the decoder network are composed of three cascaded non-locally enhanced dense blocks (abbr. NEDB). Each NEDB is designed as a residual learning module which contains a non-local feature map weighting followed by four densely connected convolution layers for hierarchical feature encoding and another convolution layer for residual inference. Moreover, we introduce the pooling striding mechanism in our encoder network to learn increasingly abstract feature representation, which results in a decrease in resolution with the enlargement of receptive filed. We further incorporate pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling in our decoder, which helps to preserve the structure and details in the resulted image.

In summary, this paper has the following contributions:

- We propose a non-locally enhanced encoder-decoder network framework for single-image rain streaks removal. It is able to learn increasingly abstract feature representation for more accurate rain streaks modeling in the encoding stage, and can remove all levels of rain streaks while promisingly preserving the texture details during decoding.
- We propose to incorporate a non-locally enhanced dense block (NEDB) in our encoder-decoder framework, which can not only fully exploit hierarchical features from all embedded convolutional layers but also well capture the long-distance dependencies for spatial context modeling.
- Experimental results on both synthetic and real datasets have demonstrated the superiority of our proposed method, which achieves significant improvements over the state-of-the-art methods.

2 RELATED WORKS

2.1 Single Image De-raining

Single image de-raining is a challenging and highly ill-posed task. Traditional approaches treat single image de-raining as an image decomposition problem, in which they model the rain streaks and rain-free scene lie in two separate sub-spaces. For example, Kang et al. [13] and Li et al. [24] rely on morphological component analysis and Gaussian mixture models (GMMs) based dictionary learning, respectively. Yu et al. [25] distinguishes the dictionaries of rain streak and rain-free scene via discriminative sparse coding. Zhu et al. [41] further considers rain direction in a joint optimization process. In these methods, although varies prior information of both rain streaks and rain-free images have been extensively exploited, due to the handcraft low-level feature representation and strong prior assumptions, they usually tend to overly smooth the details in rain free scenes.

Recently, deep learning based approaches dominate the research of image-to-image mapping for various of computer vision tasks, such as image inpainting [23], saliency detection [19–22], automatic image colorization [38] and image super-resolution[3, 5, 17, 18, 29, 40]. With the integration of several commonly used advances in network architectures such as residual connections [9] and dilated convolutions [36], recent de-raining methods [6, 35, 37] obtain impressive results by building fully convolutional architectures that learns pixel-wise mapping from rainy image to the rain-free version. Although additional considerations have been taken, such as rain

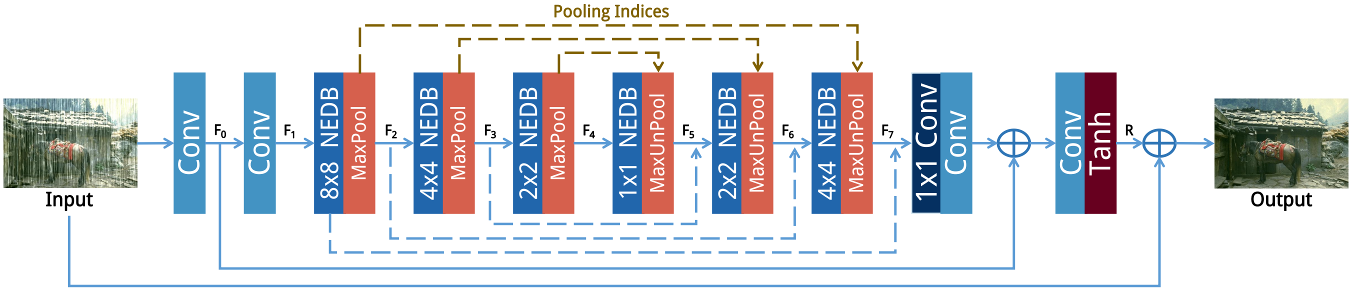


Figure 2: The overall architecture of our proposed non-locally enhanced encoder-decoder network (NLEDN). As can be observed, the input image and low-level feature activation are linked to the very end of the whole architecture via long-range skip-connections. The core of the whole architecture is a non-locally enhanced encoder-decoder, in which novel non-locally enhanced dense blocks (NEDBs) and pooling indices guided scheme are adopted.

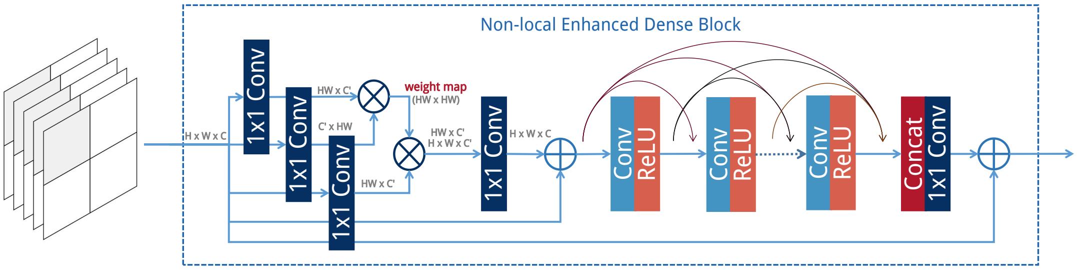


Figure 3: The architecture of our proposed non-locally enhanced dense block (NEDB). Left part shows the multi-scale input via either adopting global-level non-local enhancement which feeds the entire feature map to NEDB or dividing the feature map into a grid of regions to realize region-level non-local enhancement. Here we show by a 2×2 grid for convenience.

density [37], frequency domain knowledge [6], etc., they are still weak in removing long rain streaks in complex background scenes as well as distinguishing dense rain streaks with similar image patterns in rain-free ones. This is due to the fact that convolutions in existing CNN-based de-raining neural networks are inherently local operations with small range of spatial receptive field in each computation.

2.2 Non-local Networks

Very recently, non-local neural network is proposed to realize the computation of long-range dependencies [32] for video classification task. In each 2D non-local operation, the response at a position is computed as a weighted sum of the features at all spatial positions. This is the first component of neural network that is able to enlarge the receptive field from neighbor positions to the entire image. Interestingly, such non-local operation is majorly inspired by traditional non-local mean filtering which is also exploited in early single image de-raining method [14]. This verifies importance of the non-local enhancement in our proposed de-raining CNN engine. As far as we know, our proposed non-locally enhanced CNN architecture is the first piece of work that attempts to incorporate non-local mean calculation into a fully convolutional neural network architecture with correlation propagation for the task of pixel-wise image restoration.

3 METHOD

We introduce an end-to-end convolutional neural network for single image de-raining, called non-locally enhanced encoder-decoder network (NLEDN). Our framework contains a fully convolutional encoder-decoder network which has been proven able to learn complex pixel-wise mappings from large amount of input-output image pairs. The overall architecture of the proposed network is illustrated in Figure 2. Particularly, in order to exploit the abundant structure cues in rain streak maps and the self-similarities in rain-free nature scenes, we propose the non-locally enhanced dense block (NEDB) as the basic component in our network architecture. We carefully integrate NEDBs with both encoding and decoding layers to enable the computation of long-range spatial dependencies as well as efficient usage of the feature activation of proceeding layers. In the following sections, we introduce each component of the proposed architecture with more detail.

3.1 Entrance and Exit Layers

The proposed de-raining NLEDN takes one image with rain-streaks as input in the entrance and outputs its rain-free version in the exit. In this section, we focus on the network structure of the entrance and exit of our entire framework.

Shallow Feature Extraction In the very beginning of the whole architecture, we use two convolution layers to extract shallow features of the input rainy image, as shown in Figure 2. Formally, we have

$$F_0 = H_0(I_0), \tag{1}$$

where I_0 and $H_0(\cdot)$ denote the input rainy image and the first shallow feature extraction convolution layer, respectively. As shown in Figure 2, using the long-range skip-connections which bypass intermediate layers, we link both the input image I_0 and the shallow features F_0 with layers that are close to the exit of the whole network. The benefits of such skip-connections here are two folds: first, they provide long-range information compensation such that raw pixel values and low level feature activation are still available in the very end of the whole architecture; Second, they enable the residual learning that facilitates gradient back-propagation and the pixel-wise prediction. Next, the shallow features F_0 is fed into the second convolution layer $H_1(\cdot)$ to obtain shallow features F_1:

$$F_1 = H_1(F_0), \tag{2}$$

F_1 is used as the input to the subsequent encoding layers.

Exit layers As shown in Figure 2, the input image I_0 and the shallow features F_0 are gradually added to the feature activation of layers near the exit of the whole architecture. Particularly, one tanh layer is adopted as the nonlinear unit of the final convolution layer to obtain a rain map R with pixel value within $(-1, 1)$. The final rain-free image $\hat{Y}$ can be computed via

$$\hat{Y} = I_0 + R, \tag{3}$$

Noted that the architecture of entrance and exit layers here are not unique, we choose such architecture in order to effectively incorporate our core modules, the non-locally enhanced encoding and decoding layers, which will be elaborated in subsequent sections.

3.2 Non-locally Enhanced Encoding and Decoding

Conventional encoding-decoding networks are widely used in image-to-image translation or other pixel-wise prediction tasks. Here, the proposed architecture can be regarded as an enhanced version with non-local operations and dense connections. To achieve this, a novel non-locally enhanced dense block (NEDB) is plugged into each stage of both the encoder and the decoder.

Non-locally Enhanced Dense Block (NEDB) The detailed architecture of the proposed NEDB is illustrated in Figure 3. Specifically, we denote the input feature activation to the NEDB as F_n, which has the spatial dimension of $H_n \times W_n \times C_n$. The pair-wise function f that calculates the pair-wise relationship is defined as

$$f(F_{n,i}, F_{n,j}) = \theta(F_{n,i})^T \phi(F_{n,j}) \tag{4}$$

where $F_{n,i}, F_{n,j}$ denote the feature activation F_n at position i, j respectively. $\theta(\cdot)$ and $\phi(\cdot)$ are two feature embedding operations with different learned parameters W_θ and W_ϕ, denoted as $\theta(F_{n,i}) = W_\theta F_{n,i}$ and $\phi(F_{n,i}) = W_\phi F_{n,i}$. Following [32], we define the non-local operation in NEDB as

$$y_{n,i} = \frac{1}{C(F)} \sum_{\forall j} f(F_{n,i}, F_{n,j}) g(F_{n,j}) \tag{5}$$

where $g(\cdot)$ is the unary function that computes the representation of F_n while $C(F)$ is the normalization factor, defined as $C(F) = \sum_{\forall j} f(F_{n,i}, F_{n,j})$. In this way, the feature representation is non-locally enhanced via considering all positions ($\forall j$) for each location i.

Next, the non-locally enhanced feature activation is fed to five consecutive convolution layers which are densely connected. Specifically, following [11], we adopt direct connections from each layer to all subsequent layers. The architecture is shown in Figure 3. Hence, the l^{th} layer receives the feature activations from all preceding layers, D_0,...,D_{l-1} as input:

$$D_l = H_l([D_0, ..., D_{l-1}]) \tag{6}$$

where $[D_0,...,D_{l-1}]$ denotes the concatenation of the feature activations produced in layer 0,...,$l-1$.

Moreover, to avoid the notorious problem of gradients vanishing/exploding caused by an increase in the number of network layers and connections, we adopt local residual learning in the design of each NEDB. Formally, the final output of the m-th NEDB can be achieved by

$$F_m = \mathcal{F}(F_{m-1}, W_m) + F_{m-1}, \tag{7}$$

where $\mathcal{F}(F_{m-1}, W_m)$ represents the residual mapping to be learned in the considered block, which is actually inferred from a concatenation of feature activations from all preceding layers with a 1 × 1 convolution layer, as illustrated in Fig. 3 and formally written as

$$\mathcal{F}(F_{m-1}, W_m) = \mathrm{Conv}_{1\times1}([D_0, ..., D_L]) \,. \tag{8}$$

Pooling Indices Guided Decoding As shown in Figure 2, the encoding part consists of three consecutive NEDBs, each of which is followed by one max-pooling layer with striding that downsamples the feature activation. Symmetrically, another three NEDBs are stacked in the decoding part, each of which is followed by one max-unpooling layer that upsamples the feature activation. Moreover, skip-connections are utilized to link feature activations of encoding layers to their counterpart in the decoding layers.

Particularly, we propose to record pooling indices [1] during encoding for further upsample inferring in the decoding stage. The max-unpooling layer uses the pooling indices computed in the max-pooling step of the corresponding encoding layer to perform non-linear upsampling. Given the recorded pooling index matrix, the output feature map of max-unpooling is calculated by first initializing to the size before the max pooling operation, and then assigning the feature column at each position of the input feature map to the corresponding position (given by the index matrix) and zeroing the remaining positions. The upsampled feature activations are sparse and convolved with subsequent trainable filters to produce dense feature maps. We will show by experiments that such pooling indices guided decoding scheme is more suitable here for rain streaks removal compared with conventional bilinear upsampling.

Multi-scale Non-Local Enhancement With the above mentioned max-pooling and max-unpooling operations, the spatial size of intermediate feature activations gradually decreases in the encoding stage while gradually increases during decoding. Therefore, as the non-local operation in NEDB requires to compute pair-wise relations between every two spatial positions of the feature activation map, the computation burden increases dramatically when spatial dimension gets larger. To address this problem and construct a more flexible non-local enhancement across feature activations with different spatial resolution, we implement the non-local operation in a multi-scale manner when building the encoding and decoding layers.

Specifically, for the feature activation with lowest spatial resolution (e.g. F4 in Figure 2), the subsequent NEDB directly works on the whole feature activation map to realize a global-level non-local enhancement. For the feature activation with higher spatial resolution, we first divide it into a grid of regions (As shown in Fig. 2, the $k \times k$ NEDB indicates how the input feature map is divided before performing region-wise nonlocal operation. e.g. F_1 is divided into a grid of 8×8) and then let the subsequent NEDB work on the feature activation inside each region. Accordingly, such region-level non-local enhancement is able to prevent unacceptable computational consumption caused by directly working on high resolution feature activation. One the other hand, comparing with conventional local convolution operation which operates inside 3×3 or 5×5 windows, region-level non-local enhancement is able to retrieve long-range structural cues to facilitate better rain-streaks removal.

3.3 Loss Function

The loss function is defined as the mean absolute error (MAE) between the resulted rain-free image and its corresponding groundtruth, which is formulated as follow:

$$\mathcal{L} = \frac{1}{HWC} \sum_{i} \sum_{j} \sum_{k} \|\hat{Y}_{i,j,k} - Y_{i,j,k}\|_1, \tag{9}$$

where H, W and C denotes the height, width and channel number of the rain-free image. $\hat{Y}$ and Y denote the predicted rain-free image and ground-truth respectively.

4 EXPERIMENTAL RESULTS

4.1 Datasets and Evaluation Criteria

We evaluate the performance of our proposed method (NLEDN) on both synthetic datasets and real data. For synthetic data, we use four benchmark datasets, including the dataset provided by Fu *et al.* [6], denoted as DDN-Data; the dataset synthesized by Zhang *et al.* [37], denoted as DIDMDN-Data; the Rain100L and the Rain100H dataset provided in [35]. Specifically, the DDN-Data contains 14,000 rainy/clean image pairs, which is synthesized from 1,000 clean images with 14 kinds of different rain-streak orientations and magnitudes. Following Fu *et al.* [6], we select 9,100 pairs of them for training and the remaining 4,900 image pairs for evaluation. The DIDMDN-Data consists of 12,000 image pairs of three rain density levels (i.e. light, medium and heavy). There are roughly 4,000 images per rain-density level in the dataset. The rain-density labels are also provided and are used as extra data for model training in [37]. Noted that we did not use this extra information in our model. Rain100L is the synthesized dataset selected from BSD200 [26] with only one type of rain streaks, which consists of 200 image pairs for training and the other 100 images for testing. Compared with Rain100L, Rain100H is more challenging. It is synthesized with five streaks directions and contains 1,800 images for training plus 100 images for testing. As pointed out in [35], although some of the synthesized examples in Rain100H are inconsistent with real images, adding these data as training can further enhance the robustness of the network. For real data, we collected some of the images from the Internet and some from the released images of [37]. We have also taken some real cases using our own cameras for testing.

We evaluate the performance of the synthesized data using two metrics, including Peak Signal-to-Noise Ratio (PSNR) [12] and Structure Similarity Index (SSIM) [33]. As with existing works [35, 37], we evaluate the results in the luminance channel (i.e. Y channel of YCbCr space), which has the most significant impact on the human visual system. As the rain-free groundtruth are not available on real-world image, we evaluate the performance on real data singly based on visual comparison.

4.2 Implementation

Our proposed NLEDN has been implemented on the Pytorch framework, a flexible open source deep learning frame network. During training, we use horizontal flipping for data augmentation and resize the image to have long side smaller than 512. As the network is fully convolutional, and we set the mini-batch size to 1, the size of the input image does not have to be the same. We use adam optimizer to update the parameters of network during training. The learning rate is initially set to 0.0005, and we reduce it by 10% when the training loss stops decreasing, until 0.0001. We use a weight decay of 0.0001 and a momentum of 0.9. Because of the large differences in the size of each datasets, the time spent on training each specific model is different. It takes around 3.5 days to train a model using the training set of DDN-Data or DIDMDN-Data, and it cost around 15 hours for training on Rain100H dataset. For Rain100L dataset, it is much faster and only takes about four hours to complete a whole model training. Therefore, we conduct ablation study on Rain100L dataset in our experiment. However, as our entire model is fully convolutional, the testing process is very efficient, which only takes 1.44 seconds for the trained model to process a testing image with 512 × 512 pixels on a PC with an NVIDIA X GPU and a 3.4GHz Intel processor.

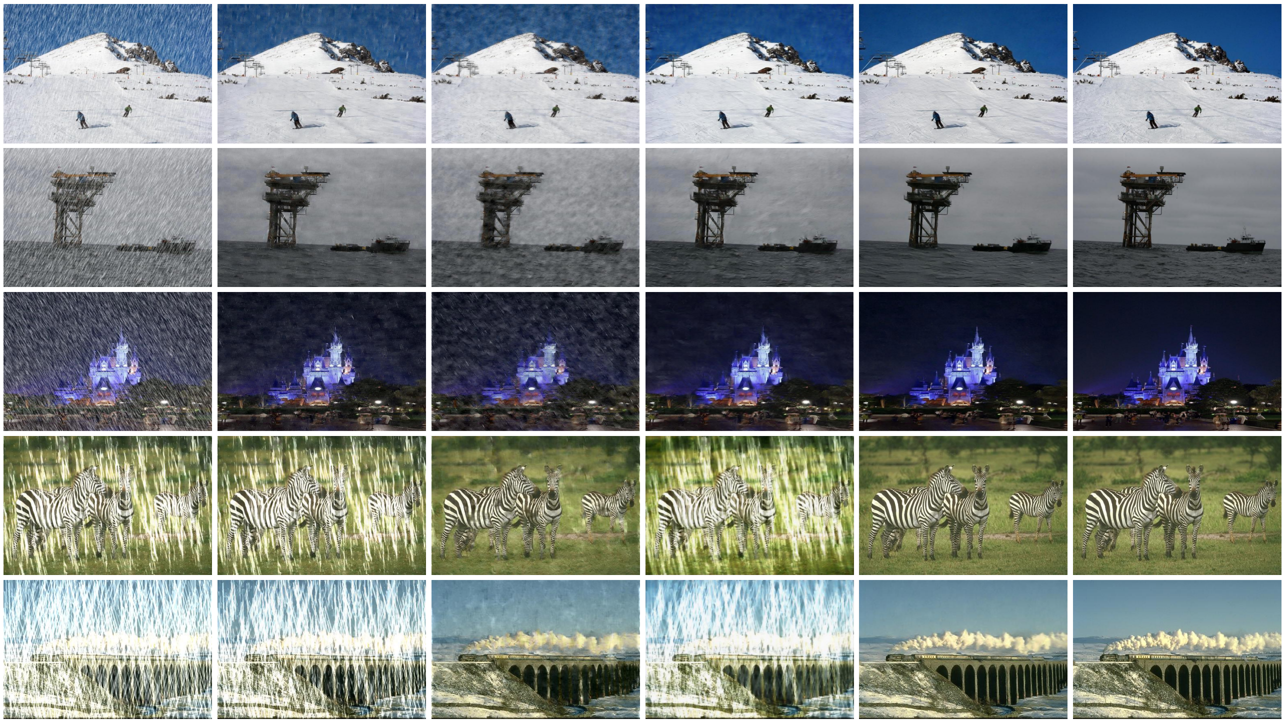
4.3 Comparison with the state-of-the-art

We compare our proposed NLEDN method against five state-of-the-art single-image de-raining methods, including discriminative sparse coding (DSC) [25], GMM-based layer prior (GMM) [24], deep detail network (DDN) [6], joint rain detection and removal (JORDER) [35] and density-aware single image de-raining using a multi-stream dense network (DID-MDN) [37]. The last three are the latest deep learning based methods. As all of these methods use different data in training their models, we trained four versions of our model based on the training set of the four synthesized dataset respectively. Moreover, for fair comparison, we have also fine-tuned the released deep model of the comparison methods on each specific training set before evaluation. When evaluating on real-world data, we test four versions of the models for each deep learning based methods (including our NLEDN) on each image and select the one with best visualization as the result for comparison.

Quantitative Evaluation. We report a quantitative comparison w.r.t PSNR and SSIM in Table 1. As can be observed, our proposed method (NLEDN) increases the PSNR metric achieved by the existing best-performing algorithms by an average of 1.07db,

Dataset	Metric	DSC [25] (ICCV'15)	GMM [24] (CVPR'16)	DDN [6] (CVPR'17)	JORDER [35] (CVPR'17)	DID-MDN [37] (CVPR'18)	Our NLEDN
DDN-Data	PSNR	22.03	25.64	**28.24**	**28.72**	26.17	**29.79**
	SSIM	0.7985	0.8360	**0.8654**	**0.8740**	0.8409	**0.8976**
DIDMDN-Data	PSNR	20.89	21.37	23.53	**30.35**	***28.30**	**33.16**
	SSIM	0.7321	0.7923	0.7057	**0.8763**	***0.8707**	**0.9192**
Rain100L	PSNR	23.39	28.25	25.99	***35.23**	**30.48**	**36.57**
	SSIM	0.8672	0.8763	0.8141	***0.9676**	**0.9323**	**0.9747**
Rain100H	PSNR	17.55	15.96	16.02	***25.21**	**26.35**	**30.38**
	SSIM	0.5379	0.4180	0.3579	***0.8001**	**0.8287**	**0.8939**

Table 1: Comparison of quantitative results in terms of PSNR and SSIM on four synthesized benchmark datasets. The three best performing algorithms are marked in red, blue, and green, respectively. Our proposed NLEDN consistently achieves the best performance. '$*$' indicates that the method uses additional data (e.g. rain density level, rain mask annotation) provided by the dataset.



(a) Input (b) DDN (c) JORDER (d) DID-MDN (e) Ours (f) Ground Truth

Figure 4: Visual comparison of rain-streaks removal results generated from state-of-the-art deep learning based methods (including our NLEDN) on synthesized rainy images. Our model consistently achieves the best visualization results in terms of effectively removing the rain streaks while preserving the image structure details.

2.81db, 1.34db and 4.03db respectively on DDN-Data, DIDMDN-Data, Rain100L and Rain100H. And at the same time, our model improves the SSIM by 2.70%, 4.90%, 0.73% and 7.87% respectively on the above four datasets. We can find that, on average, the rain-free images restored by our NLEDN are consistently closer to the groundtruth than existing state-of-the-art on the synthesized datasets, and the higher SSIM value also indicates that our method can better restore the structural information of an image. Moreover, the more complex the data set, the more significant the performance of our algorithm compared to existing algorithms. Noted that JORDER [35] use additionally provided rain mask and rain-streak annotation while training their models on Rain100L and Rain100H datasets and DID-MDN [37] use extra rain density level information while training on their DIDMDN-Data. Nonetheless, our proposed model can still greatly outperform their results without resorting to any additional data.

Qualitative Evaluation. Fig. 4 shows visual comparisons of rain-streaks removal results for five synthesized rainy images. The first three examples are synthetic heavy rain cases which are similar to real-world scenes, our models consistently achieves the best visualization results in terms of effectively removing the rain streaks while preserving the image structure details. The latter two are hard examples chosen from the Rain100H dataset, and may be rare in real world. As can be observed, both DDN [6] and DID-MDN [37]

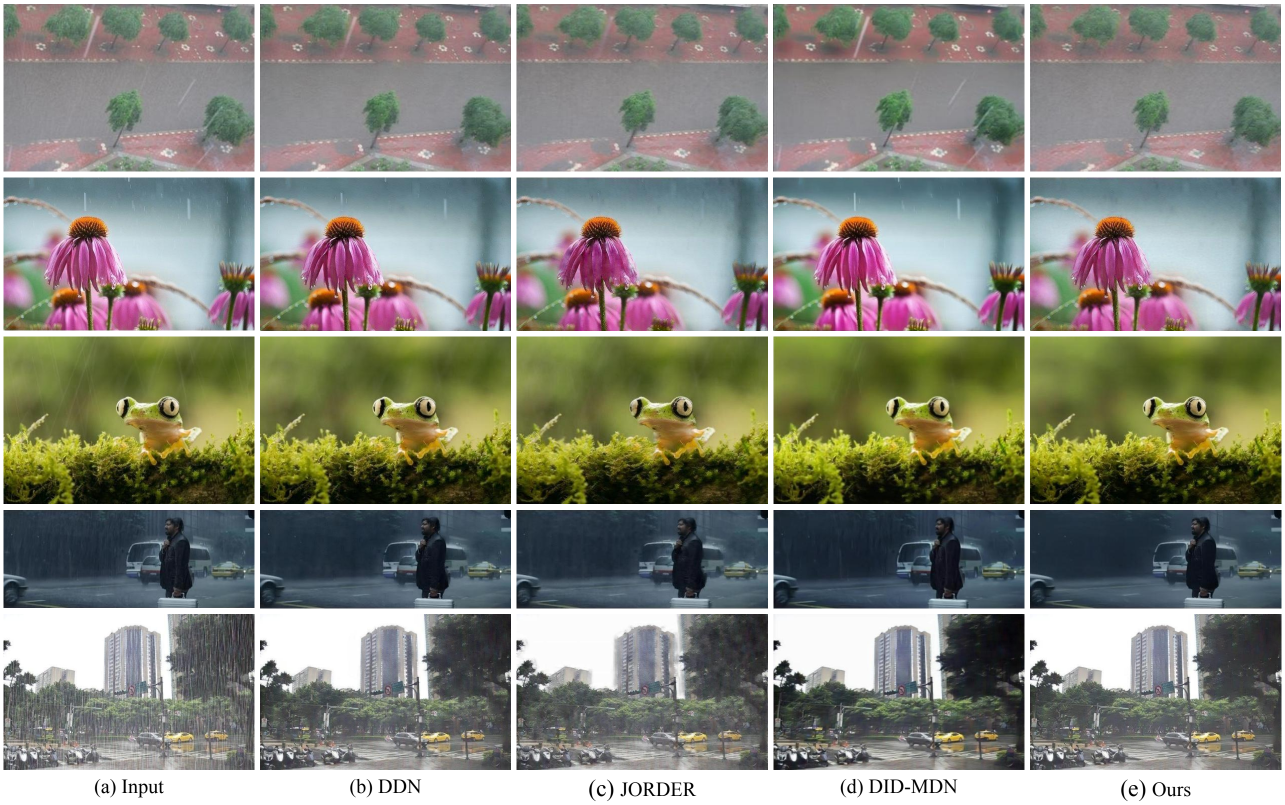


Figure 5: Visual comparison of rain-streaks removal results generated from state-of-the-art methods on real-world rainy images. Our model consistently achieves the best visualization results.

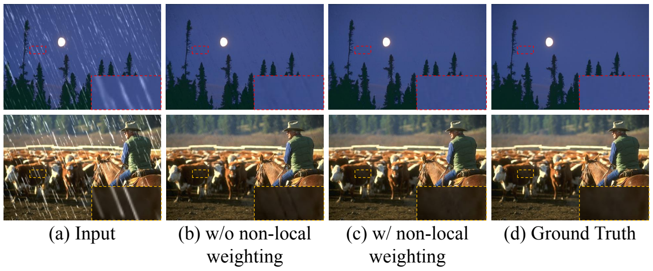


Figure 6: Visual comparison of rain-streaks removal results with and without non-local weighting enhancement in our proposed model.

are fail to remove rain streaks of these extreme cases though having been fine-tuned on this dataset. Although JORDER [35] is designed to handle such hard cases, the restored images are over-smoothing and full of artifacts. Our proposed NLEDN generates much cleaner results with promisingly structure details preserving as it is able to fully exploit the long-range spatial context information based on gradually increased size of receptive field and the non-locally weighting scheme. Fig. 5 demonstrates the results of some real images with rain-streaks in various rain densities. As can be observed, our proposed model shows the best visual performance on rain-streaks removal. It is particularly effective in removing long rain-streaks while perfectly preserving the image structural details. Though the existing best-performing DID-MDN [37] can remove long rain-streaks of high density to some extent, it suffers from artifacts and blurry if observed in a zoom-in view.

4.4 Ablation Study

As discussed in Section 3, our proposed NLEDN contains two core components towards more effective rain-streaks removal, including the encoder-decoder framework with pooling striding and max-unpooling operation, and the tailored non-locally enhanced dense block with densely connected convolution layers and a non-local feature weighting scheme. To validate the effectiveness and necessity of the internal network design of each of these two modules, we exhaustively compare NLEDN with its five variants trained and tested on the Rain100L dataset. The specific performance changes in terms of PSNR and SSIM are listed in Table 2.

R_a refers to the result of a very basic baseline which only includes the convolution layers of a single NEDB without dense connection or non-local weighting enhancement, as well as the same entrance and exit layer settings as NLEDN. Actually, it is a degenerate FCN based de-raining model with residual connection. It reaches the $PSNR = 33.23db$ and $SSIM = 0.9533$ which already outperforms

Methods	R_a	R_b	R_c	R_d	R_e	R_f
without dense connection?	✓					
single block?	✓	✓				
multiple blocks?			✓	✓	✓	✓
pooling striding?				✓		✓
pooling indices?				✓		✓
non-local operation?					✓	✓
PSNR	33.23	33.82	35.44	35.80	35.91	36.57
SSIM	0.9533	0.9596	0.9691	0.9716	0.9720	0.9747

Table 2: Ablation study on different components of our proposed non-locally enhanced encoder-decoder network framework.

the two recent deep learning based methods DDN [6] and DID-MDN [37] but inferior to JORDER [35]. R_b adds dense connection between convolutional layers to R_a. As shown in the table, adding dense connection leads to an average increase of 0.59db in terms of PSNR and 0.7% improvement on SSIM. This proves the effectiveness of dense connections. Due to space limitations, in subsequent ablation studies, we add dense connection to each NEDB by default and discuss on the role of other network components.

R_c is a simple concatenation of multiple dense blocks with neither non-local weighting nor receptive field controlling (pooling striding). For comparison, the number of blocks is set to the same as that of NLEDN. As shown in Table 2, simply concatenating multiple dense blocks can bring significant performance improvement, which increases the average PSNR by 1.62db while at the same time boosts the SSIM by 1% when compared to its single block version R_b. This verifies the effectiveness of deeper feature representation in rain-streaks removal. In our experiments, we find that a cascade of 6 dense blocks leads to the best performance in our validation. Concatenating more than 6 dense blocks even leads to a performance deterioration. R_d is directly modified on R_c by adding pooling striding and corresponding skip connection between blocks guided by pooling indices. As illustrated in the table, adding a pooling indices based receptive filed controlling scheme further leads to an increase of 0.36db in terms of PSNR and a 0.3% improvement on SSIM.

R_e and R_f focus on the effectiveness of introducing non-local feature weighting to the network design. Firstly, we directly add multi-scale non-local enhancement to R_c and observe a performance boost on PSNR from 35.44db to 35.91db and an increase of 0.3% in terms of SSIM. The performance improvement is more significant when adding non-local operation to R_d, which forms our full model R_f (R_f = NLEDN). As shown in the table, the introduction of non-local operation contributes an average of 0.77db and 0.32% improvement in terms of PSNR and SSIM respectively. This verifies the effectiveness and universality of non-local optimization for rain-streaks removal on single images. A visual comparison is provided in Fig. 6. As can be seen, the model without non-local enhancement suffers from some obvious artifacts and shows powerless for most of the long rain streaks. This further proves the effectiveness of long-distance dependencies modeling in rain-streaks removal.

5 CONCLUSION

In this paper, we have introduced a non-locally enhanced encoder-decoder network framework for rain streaks removal from single images. It is designed as a concatenation of an encoder network followed by a corresponding decoder network, which are both composed of a series of tailored, non-locally enhanced dense blocks (NEDB). The NEDB is designed to not only fully exploit hierarchical features from densely connected convolutional layers but also well capture the long-distance dependencies and structural information by employing a non-locally weighting operation at a specific range of feature maps. Experimental results on both synthetic and real datasets have demonstrated that our proposed method can effectively remove rain-streaks on rainy image of various density while promisingly preserve the image texture similar to the rain streaks, which greatly outperforms the state-of-the-art. In our future research work, we plan to extend the proposed algorithm to a wider range of image restoration tasks, including but not limited to image denoising, image dehazing and image super-resolution.

REFERENCES

[1] V Badrinarayanan, A Kendall, and R Cipolla. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. *IEEE Transactions on Pattern Analysis Machine Intelligence* PP, 99 (2017), 2481–2495.

[2] JÃľrÃťmie Bossu, Nicolas HautiÃĺre, and Jean Philippe Tarel. 2011. Rain or Snow Detection in Image Sequences Through Use of a Histogram of Orientation of Streaks. *International Journal of Computer Vision* 93, 3 (2011), 348–367.

[3] Qingxing Cao, Liang Lin, Yukai Shi, Xiaodan Liang, and Guanbin Li. 2017. Attention-Aware Face Hallucination via Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1656–1664.

[4] Yi Lei Chen and Chiou Ting Hsu. 2014. A Generalized Low-Rank Appearance Model for Spatio-temporally Correlated Rain Streaks. In *IEEE International Conference on Computer Vision.* 1968–1975.

[5] C. Dong, C. C. Loy, K. He, and X. Tang. 2016. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis Machine Intelligence* 38, 2 (2016), 295–307.

[6] 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.

[7] 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.*

[8] Kshitiz Garg and Shree K Nayar. 2005. When does a camera see rain?. In *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2. 1067–1074.

[9] 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.

[10] De-An Huang, Li-Wei Kang, Min-Chun Yang, Chia-Wen Lin, and Yu-Chiang Frank Wang. 2012. Context-aware single image rain removal. In *Proceedings of the IEEE International Conference on Multimedia and Expo.* 164–169.

[11] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. 2016. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

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

[13] L. W. Kang, C. W. Lin, and Y. H. Fu. 2012. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*

A Publication of the IEEE Signal Processing Society 21, 4 (2012), 1742–1755.

[14] Jin Hwan Kim, Chul Lee, Jae Young Sim, and Chang Su Kim. 2014. Single-image deraining using an adaptive nonlocal means filter. In *IEEE International Conference on Image Processing*. 914–917.

[15] 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), 2658–70.

[16] Hiroyuki Kurihata, Tomokazu Takahashi, Ichiro Ide, Yoshito Mekada, Hiroshi Murase, Yukimasa Tamatsu, and Takayuki Miyahara. 2005. Rainy weather recognition from in-vehicle camera images for driver assistance. In *Proceedings of the IEEE Intelligent Vehicles Symposium*. 205–210.

[17] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep laplacian pyramid networks for fast and accurate superresolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[18] Christian Ledig, Zehan Wang, Wenzhe Shi, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, and Alykhan Tejani. 2016. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. (2016), 105–114.

[19] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. 2017. Instance-level salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 247–256.

[20] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. 2018. Flow Guided Recurrent Neural Encoder for Video Salient Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3243–3252.

[21] Guanbin Li and Yizhou Yu. 2016. Visual saliency detection based on multiscale deep CNN features. *IEEE Transactions on Image Processing* 25, 11 (2016), 5012–5024.

[22] Guanbin Li and Yizhou Yu. 2018. Contrast-Oriented Deep Neural Networks for Salient Object Detection. *IEEE Transactions on Neural Networks and Learning Systems* (2018).

[23] Haofeng Li, Guanbin Li, Liang Lin, and Yizhou Yu. 2017. Context-Aware Semantic Inpainting. *arXiv preprint arXiv:1712.07778* (2017).

[24] 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.

[25] Yu Luo, Yong Xu, and Hui Ji. 2015. Removing Rain from a Single Image via Discriminative Sparse Coding. In *IEEE International Conference on Computer Vision*. 3397–3405.

[26] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2. 416–423.

[27] 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.

[28] Minmin Shen and Ping Xue. 2011. A fast algorithm for rain detection and removal from videos. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 1–6.

[29] Assaf Shocher, Nadav Cohen, and Michal Irani. 2018. "Zero-Shot" Super-Resolution using Deep Internal Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[30] Shao-Hua Sun, Shang-Pu Fan, and Yu-Chiang Frank Wang. 2014. Exploiting image structural similarity for single image rain removal. In *Proceedings of IEEE International Conference on Image Processing*. 4482–4486.

[31] Alexandru Vasile, Irina Vasile, Adrian Nistor, Luige Vladareanu, Mihaela Pantazica, Florin Caldararu, Andreea Bonea, Andrei Drumea, and Ioan Plotog. 2010. Rain sensor for automatic systems on vehicles. In *Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies V*, Vol. 7821. International Society for Optics and Photonics, 78211W.

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[33] 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.

[34] Xinwei Xue, Xin Jin, Chenyuan Zhang, and Satoshi Goto. 2012. Motion robust rain detection and removal from videos. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing*. 170–174.

[35] 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*.

[36] Fisher Yu and Vladlen Koltun. 2015. Multi-Scale Context Aggregation by Dilated Convolutions. (2015).

[37] 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*.

[38] Wei Zhang, Chao-Wei Fang, and Guan-Bin Li. 2017. Automatic Colorization with Improved Spatial Coherence and Boundary Localization. *Journal of Computer Science and Technology* 32, 3 (2017), 494–506.

[39] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. 2006. Rain removal in video by combining temporal and chromatic properties. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 461–464.

[40] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[41] Lei Zhu, Chi Wing Fu, Dani Lischinski, and Pheng Ann Heng. 2017. Joint Bi-layer Optimization for Single-Image Rain Streak Removal. In *IEEE International Conference on Computer Vision*. 2545–2553.