Removing Rain from Single Images via a Deep Detail Network

Xueyang Fu[1] Jiabin Huang[1] Delu Zeng[2] Yue Huang[1] Xinghao Ding[1] John Paisley[3]

[1]Xiamen University, China [2]South China University of Technology, China [3]Columbia University, USA

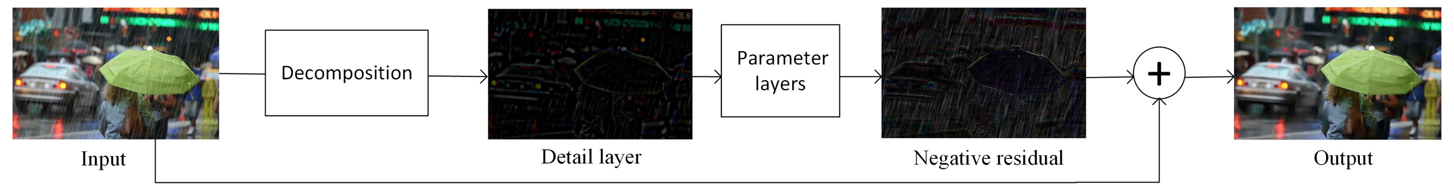
Testing Code and dataset: http://smartdsp.xmu.edu.cn/cvpr2017.html or https://github.com/XMU-smartdsp/Removing_Rain

IEEE 2017 Conference on Computer Vision and Pattern Recognition

CVPR July 21-26 Honolulu 2017

Objective

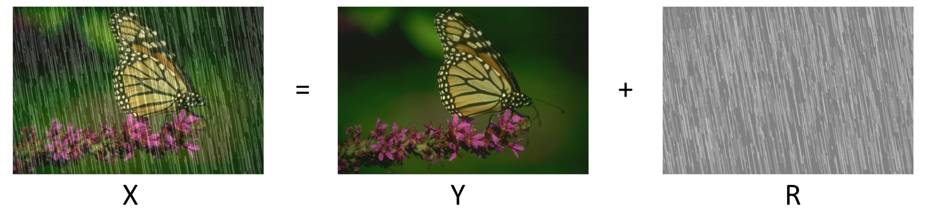
Removing rain streaks from single images without any additional information. The proposed framework for single-image rain removal:



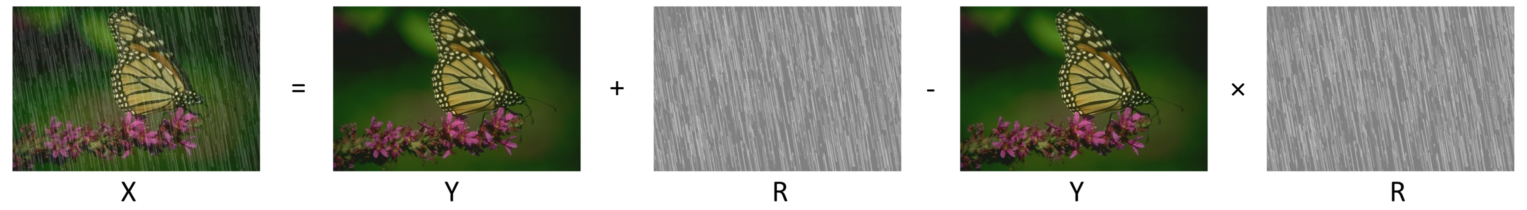
Analysis & Motivation

Assuming **X** denotes a rainy image, **Y** denotes a clean image and **R** denotes a rain streaks image.

1. linear additive composite model:



2. screen blend model (more realistic but difficult to handle) [1]:



To address this difficult problem, we utilize the powerful non-linear modeling ability of deep learning. Previous deep learning based methods aim to directly minimize the objective function

$$\mathcal{L} = \sum_i \|h(\mathbf{X}_i) - \mathbf{Y}_i\|_F^2, \quad (1)$$

where h is the network, F is the Frobenius norm, i is the ith training data.

Our observations:

1. **Negative residual mapping (neg-mapping)** can reduce the solution space by compressing the mapping range and make the learning process much easier.

$$\mathcal{L} = \sum_i \|h(\mathbf{X}_i) + \mathbf{X}_i - \mathbf{Y}_i\|_F^2 = \sum_i \|h(\mathbf{X}_i) - (\mathbf{Y}_i - \mathbf{X}_i)\|_F^2. \quad (2)$$

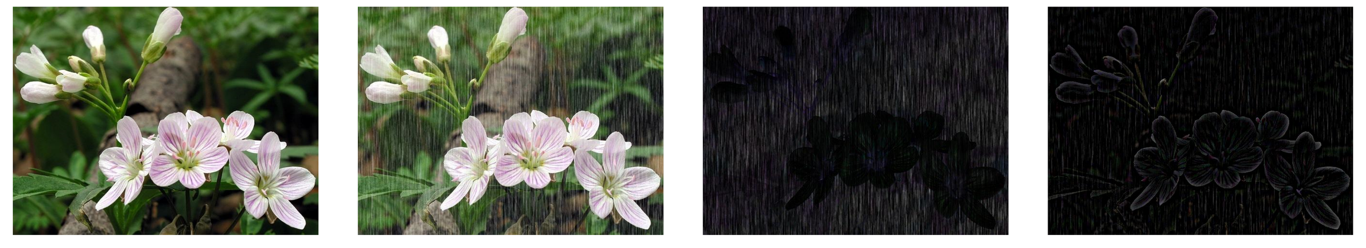
2. Using **detail image** can further improves the de-raining quality due to the sparsity.

$$\mathbf{X} = \mathbf{X}_{\mathrm{detail}} + \mathbf{X}_{\mathrm{base}}. \quad (3)$$

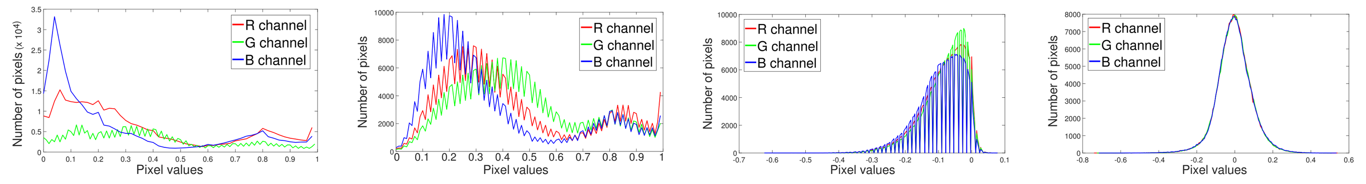
Since we train the network on the detail layer, we refer to this as a "**deep detail network**". Our final objective function is

$$\mathcal{L} = \sum_{i=1}^N \|f(\mathbf{X}_{i,\mathrm{detail}}, \mathbf{W}, \mathbf{b}) + \mathbf{X}_i - \mathbf{Y}_i\|_F^2. \quad (4)$$

where N is the number of training images, $f(\cdot)$ is ResNet, $\mathbf{W}$ and $\mathbf{b}$ are network parameters.



(a) Clean image $\mathbf{Y}$ (b) Rainy image $\mathbf{X}$ (c) $|\mathbf{Y} - \mathbf{X}|$ (d) Detail layer $\mathbf{X}_{\mathrm{detail}}$



(e) Histogram of $\mathbf{Y}$ (f) Histogram of $\mathbf{X}$ (g) Histogram of $\mathbf{Y} - \mathbf{X}$ (h) Histogram of $\mathbf{X}_{\mathrm{detail}}$

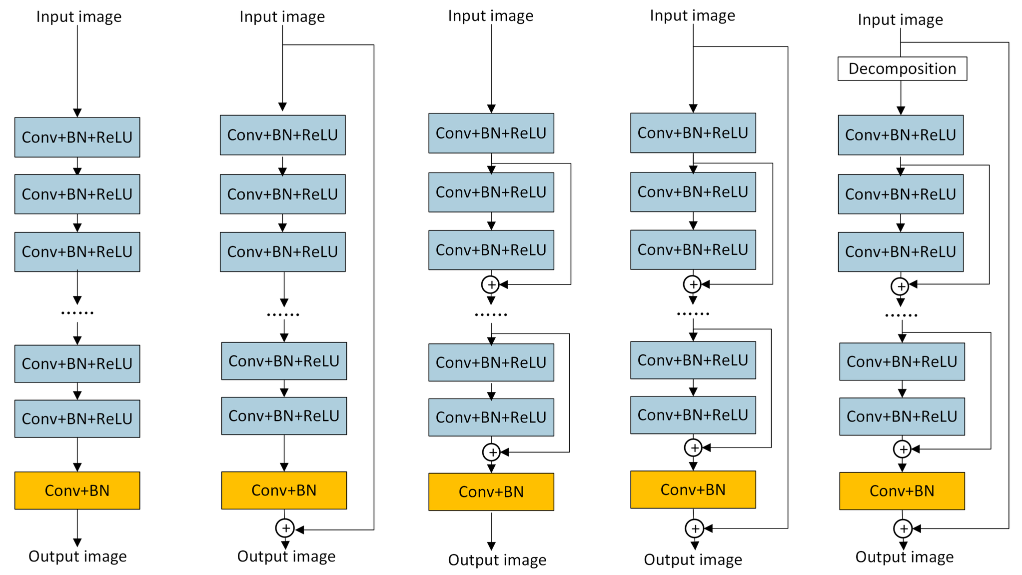
Figure: Range reduction and sparsity of the negative residual and detail layer.

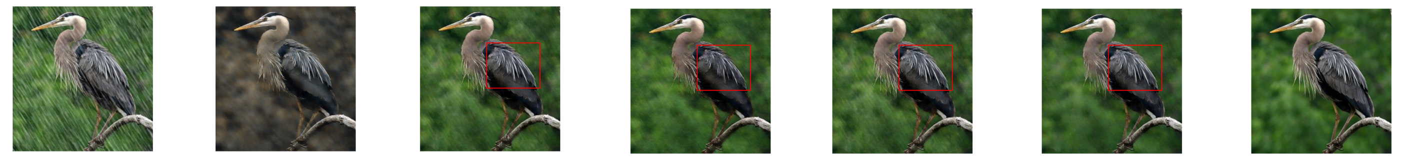
Contributions

1. We use a "neg-mapping" defined to be the difference between clean and rainy images. Predicting the residual can significantly reduce the mapping range and make the learning process much easier.
2. We exploit a priori knowledge and use the sparse detail layer as the input. We find that the sparsity can further improve the de-raining quality.
3. We create and use a synthetic dataset of 14,000 rainy/clean image pairs to train our network. Although the network is trained on synthetic rain data, we find that it generalizes very well to real-world rainy images.

Network

Network structures:





(a) Rainy image (b) Direct network (c) Neg-mapping (d) ResNet (e) Neg-map+ResNet (f) Final network (g) Ground truth

Figure: De-rained results of different network structures. SSIM of (a)-(f) are 0.774, 0.490, 0.936, 0.926, 0.938 and **0.940**, respectively. All network depths are set to 26.

Training convergence:

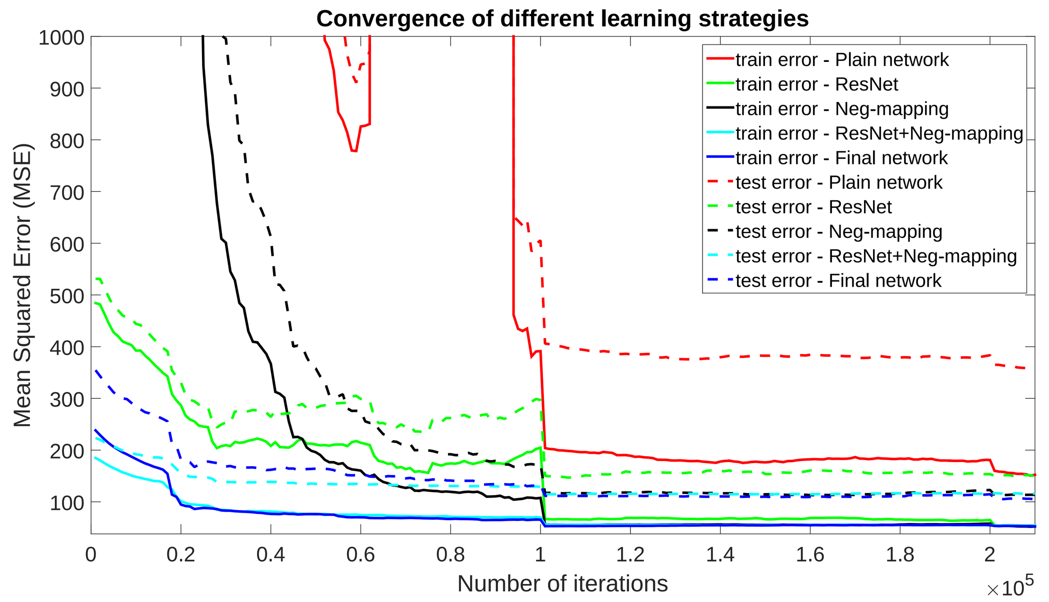


Figure: The drops at 10^5 and 2×10^5 iterations are due to the scheduled learning rate division.

Parameters: To balance the trade-off between performance and speed, we chose network depth = 26, filter sizes = 3 and filter numbers = 16.

Table: Average SSIM using different network sizes.

	filter numbers = 16	filter numbers = 32	filter numbers = 64
depth = 14	0.906	0.912	0.915
depth = 26	0.916	0.920	0.920
depth = 50	0.921	0.926	0.928

Training data: We collect 1,000 clean images [5, 6] and each one was used to generate 14 rainy images with different streak orientations and magnitudes. Our dataset contains 14,000 pairs of rainy/clean images. We randomly selected 9,100 images from which we generated 3 million 64 × 64 rainy/clean patch pairs. The remaining 4,900 image pairs are used to evaluate the trained network.

Experiments

All experiments are performed on a PC with Intel(R) Xeon(R) CPU E5-2670, 64GB RAM and GTX TITAN X.

Synthetic test data:

(a) Ground truth (b) Rainy images (c) Method [1] (d) Method [2] (e) Our results

Figure: Three synthetic images with different orientations and magnitudes: "girl", "flower", "umbrella".

Table: Quantitative measurement results using SSIM [3] on synthesized test images.

Images	Ground truth	Rainy image	Method [1]	Method [2]	Ours
girl	1	0.65	0.71	0.80	**0.90**
flower	1	0.69	0.77	0.81	**0.92**
umbrella	1	0.75	0.80	0.82	**0.86**
4,900 test images	1	0.78 ± 0.12	0.83 ± 0.09	0.87 ± 0.07	**0.90 ± 0.05**

Running time:

Table: Comparison of test running time (seconds).

Image size	[1]	[2]	Ours (CPU)	Ours (GPU)
250 × 250	54.9	169.6	1.9	0.2
500 × 500	189.3	674.8	6.3	0.3
750 × 750	383.9	1468.7	12.6	0.5

Comparison with ResNet [7]:

Table: Average SSIM on 100 synthetic images.

depth	8	14	20	26	50
ResNet [7]	0.896	0.904	0.909	0.907	**0.917**
Ours	0.896	0.906	0.915	**0.916**	0.921

Real-world test data:

(a) Rainy images (b) Method [1] (c) Method [2] (d) Our results

Figure: Three results on real-world rainy images: "street", "people" and "car".

Heavy rain images: When dealing with heavy rain images can become hazy. We found that applying a dehazing method [4] as pre-processing is useful.

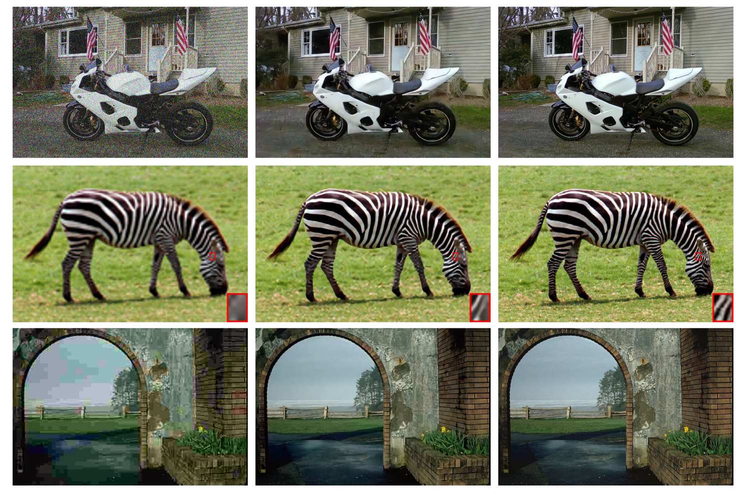
(a) Rainy images (b) De-hazed (a) (c) De-rained (b)

Figure: An example of heavy rain removal with dehazing.

Extension

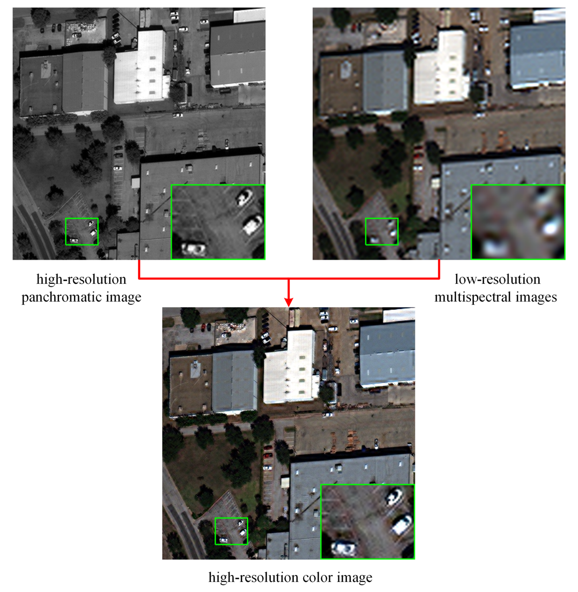
Our network is actually a general framework for several low-level image processing tasks.

Ordinary image processing tasks: Denoising, super-resolution, JPEG artifacts reduction, etc.



(a) Inputs (b) Ours (c) Ground truth

Pan-sharpening:



References

[1] Y. Luo et al. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 2015.

[2] Y. Li et al. Rain streak removal using layer priors. In *CVPR*, 2016.

[3] Z. Wang et al. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[4] W. Ren et al. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, 2016.

[5] G. Schaefer and M. Stich. UCID: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia*, 2003.

[6] P. Arbelaez et al. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):898–916, 2011.

[7] K. He et al. Deep residual learning for image recognition. In *CVPR*, 2016.

Contact Information

Web: http://smartdsp.xmu.edu.cn

Corresponding author: Xinghao Ding (dxh@xmu.edu.cn)

Address: Room 504, Administration Building #C, Haiyun Park, Xiamen University, Fujian 361005, China

QR code of our lab