

R²N: A Novel Deep Learning Architecture for Rain Removal from Single Image

Yecai Guo^{1,2,*}, Chen Li^{1,2} and Qi Liu³

Abstract: Visual degradation of captured images caused by rainy streaks under rainy weather can adversely affect the performance of many open-air vision systems. Hence, it is necessary to address the problem of eliminating rain streaks from the individual rainy image. In this work, a deep convolution neural network (CNN) based method is introduced, called Rain-Removal Net (R²N), to solve the single image de-raining issue. Firstly, we decomposed the rainy image into its high-frequency detail layer and low-frequency base layer. Then, we used the high-frequency detail layer to input the carefully designed CNN architecture to learn the mapping between it and its corresponding de-rained high-frequency detail layer. The CNN architecture consists of four convolution layers and four deconvolution layers, as well as three skip connections. The experiments on synthetic and real-world rainy images show that the performance of our architecture outperforms the compared state-of-the-art de-raining models with respects to the quality of de-rained images and computing efficiency.

Keywords: Deep learning, convolution neural networks, rain streaks, single image de-raining, skip connection.

1 Introduction

In recent years, with the development of computer, network communication, image processing electronic and other related fields, the intelligent video surveillance system which based on video and image processing has achieved many promising progresses. The system plays an extremely role in maintaining public security. In such condition, some computer vision tasks that related to the intelligent surveillance system have become hot research topics, such as target tracking, track identity, and object detection [Garg and Nayar (2004)] etc. The algorithms designed for these tasks can work well on the high visibility image or video data, but obviously deteriorative when they are facing low-quality data. This is because of most of these algorithms are always trained on the high-quality dataset. However, in the real scene, bad weather such as rainy day is always not inevitable, which seriously degrades the color and content of the captured video and

¹School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing Jiangsu, 210044, China.

²Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing Jiangsu, 210044, China.

³School of Computing, Edinburgh Napier University, Edinburgh, EH10 5DT, UK.

* Corresponding Author: Yecai Guo. Email: guo-yecai@163.com.

image data. This condition may occur if some safe accidents happened in public areas are recorded by surveillance cameras or mobile phones in rainy days. In this circumstance, the captured video or image data contains a large number of rain streaks, which leads to the refractions and reflections of important contents and distorts the image signal as well as reduces the image quality. These negative impacts caused by rain streaks on the captured images or videos may significantly affect the performances of many vision-oriented algorithms mentioned above [Huang, Kang, Wang et al. (2014); Kang, Lin and Fu (2012)]. In order to maintain the performances of the security-related outdoor monitoring systems and to enhance the visual quality of the degraded images, it is essential to remove rain streaks automatically from the single images captured under a rainy condition.

Similar to the image denoising problem, the de-raining task can be formulated as:

$$I - R = D \quad (1)$$

where I , R and D represent the rainy image, rain streaks and clear image, respectively.

In the past few years, many researchers have paid attention to address the de-raining issue and have proposed many efficient algorithms. According to different concerns, these proposed algorithms can be roughly divided into two categories, i.e., video-based algorithms and single-image based algorithms. Video-based methods recover rainy images from the video sequences [Bossu, Hautière and Tarel (2011); Brewer and Liu (2008)]. Single-image based methods consider the rain removal issue from the single image angle, by regarding the issue as an image layer decomposition problem [Chen and Hsu (2013); Huang, Kang, Wang et al. (2014); Kang, Lin and Fu (2012); Sun, Fan and Wang (2014)], or by learning an end-to-end mapping between the rainy images and their corresponding ground truths. These single-image based methods mainly including structural similarity constraints [Sun, Fan and Wang (2014)], generalized low-rank model [Chen and Hsu (2013)], morphological component analysis with sparse coding [Huang, Kang, Wang et al. (2014); Kang, Lin and Fu (2012)], and diffusion-based or filter-based method, e.g. the nonlocal means filter [Kim, Lee, Sim et al. (2013)].

Although the above-mentioned methods have achieved varying degrees of success, however, most of them exist several limitations due to the following aspects: (1) For the underlying operation of many methods is performed on a local receptive field or a small image patch, hence, the spatial information between the different local receptive fields or different image patches, which is important for de-raining, is generally neglected. (2) Because of the rain streaks and images texture are internally overlapping, texture details of non-rain regions may be removed by most methods, which leads to the recovered rainy image exists some over-smoothness regions [Huang, Kang, Yang et al. (2012)]. In order to overcome the limitations above, in this work, we aimed at designing a novel convolution neural network (CNN) that can remove rain streaks and keep the details of restored clean images simultaneously. Specially, we investigate a CNN model named "Rain-removal Net (R²N)" for the single image de-raining. The ideas of the model are as follows:

(1) Similar to the work of Kang et al. [Kang, Lin and Fu (2012)], the raw rainy image is decomposed into its high-frequency detail layer and low-frequency base layer, in which the detail layer keeps the object's details as well as the rain streaks and the base layer

maintains the whole structure of the image. The detail layer is used to input the deep model for de-raining.

(2) We introduced a CNN model that can detect and remove rain streaks simultaneously. The CNN architecture with the convolution layers and deconvolution layers, in which the convolution layers serve as a feature extractor encoding the image contents while removing the rain streaks and the deconvolution layers recover the details of raw image. Moreover, for the purpose of helping the deconvolution layers recover more detail information, three skip connections between convolution and deconvolution operations are introduced.

(3) By adding the output of the CNN model to the low-frequency base layer, we obtained the final de-rained image.

To sum up, the contributions of this work are:

(1) We introduced a CNN architecture to address the single image de-raining issue. The designed CNN architecture contains a stack of convolution layers and deconvolution layers. Between the convolution and deconvolution layers, three skip connections are added. Compared with previous CNN-based methods, the proposed network is more depth, and we use smaller filter kernel sizes, which is conducive for digging deeper features and eliminating rain streaks. Additionally, with fewer feature maps and smaller filter, our CNN model has the advantages of fewer parameters and less computing effort. Thus, the training convergence and testing speed are faster than the competitive methods.

(2) Experiments on the publicly available synthesized dataset and real-world rainy images demonstrate the effectiveness of the proposed model. Our CNN architecture outperforms other recent state-of-the-art methods.

The remainder of this work is organized as follows. Section 2 gives a brief review of existing methods for image de-raining. Section 3 presents the detailed discussion on the proposed single image de-raining method. The results on both synthetic images and real images are provided in Section 4. Section 5 concludes this work.

2 Related works

In recent years, a large number of methods have been proposed to address the image de-raining problem. In this section, we give a review of the current methods used for image de-raining from the video sequences-based and single image-based perspectives.

Video sequences-based methods. Image de-raining from the video sequences has been extensively investigated. In Garg et al. [Garg and Nayar (2004, 2007)], the authors presented a comprehensive investigation of the visual effects caused by rain on imaging system and developed a rain detection and removal algorithm from video sequences. The rain streaks detection has two constraints [Li, Tan, Guo et al. (2016)]: (1) firstly, due to dynamic rain, the intensity changes of them inside some frames are relatively high. (2) Second, since other objects in videos may be dynamic, through verifying whether the relationships of the intensity changes along the rain streaks and the background intensity are linear, the rain streaks could be distinguished. The false alarms caused by the first constraint will be reduced by second constraint. After the rain streaks are detected, by taking the average intensity of the pixels taken from the subsequent and previous frames,

the model removes the rain streaks. Zhang et al. [Zhang, Li, Qi et al. (2006)] introduced a rain removal model that integrates both chromatic and temporal properties of rain streaks for video sequence. Through the chromatic property, they found that the changes of the R, G and B channels are approximately the same for the pixels representing streaks. The algorithm uses the two properties to detect and remove the streaks captured under the dynamic or stationary scenes. Bossu et al. [Bossu, Hautière and Tarel (2011)] proposed a selection rule based on the photometry and size information properties to select the rain streaks. They used the Gaussian mixture model to separate rain later from the raw image sequences.

Single image-based methods. Since there has no any temporal information can be exploited, compared with the video sequences-based rain removal issue, the single image-based de-raining (SIBD) problem is more challenging. Some literature regard the SIBD issue as a problem of image decomposition. Kang et al. [Kang, Lin and Fu (2012)] adopted a bilateral filter to decompose the original image into the low-frequency and high-frequency parts. By using sparse coding and dictionary learning, they decomposed the high-frequency part into “non-rain part” and “rain part”, and then the rain part can be removed. Through exploring the orientation and shape of the elliptical kernel, in [Kim, Lee, Sim et al. (2013)], the authors found that the shapes of the rain streaks are elongated ellipses with the vertical orientations. By analyzing the aspect ratio of the ellipses kernel and the rotation angle, the rain streak regions are detected. Then, these rain streaks were removed by using the non-local means filtering. Based on a nonlinear generative model, Luo et al. [Luo, Xu and Ji (2015)] introduced a dictionary learning-based method for single image de-raining. They used discriminative sparse coding to separate rain layer and de-rained image layer. Based on the assumption that rain streaks are low-rank, Chen et al. [Chen and Hsu (2013)] proposed a generalized low-rank model for single image de-raining where the nice properties of low-rank pattern are adopted. In Li et al. [Li, Tan, Guo et al. (2016)], the authors presented a method that utilizes patch-based priors based on Gaussian mixture models for both rain layers and de-raining layers.

Recently, deep learning technology has achieved significant successes in many computer vision domains, including image de-blurring [Schuler, Hirsch, Harmeling et al. (2016)], image de-noising [Zhang, Zuo, Gu et al. (2017)], super-resolution [Huang, He, Sun et al. (2017)], style transfer [Gatys, Ecker and Bethge (2016); Yan, Zhang, Wang et al. (2016)], and in-painting [Xie, Xu and Chen (2012)] etc. There also have some works employ the deep learning technology to address the single image de-raining problem. These deep learning-based methods attempt to learn a non-linear mapping between the rainy image and its corresponding ground truth. For example, Zhang et al. [Zhang, Sindagi and Patel (2017)] introduced a conditional generative adversarial model for SIBD. In order to have a better visual appeal of the final results, they defined a new loss function that combines Euclidean loss, adversarial loss and perceptual loss. Fu et al. [Fu, Huang, Ding et al. (2017)] proposed a convolution neural network named DrainNet for single image de-raining, similar to some image decomposition methods, they first decompose the original rainy image into base and detail layers. Then they used the detail layer to train the designed CNN architecture. Yang et al. [Yang, Tan, Feng et al. (2016)] designed a region-dependent rain image model for additional rain streaks detecting. Based on this model, they proposed a fully CNN model to detect and remove the rain simultaneously.

3 Proposed method

In this part, we have presented the detailed method proposed by this work. First, we have shown the proposed overall CNN architecture. Then, we have introduced the image decomposition approach. Finally, we described the training scheme of the proposed deep model.

3.1 Network architecture

The architecture of the proposed “Rain-removal Net (R²N)” is presented in Fig. 1. During the training phase, the original rainy image is first decomposed into detail and base layers and we use the detail layer as the input of the architecture. Then the detail layer passes through a set of convolution and deconvolution operations, in which the sizes of kernels of these convolution and deconvolution operations are comparatively small (the kernels of first convolution and last deconvolution layers with the size of 9×9 , 1×1 , respectively, and the kernels of rest of layers with the size of 3×3). Both the strides of convolution and deconvolution operations of the architecture are set to 1 pixel. After a stack of convolution and deconvolution layers, the output layer is following. Furthermore, we added three skip connections between convolution and deconvolution layers to enhance the recovery ability of the deconvolution operations. The detailed parameter settings of CNN architecture are presented in Tab. 1. Next, we presented the roles of the convolution and deconvolution layers in the designed deep architecture.

Table 1: Layer parameters of the Rain-removal Net (R²N)

| Layer Name | Parameters |
|---------------------|--|
| Image processing | decomposed Base layer and detail layer |
| Input | Detail layer |
| Layer 1 | Conv. (9,9,128), stride=1; Tanh |
| Layer 2 | Conv. (3,3,128), stride=1; Tanh |
| Layer 3 | Conv. (3,3,128), stride=1; Tanh |
| Layer 4 | Conv. (3,3,128), stride=1; Tanh |
| Layer 5 | Deconv. (3,3,128), stride=1; Tanh |
| Concatenate layer 1 | Concatenate (layer 5, layer 3) |
| Layer 6 | Deconv. (3,3,128), stride=1; Tanh |
| Concatenate layer 2 | Concatenate (layer 6, layer 2) |
| Layer 7 | Deconv. (3,3,128), stride=1; Tanh |
| Concatenate layer 3 | Concatenate (layer 7, layer 1) |
| Layer 8 | Deconv. (1,1,3), stride=1; Null; |
| Add layer | Add (Layer 8, Base layer) |
| Output | Rain-removal image |

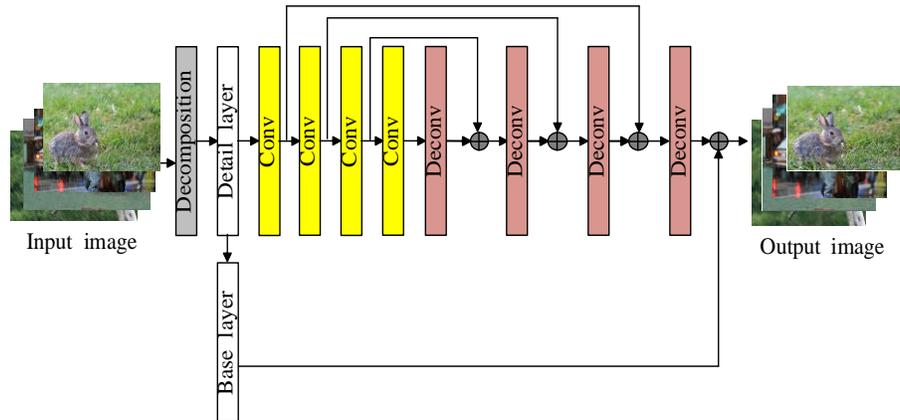


Figure 1: The proposed "R²N" architecture

3.1.1 Convolution layers

The proposed "R²N" network includes four convolution layers. In our network, the convolution layers are used as feature extractors, which maintains the primary components of the original images and removes the rain streaks simultaneously. We have set the kernel size of the first convolution layer to 9×9 , for the larger kernel size can capture more spatial information of the raw image. Both the kernel sizes of the rest of three convolution layers are set to 3×3 . Through using small kernel size, we can increase the depth of the network and embed more non-linear activation functions, which can enhance the discriminative ability of the leaned deep feature. Both the numbers of the feature map of these convolution layers are set to 128. Note that the designed architecture does not consist of un-pooling or pooling operations, for the un-pooling and pooling operations may discard some important information of the original image that is useful for the recovering phase.

3.1.2 Deconvolution layers

In the proposed architecture, the original detail layer is successively concentrated into a smaller size abstraction by the convolution operations, during which the critical details of the raw detail layer are discarded. In other words, the convolution operations reduce the resolution of the detail layer. In order to restore the details of the original detail layer, we introduced the deconvolution operations. After the four convolution layers, four deconvolution layers are added for recovering the details of the raw image. Similar to the convolution layers, both the kernel sizes of the first three deconvolution layers are set to 3×3 . For the purpose of improving the expressive ability of the proposed model, we set the kernel size of the last deconvolution layer to 1×1 . Moreover, the numbers of the feature maps of each deconvolution layer are set to 128, 128, 128 and 3, respectively. After the four deconvolution layers, the network outputs the clean version of the detail layer.

3.1.3 Skip connections

As shown in Fig. 1, the proposed network includes several convolution operations. These operations may seriously damage the details of the original detail layer, which makes it difficult for deconvolution layers recovering them. However, the feature maps produced by the convolution layers include many important details, adding these feature maps to the outputs of deconvolution layers can improve the restoring ability of deconvolution operations. Therefore, we introduced three skip connections between convolution layers and deconvolution layers. Moreover, analogous to the deep residual networks proposed by [He, Zhang, Ren et al. (2016)], the skip connections are helpful for back-propagating the gradients to the bottom layers, which makes the training process more steady.

3.2 Training on detail layers

Similar to the work of [Fu, Huang, Ding et al. (2017)], through using the guided filtering method proposed by [He, Sun and Tang (2013)], we first decomposed the original rainy image into its detail layer and base layer, in which the detail layer maintains high-frequency components like rain streaks and other edge information and the base layer retains the low-frequency basic information. Given a rainy image I , the relationship between it, its detail and base layers can be written as:

$$I = I_{\text{detail}} + I_{\text{base}} \tag{2}$$

in which I_{detail} and I_{base} represent the detail layer and base layer, respectively.

Then the detail layer is used to input the designed CNN architecture. Training on the detail layer has the following advantages: (1) since most regions of the detail layer are near to zero, compare with the original image, it is sparser. Thus, directly using the detail layer to train the architecture is helpful for reducing the computational resources. (2) Training on the detail layer can improve the convergent performance of the deep learning system [Fu, Huang, Ding et al. (2017)], which can reduce the time consumption of the training phase.

After the model is trained, the de-rained detail layer can be produced by the output layer. Then we add the output to the base layer to generate the final de-rained result. This process can be formulated as:

$$E = f_W(I_{\text{detail}}) + I_{\text{base}} \tag{3}$$

where f_W is the output of the designed deep architecture and E represents the final de-rained image.

3.3 Objective function

We denote the corresponding clean image of the rainy image I as G . Then the network architecture is trained to achieve the goal that minimizes the initial loss function as:

$$L = \frac{1}{N} \sum_{n=1}^N \left\| f_W(I^n) - G^n \right\|_F^2 \tag{4}$$

where n represents the indices of the images, N is the mini-batch size, F denotes the Frobenius norm and W are the parameters need to be learned. Since our architecture is trained on detail layer $I_{\text{det ail}}$, then the loss function in (4) can be rewritten as:

$$L = \frac{1}{N} \sum_{n=1}^N \left\| f_W(I_{\text{det ail}}^n) - G_{\text{det ail}}^n \right\|_F^2 + \lambda \|W\|^2 \quad (5)$$

in which $G_{\text{det ail}}$ represents the detail layer of image G . In order to avoid overfitting, the L₂ penalty regularization is introduced to add the loss function. λ is the balance coefficient.

3.4 Training strategies and dataset

We trained the proposed architecture on the synthesized dataset created by Fu et al. [Fu, Huang, Ding et al. (2017)]. The dataset consists of 350 clean open-air images selected from Google image search, UCID dataset [Arbelaez, Maire, Fowlkes et al. (2011)] and BSD dataset [Schaefer and Stich (2003)]. Each clean image is employed to synthesize 14 rainy images with different orientations and intensity of rain streaks. Two hundred thousand 64×64 rainy/clean patch pairs are randomly selected as training data. These training data are separated into mini-batches. Furthermore, we used stochastic gradient descent (SGD) to back-propagating the gradients for the network parameters and adopt tanh function as the inspirit function of the architecture. For each iteration (s), the parameters of the architecture are updated as followings:

$$W_{s+1} = W_s - \alpha (f_W(I_{\text{det ail}}^n) - G_{\text{det ail}}^n)^T \frac{\partial f_W(I_{\text{det ail}}^n)}{\partial W} \quad (6)$$

$$b_{s+1} = b_s - \alpha (f_W(I_{\text{det ail}}^n) - G_{\text{det ail}}^n)^T \quad (7)$$

where T is a transpose operator and α represents the learning rate.

4 Experiments

The Tensorflow framework was used to perform the experiments. We spend roughly 2-3 hours on the convergence of training phase with two NVIDIA GTX Taitan-xp GPUs. The learning rate and batch size are set to 0.002 and 10, respectively. We set the training epoch to 6. Compared with the former deep learning-based models, our architecture needs fewer epochs for converging, and each epoch spends less time. We supposed that this is due to the following facts: (1) our network with fewer feature maps and smaller kernel sizes, which is helpful to reduce the parameters and calculation time. (2) We used fewer rainy/clean patch pairs for training, which further reduces the computing effort.

4.1 Test data and evaluation protocol

(1) Data: Both synthesized and real-world rainy images were used to evaluate the proposed architecture. During the test process, the newly rainy images from the 350 clean images are synthesized. As regards to the real-world rainy images, we selected the images with the different orientations and magnitudes of the rainy pixel from the Google

image search.

(2) Evaluation Protocol: We used the structural similarity index (SSIM) [Wang, Bovik, Sheikh et al. (2004)] to estimate the performances of the proposed CNN architecture as well as the comparative models. The higher SSIM values, the closer to corresponding ground truths the de-rained images are. Note that the SSIM value of the ground truth is 1.

4.2. Comparison with state-of-the-art

We used the following representative single image de-raining models for comparisons:

- (1) DOR: a specialized convolution neural network-based approach [Eigen, Krishnan and Fergus (2013)]
- (2) DSC: the dictionary learning based/discriminative sparse coding-based model [Luo, Xu and Ji (2015)]
- (3) GMM-LP: simple patch-based priors model [Li, Tan, Guo et al. (2016)]
- (4) DrainNet: convolution neural network-based method [Fu, Huang, Ding et al. (2017)]

Results on synthesized rainy images: In this set of experiments, we compared the proposed architecture with the four representative de-raining methods on the newly synthesized rainy images. Since the ground truths of these synthesized rainy images are known, the structure similarity index (SSIM) between the de-rained image and its corresponding ground truth can be calculated for quantitative measuring. The SSIM experiment results on four newly synthesized rainy images are presented in Tab. 2. From Tab. 2, we can see that the proposed architecture achieves the highest SSIM values on both four synthesized rainy images, which verifies the effectiveness of the proposed architecture.

Table 2: SSIM results on synthesized test image

| Image | Ground truth | Rainy image | DOR | DSC | GMM-LP | DrainNet | Ours |
|--------------|--------------|-------------|------|------|--------|----------|-------------|
| Dock | 1 | 0.86 | 0.88 | 0.88 | 0.90 | 0.92 | 0.93 |
| Umbrella | 1 | 0.75 | 0.83 | 0.83 | 0.86 | 0.88 | 0.90 |
| Rabbit | 1 | 0.72 | 0.81 | 0.77 | 0.79 | 0.85 | 0.86 |
| Bird | 1 | 0.57 | 0.79 | 0.62 | 0.75 | 0.82 | 0.85 |

We also presented some visual comparisons for these synthesized images above. The visual results are shown as Fig. 2. From Fig. 2, we can observe that both DOR and DSC can reduce the denseness of the rain streaks as well as eliminate the partial rain streaks, but they cannot eliminate the rain streaks completely. GMM-LP could completely remove the rain streaks. However, their de-rained results contain over-smoothed regions and cannot retain the details of the ground truth. Compared with other single image de-rain approaches, our architecture can eliminate the most rain steaks and reserve the details of the original clean image simultaneously. Note that the visual effects of the de-rained images generated by DrainNet can be almost comparable to ours, but both the

SSIM values of our model on the four de-rained images are higher than theirs, which demonstrates the effectiveness of the proposed method.



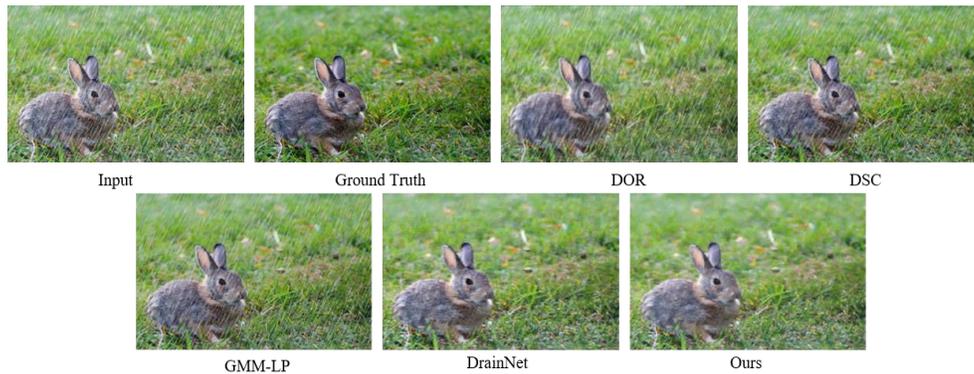


Figure 2: Results comparison on four synthesized rainy images

Results on Real-world data: We also used the real-world rainy images to evaluate the proposed method. The testing results on three real-world rainy images are presented in Fig. 3. From (a) in Fig. 3, we can see that both the de-rained results of DOR, GMM-LP and DSC can only remove partial rain streaks, and the de-rained results of them obviously contain many rain streaks. Even though the de-rained results of DrainNet are able to achieve good performance, compared with our method, it cannot retain the details of the de-rained images. For the purpose of having better visual comparisons, we present one specific regions-of-interest for both DrainNet and our model. From the regions-of-interest, we can observe that our model can maintain more features and details related to the original images. In (b) and (c), we present one specific regions-of-interest for all methods. Through observing these regions-of-interests, we can see that our model achieves the best visual performances and is able to remove rain streaks and retain details simultaneously, which further verifies the validity of the proposed model.

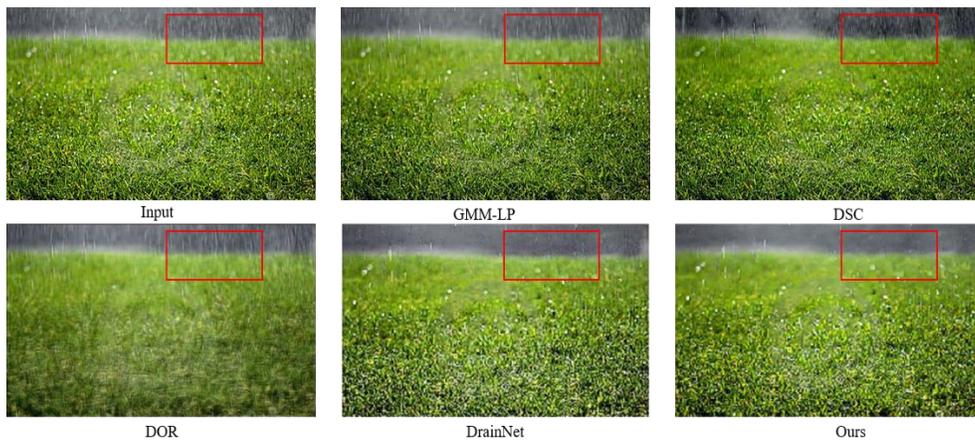


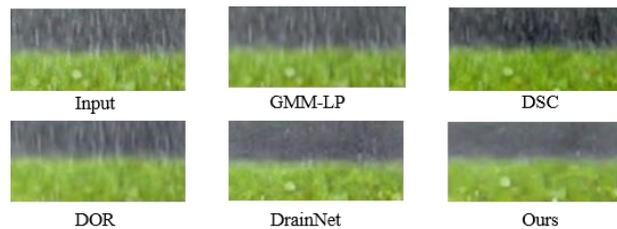


(a)



(b)





(c)

Figure 3: Results on real-world rainy images

5 Conclusions

In this work, we have presented an end-to-end convolution neural network-based architecture for removing the rain from single rainy images. The proposed architecture contains a stack of convolution and deconvolution layers that can remove rain streaks and maintain the details of the background of the original image simultaneously. For the purpose of boosting the recovering ability of the deconvolution operations as well as improving the steady of training phase, we introduced three skip connections between the de-convolution and convolution operations. Moreover, compared with other deep de-raining methods, the proposed deep architecture with deeper depth and smaller kernel sizes. Thus, the architecture can extract higher-level deep descriptors, which is helpful for improving the de-raining ability of the deep architecture. Experiments on real-world and synthetic rainy images demonstrate that the performances of the proposed architecture outperform other state-of-the-art models.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (Grant No. 61673222) and Jiangsu Universities Natural Science Research Project (Grant No. 13KJA510001) and Major Program of the National Social Science Fund of China (Grant No. 17ZDA092).

References

- Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J.** (2011): Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898-916.
- Bossu, J.; Hautière, N.; Tarel, J. P.** (2011): Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International Journal of Computer Vision*, vol. 93, no. 3, pp. 348-367.
- Brewer, N.; Liu, N. J.** (2008): Using the shape characteristics of rain to identify and remove rain from video. *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, vol. 5342, pp. 451-458.

- Chen, Y. L.; Hsu, C. T.** (2013): A generalized low-rank appearance model for spatio-temporally correlated rain streaks. *IEEE International Conference on Computer Vision*, pp. 1968-1975.
- Eigen, D.; Krishnan, D.; Fergus, R.** (2013): Restoring an image taken through a window covered with dirt or rain. *IEEE International Conference on Computer Vision*, pp. 633-640.
- Fu, X. Y.; Huang, J. B.; Ding, X. H.; Liao, Y. H.; Paisley, J.** (2017): Clearing the skies: a deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944-2956.
- Garg, K.; Nayar, S. K.** (2004): Detection and removal of rain from videos. *IEEE Computer Society Conference on Computer*, vol. 1, pp. 528-535.
- Garg, K.; Nayar, S. K.** (2007): Vision and rain. *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3-27.
- Gatys, L. A.; Ecker, A. S.; Bethge, M.** (2016): Image style transfer using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414-2423.
- He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J.** (2016): Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778.
- He, K. M.; Sun, J.; Tang, X. O.** (2013): Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397-1409.
- Huang, D. A.; Kang, L. W.; Wang, Y. C. F.; Lin, C. W.** (2014): Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 83-93.
- Huang, D. A.; Kang, L. W.; Yang, M. C.; Lin, C. W.; Wang, Y. C. F.** (2012): Context-aware single image rain removal. *IEEE International Conference on Multimedia and Expo*, pp.164-169.
- Huang, H. B.; He, R.; Sun, Z. N.; Tan, T. N.** (2017): Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1698-1706.
- Kang, L. W.; Lin, C. W.; Fu, Y. H.** (2012): Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742-1755.
- Kim, J. H.; Lee, C.; Sim, J. Y.; Kim, C. S.** (2013): Single-image deraining using an adaptive nonlocal means filter. *IEEE International Conference on Image Processing*, pp. 914-917.
- Li, Y.; Tan, R. T.; Guo, X. J.; Lu, J. B.; Brown, M. S.** (2016): Rain streak removal using layer priors. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2736-2744.
- Luo, Y.; Xu, Y.; Ji, H.** (2015): Removing rain from a single image via discriminative sparse coding. *IEEE International Conference on Computer Vision*, pp. 3397-3405.

Schaefer, G.; Stich, M. (2003): Ucid: An uncompressed color image database. *Storage and Retrieval Methods and Applications for Multimedia*, pp. 472-480.

Schuler, C. J.; Hirsch, M.; Harmeling, S.; Schölkopf, B. (2016): Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1439-1451.

Sun, S. H.; Fan, S. P.; Wang, Y. C. F. (2014): Exploiting image structural similarity for single image rain removal. *IEEE International Conference on Image Processing*, pp. 4482-4486.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. (2004): Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612.

Xie, J. Y.; Xu, L. L.; Chen, E. H. (2012): Image denoising and inpainting with deep neural networks. *International Conference on Neural Information Processing Systems*, vol. 1, pp. 341-349.

Yan, Z. C.; Zhang, H.; Wang, B. Y.; Paris, S.; Yu, Y. Z. (2016): Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics*, vol. 35, no. 2, pp. 1-15.

Yang, W. H.; Tan, R. T.; Feng, J. S.; Liu, J. Y.; Guo, Z. M. et al. (2016): Joint rain detection and removal via iterative region dependent multi-task learning. *CoRR*, abs/1609.07769, 2, 3.

Zhang, H.; Sindagi, V.; Patel, V. M. (2017): Image de-raining using a conditional generative adversarial network. arXiv preprint arXiv:1701.05957.

Zhang, K.; Zuo, W. M.; Gu, S. H.; Zhang, L. (2017): Learning deep cnn denoiser prior for image restoration. *IEEE Conference on Computer Vision and Pattern*, pp. 2808-2817.

Zhang, X. P.; Li, H.; Qi, Y. Y.; Leow, W. K.; Ng, T. K. (2006): Rain removal in video by combining temporal and chromatic properties. *IEEE International Conference on Multimedia and Expo*, pp. 461-464.