Residual Multiscale Based Single Image Deraining

Yupei Zheng[†1]
ypzhengx@bjtu.edu.cn
Xin Yu[†2]
xin.yu@anu.edu.au
Miaomiao Liu[2]
miaomiao.liu@anu.edu.au
Shunli Zhang[*1]
slzhang@bjtu.edu.cn

[1] School of Software Engineering
Beijing Jiaotong University
Beijing, China

[2] Australian National University
Canberra, Australia

Abstract

Rain streaks deteriorate the performance of many computer vision algorithms. Previous methods represent rain streaks by different rain layers and then separate those layers from the background image. However, it is rather difficult to decouple a rain image into rain and background layers due to the complexity of real-world rain, such as various shapes, directions, and densities of rain streaks. In this paper, we propose a residual multiscale pyramid based single image deraining method to alleviate the difficulty of rain image decomposition. In particular, we remove rain streaks in a coarse-to-fine manner. In this fashion, the heavy rain can be significantly removed in the coarse-resolution level of the pyramid first, and the light rain will then be further removed in the high-resolution level. This allows us to avoid distinguishing the densities of rain streaks explicitly since the inaccurate classification of rain densities may lead to over- or insufficient-removal of rain. Furthermore, the residual between a recovered image and its corresponding rain image can provide vital clues of rain streaks. We therefore exploit such residual as an attention map for deraining in its consecutive finer-level. Benefiting from the residual attention maps, rain layers can be better extracted from a higher-resolution input image. Extensive experimental results on synthetic and real datasets demonstrate that our method outperforms the state of the art significantly.

1 Introduction

As a common weather phenomenon in daily life, rain streaks often reduce the quality of images (e.g. blurry and heavy occlusion) captured in such weather conditions. It further seriously deteriorates the performance of widely used algorithms on object recognition [28], object tracking [5] and detection [4] for outdoor surveillance systems. Therefore, removing rain streaks from a single image is critical for a wide range of outdoor vision-based applications.

[†] Joint first authors.
[*] Shunli Zhang is the corresponding author.

Traditional single-image based deraining methods [2, 14, 18, 20, 31] remove rain by learning discriminative representations of rain streaks and background details. However, these methods may fail to deal with real-world rain owing to the complexity of real-world rain such as the overlapping between rain streaks and complex background texture.

Recently, due to the powerful feature representations of convolutional neural networks (CNN), deep learning based deraining methods [6, 7, 17, 25, 27, 29, 30] have been proposed to remove rain streaks in a single image and achieved promising performance. These methods usually assume rain can be decomposed into several rain layers, each of which contains one type of rain streaks with similar features, such as directions, shapes, and scales. They make an effort to learn representations of different rain layers and then subtract these layers from the rain image. However, these methods have the following limitations:

- In real-world cases, it is challenging to determine the accurate number of the rain layers accounting for different types of rain streaks, (*e.g.*, different shapes, directions, and densities). Inaccurate estimation of the number of layers will result in removing rain streaks incompletely or excessively.
- Rain streaks with different densities may appear in the same image and different rain layers always overlap with each other. The work [29] suggests different models should be used to handle different densities of rain (*i.e.*, heavy, medium, and light rain streaks). However, it is difficult to establish a model that recognizes rain density accurately. Thus, previous methods may suffer from either inadequately removing rain or overly smoothing background images.

To address the above issues, we propose a multiscale pyramid based deraining method to remove rain streaks while recovering background details in a coarse-to-fine manner. In this way, heavy rain streaks are firstly removed in the coarse level of the image pyramid. As the image resolution increases, we gradually recover background details while removing lighter rain from a rain image. In other words, we first recover the structure of the background image in the coarse level and then reconstruct fine details of the background image in high-resolutions. Therefore, we do not explicitly estimate the density of rain, and thus avoid inadequately or overly removing rain. Moreover, since the residual between the recovered image and its corresponding rain image in the higher resolution can represent the missing background details as well as rain regions, we exploit the residual as an input attention map. In this manner, our network can distinguish which regions are contaminated by rain as well as the densities of the contamination, and then it can remove rain streaks in accordance with the weights of the attention map (*i.e.* the residual). Therefore, our network significantly alleviates the effects of overly or inadequately removing rain.

Overall, the main contributions of this paper are summarized as follows:

- We propose a residual multiscale based deraining framework to remove rain streaks and recover background details progressively. To the best of our knowledge, our method is the first attempt to address the deraining task in a coarse-to-fine fashion.
- We fully exploit the residual between a restored image and its corresponding rain image as an attention map to provide the density and intensity information of rain, thus facilitating identifying rain streaks and recovering background details.
- By exploiting the residual based pyramid framework, our method bypasses the procedure of identifying the density of rain streaks and thus avoids the artifacts of incompletely or excessively removing rain caused by inaccurate density estimation of rain.
- The experimental results demonstrate that our proposed method outperforms state-of-the-art methods by a large margin of 1.14dB on the generalization ability test set.

2 Related Work

Many methods have been developed to remove rain streaks in the past decades. According to different input sources, deraining methods can be categorized into two groups: video-based and single-image rain streak removal methods.

Video-based rain streak removal methods leverage the temporal and spatial information discrepancy across frames to detect and remove rain streaks in videos. The work [8] develops a correlation model that captures the dynamics of rain and a physics-based motion blur model that explains the photometry of rain. Then it employs those two models to detect and remove rain from videos. Later, several video-based deraining methods have been proposed and achieved appealing performance [1, 9, 10, 19, 22, 24]. However, they highly rely on the temporal consistency of contents in videos, which is not available in a single image.

For the single image based deraining task, Rubinstein *et al*. [21] apply dictionary-based sparse priors to describe rain images as a combination of atoms from a pre-specified dictionary. Kang and Fu [14] first decompose an input rain image into a high-frequency layer and a low-frequency layer, and then remove rain streaks in the high-frequency layer via sparse coding techniques. Chen *et al*. [3] use a low-rank structure to model rain streaks and then exploit the spatial-temporal correlations to remove rain. Luo *et al*. [20] utilize a sparse coding based method to separate rain layers from background. Li *et al*. [18] propose a Gaussian mixture model (GMM) to model rain streaks and the background separately and remove rain from the background image.

Very recently, due to the strong feature extraction ability of deep convolutional neural networks (CNNs), Fu *et al*. [6, 7] first exploit a CNN to extract rain layers. They decompose an image into a low-frequency layer and a high-frequency detail layer. The detail layer is taken as the input of a CNN for rain removal. Li *et al*. [16] train parallel sub-networks to handle different scales of rain streaks in a recurrent manner. Yang *et al*. [27] develop a recurrent contextualized dilated network to jointly detect and remove rain. Li *et al*. [17] remove rain streaks by utilizing squeeze-and-excitation context aggregation to learn different weights for different rain layers. Zhang *et al*. [29] first classify an input rain image into different rain levels, such as light, medium and heavy levels, and then remove rain according to the classification results. Wang *et al*. [25] employ two-round four-directional IRNN architecture [15] to distinguish rain and contextual features. They also design a direction-aware attention mechanism to remove rain streaks from rain images.

Although previous deep learning based methods achieve promising progress in many cases, they still have some limitations. Since those methods aim to represent rain by learning different rain layers and then separate those rain layers from the background image in the original image resolution, rain layers may not represent rain streaks accurately due to the intrinsic overlapping between different rain layers. Furthermore, multiple densities of rain streaks may exist in an image. Thus, using a single model at a single image scale may not remove different densities of rain effectively. Therefore, the performance of those methods tends to saturate.

3 Proposed Method

Our proposed deraining network is based on a multiscale pyramid framework. The processing of each scale of the image is referred to as a *stage*. We aim to remove different densities of rain streaks at different stages (see Figure 1). In this manner, heavy rain can be relatively easy to be removed in low-resolution images compared to their original resolution versions. Therefore, our method eases the procedure of removing rain with multiple densities in an

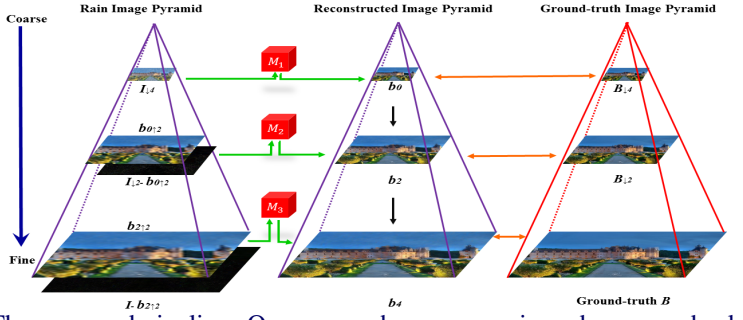


Figure 1: The proposed pipeline. Our approach removes rain and recovers background via three stages. I is an input rain image. M_n denotes the nth stage of image restoration. $\downarrow n$ and $\uparrow n$ represent downsampling and upsampling an image by a factor of $n\times$.

image. Moreover, the residual between our restored rain-free image and its corresponding rain image is also fully explored in our network. Specifically, it provides information about the regions which are contaminated by rain. It thus lets our network focus on those regions. Details about the rain image formation and our proposed multiscale pyramid framework are presented as follows.

3.1 Rain Image Formulation

A rain image I is often defined as [27]:

$$I = B + \sum_{i=1}^{N} R_i, \tag{1}$$

where B is a clean background image, R_i means the i-th rain layer with similar characteristics, and N denotes the total number of rain layers. However, rain streaks often overlap with each other and mix with the background in real-world cases, so that it is difficult to separate them into individual layers. Moreover, inaccurate estimation of the number of rain layers often leads to inadequately or overly rain removal. Instead of estimating the rain layers, we introduce the following residual multiscale pyramid framework which progressively removes the rain streaks with different densities at different stages.

3.2 Residual Multiscale Pyramid Framework

We establish a 3-level image pyramid[‡] to remove rain streaks gradually (*i.e.*, from heavy to light densities), and restore background images from structure to fine details.

3.2.1 Stage 1: Heavy rain removal & Image structure extraction

Since a downsampling operation with an aggressive scaling factor can remove severe rain and retain background structure, stage 1 is designed to remove heavy rain and recover the structure of the background in the coarse level.

As shown in Figure 2, the input of stage 1 is a rain image which is downsampled by a factor of $4\times$, denoted as $I_{\downarrow 4}$. The output of stage 1 is a restored image in the coarse level, denoted as b_0. The model of stage 1, denoted as M_1, is constructed by an autoencoder which consists of Conv(Deconv)-Relu-BN layers. We also employ skip connections to prevent from information loss and facilitate gradient backpropagation in the autoencoder.

Since stage 1 mainly removes heavy rain streaks while restoring image structure, image details can hardly be recovered at this stage. Thus, we only enforce the intensity similarity between the estimated image and its ground-truth counterpart by an Euclidean loss:

[‡]The building of an image pyramid depends on the image resolution. We adopt three levels for the experiments in our paper.

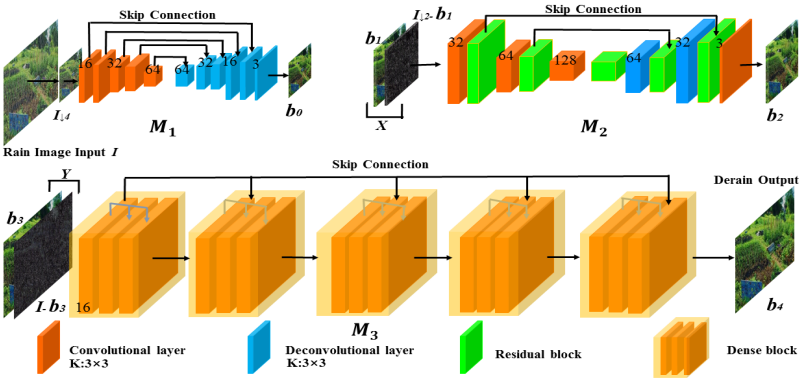


Figure 2: The structure of our three stage models in the pyramid. K denotes the convolutional kernel size.

$$L_{e1} = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\|b_0^{c,w,h} - B_{\downarrow 4}^{c,w,h}\|_2^2 = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\|M_1(I_{\downarrow 4}^{c,w,h}) - B_{\downarrow 4}^{c,w,h}\|_2^2, \quad (2)$$

where C, W and H represent the number of the channels, width and height of the image respectively.

3.2.2 Stage 2: Medium rain removal & Image details restoration

After obtaining the result b_0 from stage 1, we upsample b_0 by a factor of $2\times$, denoted as $b_1(b_1 = b_{0\uparrow 2})$. Then the residual between b_1 and the corresponding rain image $I_{\downarrow 2}$ is marked as r_1. Note that, the residual reflects the regions corrupted by rain streaks and the missing background details. Thus we employ the residual r_1 as an attention map to further remove medium rain and recover background details in the second stage.

We observed that the rain image $I_{\downarrow 2}$ in stage 2 mostly contains more details of background and medium rain streaks. The network architecture of the second stage model is also constructed by an autoencoder, denoted as M_2 (see Figure 2). Since we aim to recover more background details in this stage, we replace convolutional layers with residual blocks to increase the feature extraction ability. The recovered image from stage 2 is denoted as b_2. In stage 2, we utilize not only a pixel-wise intensity similarity loss but also a feature-wise perceptual loss in order to restore more background details while removing medium rain. The pixel-wise intensity similarity loss is written as:

$$L_{e2} = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\|M_2(X^{c,w,h}) - B_{\downarrow 2}^{c,w,h}\|_2^2, \quad (3)$$

where X indicates the input of the second stage model. In our algorithm, we concatenate b_1 and r_1 as our input X. The perceptual loss is expressed as:

$$L_{p2} = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\|\Psi(M_2(X^{c,w,h})) - \Psi(B_{\downarrow 2}^{c,w,h})\|_2^2, \quad (4)$$

where Ψ represents the extracted feature maps by a pre-trained model [13, 23]. In our method, we use the feature maps from the layer Relu2_2 in VGG16.

3.2.3 Stage 3: Light rain removal & Image fine details recovering

Previous methods remove different densities of rain streaks only at the original resolution, and thus may fail to remove rain in all different densities and types. In contrast, with the help of our residual based multiscale pyramid framework, we only need to focus on removing

light rain at stage 3 since heavy and medium rain have been removed and identified at the former two stages. Therefore, in the last stage, we aim at enhancing fine image details as well as removing light rain streaks, which makes the rain streak removal task much easier. As shown in Figure 2, the model in stage 3 is represented as M_3. M_3 is constructed by an autoencoder structure with dense blocks [12]. Similar to stage 2, we upsample the result of stage 2, namely b_2, by a factor of $2\times$, denoted as $b_3 (b_3 = b_{2\uparrow 2})$. The residual between b_3 and I is r_2. We also concatenate r_2 and b_3 as the input, denoted as Y, for model M_3. In order to achieve fine details of a clean full-resolution background image b_4, we employ three loss functions to train M_3. They are (i) a pixel-wise intensity similarity loss L_{e3}:

$$L_{e3} = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} \|M_3(Y^{c,w,h}) - B^{c,w,h}\|_2^2, \tag{5}$$

(ii) a perceptual loss L_{p3}:

$$L_{p3} = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} \|\Psi(M_3(Y^{c,w,h})) - \Psi(B^{c,w,h})\|_2^2, \tag{6}$$

and (iii) a loss L_{ssim} (SSIM), similar to [26], to measure the structure similarity between the restored image with fine structure and the ground-truth images:

$$L_{ssim} = 1 - \frac{1}{CWH} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} SSIM(M_3(Y^{c,w,h}), B^{c,w,h}), \tag{7}$$

In Eqn. (7), SSIM between two image patches s and t is defined by:

$$SSIM = \frac{(2u_s u_t + c_1)(2\sigma_{st} + C_2)}{({u_s}^2 + {u_t}^2 + C_1)({\sigma_s}^2 + {\sigma_t}^2 + C_1)}, \tag{8}$$

where u_s and u_t represent the average intensity for patch s and t respectively, σ_s^2 and σ_t^2 indicate the variance of their corresponding patch, and σ_{st} is the covariance between s and t. The patch size is set as 11 in our method.

3.3 Implementation Details

In our experiments, the resolution of input rain images is 512×512 pixels. We construct our 3-level image pyramid and each level in the pyramid is a downsampled version of its previous level by a factor of $2\times$. Thus the image resolutions in our pyramid are 128×128, 256×256 and 512×512 pixels, and images in the higher level of the pyramid correspond to the coarse resolution. The residuals are calculated on the fly by subtracting the upsampled derained images from the rain-free counterparts. In stage 1, the autoencoder M_1 is constructed by five convolutional layers and five deconvolutional layers. The kernel size for all the convolutional and deconvolutional layers is 3. In stage 2, we replace the convolutional layers with residual blocks [11] in M_2 to extract features. In stage 3, we exploit dense blocks [12] to extract features in M_3. The number of dense blocks [12] is set as 5. The growth rate and kernel size of each dense layer in the dense block are set as 16 and 3, respectively. In order to train our models, we use the downsampled rain-free images as supervision signals, and we employ Adam algorithm for the training stage to update the parameters of our models. The learning rate is initially set to 0.002. We will release our training protocols.

4 Experiments

We train our network on a synthetic dataset and test it on both the synthetic and real-world rain images. Moreover, we also compare our approach with seven state-of-the-art methods, *i.e.*, discriminative sparse coding (DSC) [20], Gaussian mixture model (GMM) [18], CNN [6], DETAIL [7], RESCAN [17], DID-MDN [29] and SPANet [25]. For fair comparisons, we adopt the training codes or pre-trained models provided by these methods to generate the deraining results.

Table 1: Quantitative comparison on Test1 and Test2 test sets.

	Metric	[20]	[18]	[6]	[4]	[17]	[29]	[25]	Ours
Test1	PSNR(dB)	21.44	22.75	22.07	27.33	27.42	27.95	28.64	**29.41**
Test1	SSIM	0.79	0.84	0.84	0.90	0.89	0.91	0.91	**0.92**
Test2	PSNR(dB)	20.08	20.66	19.73	25.63	24.96	26.07	25.18	**27.21**
Test2	SSIM	0.78	0.81	0.83	0.89	0.87	0.91	0.87	**0.91**

4.1 Datasets and Evaluation Metric

We employ a synthetic dataset [29] to train our network as well as conduct qualitative and quantitative evaluations. The dataset [29] contains 12,000 pairs of images for training, denoted as Train12000, including 4,000 light, medium, heavy rain images respectively. [29] also provides 1,200 pairs of synthetic rain images as the test set, denoted as Test1. To test the generalization capability of our proposed method, we also test our model based on the 1,000 images from [4], denoted as Test2. Note that, we do not fine-tune our network when examing the generalization ability on the test set [4].

Moreover, we also test our network on a real-world rain dataset [30] which contains 52 images downloaded from the Internet, denoted as Real-Rain52. The collected images are diverse in terms of image contents as well as intensities and directions of rain streaks. Since there is no corresponding ground-truths for those rain images, we only compare the derained images with other methods qualitatively.

In our experiments, we employ Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [26] to evaluate the performance quantitatively on the synthetic dataset Test1 and Test2. Due to the space limit, more visual results are provided in the supplementary material.

4.2 Results on Synthetic Dataset

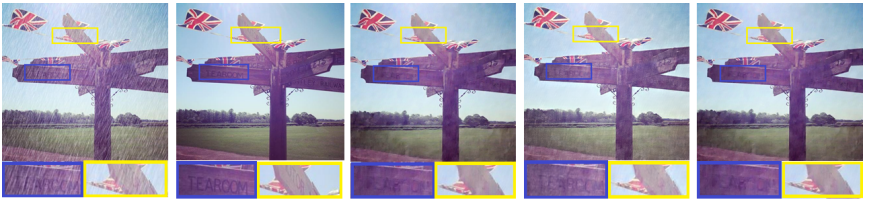
We first evaluate our network on the synthetic dataset [30] (*i.e.*, Test1) and compare our deraining results with the results of the state-of-the-art. Here, we use the same training and test split protocols as [29], and also test all the other methods on the same test set. As indicated in Table 1, our method achieves superior deraining performance to the state-of-the-art methods in terms of the average PSNR and SSIM. Note that, our approach outperforms the second best method by a margin of 0.77 dB.

Furthermore, we also test our trained network on Test2 to evaluate the generalization ability of our network. Table 1 demonstrates that our method obtains 1.14 dB PSNR improvement over the second best. This implies that our method is not restricted by the rain in the training set. With the help of the pyramid framework, our approach can address different densities of rain, thus obtaining appealing generalization ability.

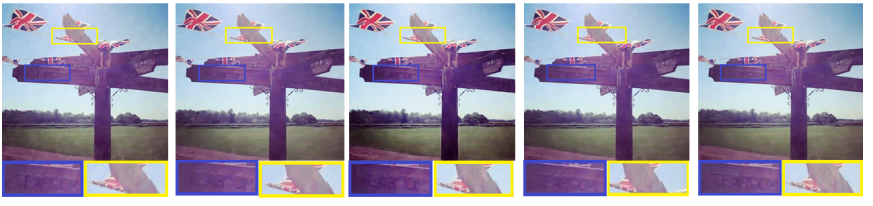
We also provide a qualitative comparison with other methods in Figure 3. As visable in the results, our approach can recover much clearer images with more authentic details than other methods. For instance, the characters are visible in our derained result. More qualitative comparsions are shown in the supplementary material.

4.3 Results on Real-World Dataset

Since ground-truth rain-free images are not available for real-world rain images, we only visually evaluate the deraining performance on Real-Rain52. As seen in Figure 4, rain streaks and marks still remain in the results of other methods. This phenomenon implies that removing different sizes and densities of rain by a single model may not be sufficient. In contrast, our method can recover clearer background images and achieves better visual quality, thus

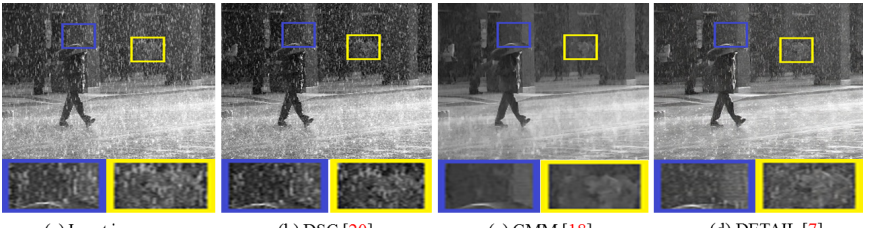


(a) Input image (b) Ground-truth (c) DSC [20] (d) GMM [18] (e) CNN [6]

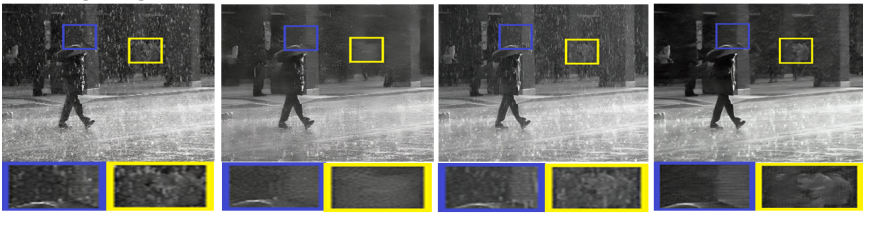


(f) DETAIL [7] (g) RESCAN [17] (h) DID-MDN [29] (i) SPANet [25] (j) Ours

Figure 3: Comparisons with the state-of-the-art on the Test1. Note that, the words "TEA-ROOM" and "TOP" are blurred by other methods but are clearer in our derained result. Best viewed on the screen.



(a) Input image (b) DSC [20] (c) GMM [18] (d) DETAIL [7]



(e) RESCAN [17] (f) DID-MDN [29] (g) SPANet [25] (h) Ours

Figure 4: Comparisons with the state-of-the-art on the Real-Rain52. In the input rain image, several pedestrians are 'occluded' by the rain streaks. Our method can better remove rain streaks than the other approaches, where persons in the distance are visible in our result. Best viewed on the screen.

demonstrating that our method is able to remove rain in different densities and different shapes more effectively.

4.4 Ablation Study

Multiscale pyramid network: In order to investigate the effects of the coarse-to-fine deraining pipeline, we conduct the following experiments in Table 2: (A1) removing rain streaks in the image without using the coarse-to-fine pipeline, (B1) and (C1) are designed to demonstrate how different image levels in the pyramid contribute to the final performance. Table 2 indicates that by using multiscale pyramid framework we achieve better deraining performance in comparison to removing rain only in the original image resolution. As shown in

Figure 5: Illustration of result with and without multiscale strategy. PSNR/SSIM results are included for reference. Best viewed on the screen.

Figure 6: Illustration of our derained result in different stages. In (a), (b) and (c), the left images are rain images and the right ones are our derained results. Best viewed on the screen.

Figure 5, compared to deraining on only one scale, multiscale strategy helps for removing heavy rain. Figure 6 illustrates intermediate derained results at different stages. We recover image details gradually as the resolution increases.

Residual attention mechanism: We evaluate the impacts of the residual attention for image deraining, denoted as (A0) in Table 2. In (A0), we only feed the upsampled derained image to the next level in the pyramid for recovering background details. As seen in Table 2, without using the residual attention mechanism, the performance of the rain removal decreases 1.43 dB and 1.86 dB on Test1 and Test2 respectively. This indicates that our residual attention mechanism facilitates removing rain streaks significantly.

Different losses: To illustrate the impacts of different losses on the final deraining performance, we conduct experiments (A2), (B2), (C2), (D2) and (E2) in Table 3. We demonstrate that the perceptual loss and SSIM loss improve the final deraining performance. Note that, only using the pixel-wise intensity similarity loss can achieve higher PSNR, but the derained results tend to be blurry. In (E2), adding the SSIM loss in stage 2 only improves 0.03 dB, but it costs more time to train M_2 compared to (A2). Thus, considering the trade-off between the performance and computational efficiency, we do not use the SSIM loss in stage 2.

5 Conclusion

We present a residual attention based multiscale deraining network to remove rain streaks in a coarse-to-fine manner. With the help of the pyramid framework, we can remove rain in different densities while recovering background progressively from structure to fine details. Our framework also eases the deraining procedure since rain becomes less obvious in coarse scales. Since the residual between the upsampled rain-free image and its corresponding rain image also provides important clues for localizing the regions contaminated by rain, our network can better focus on recovering rain-contaminated regions by embedding the residual information into the pyramid framework. Therefore, our method achieves superior deraining performance compared to the state-of-the-art.

Acknowledgement. This work was supported by the National Natural Science Foundation

Table 2: Ablation study of pyramid architecture and residual attention on Test1 and Test2.

	Metric	(A1) 512	(B1) 512/256	(C1) 512/256/128	(A0) wo/ Residual
Test1	PSNR(dB)	26.14	27.67	**29.41**	27.98
Test1	SSIM	0.87	0.88	**0.92**	0.88
Test2	PSNR(dB)	24.24	25.16	**27.21**	25.35
Test2	SSIM	0.86	0.88	**0.91**	0.88

Table 3: Ablation study of loss functions at different stages on Test1 and Test2.

Methods	PSNR(dB)(Test1)	SSIM(Test1)	PSNR(dB)(Test2)	SSIM(Test2)
(A2) $L_2+L_2L_P+L_2L_PL_S$	**29.41**	**0.92**	**27.21**	**0.91**
(B2) $L_2+L_2L_P+L_2L_P$	29.17	0.89	26.91	0.88
(C2) $L_2+L_2L_P+L_2$	29.23	0.87	26.52	0.86
(D2) $L_2+L_2+L_2L_PL_S$	29.29	0.91	26.98	0.90
(E2) $L_2+L_2L_PL_S+L_2L_PL_S$	29.32	0.92	27.04	0.90

of China (No.61601021), the Beijing Natural Science Foundation (L172022), China Scholarship Council (No.201807095044) and the Fundamental Research Funds for the Central Universities (2016RC015). The research in this paper was also supported by the Australian Research Council (DE140100180), and Australian Research Council Centre of Excellence for Robotic Vision (CE140100016).

References

[1] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International journal of computer vision*, 86(2-3):256, 2010.

[2] Yi Chang, Luxin Yan, and Sheng Zhong. Transformed low-rank model for line pattern noise removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1726–1734, 2017.

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

[4] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Xingping Dong, Jianbing Shen, Wenguan Wang, Liu Sheng Yu, Ling Shao, and Fatih Murat Porikli. Hyperparameter optimization for tracking with continuous deep q-learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 518–527, 2018.

[6] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

[7] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, pages 3855–3863, 2017.

[8] Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[9] Kshitiz Garg and Shree K Nayar. When does a camera see rain? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1067–1074. IEEE, 2005.

[10] Kshitiz Garg and Shree K Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

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

[15] Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[16] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Single image deraining using scale-aware multi-stage recurrent network. *arXiv preprint arXiv:1712.06830*, 2017.

[17] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *European Conference on Computer Vision*, pages 262–277. Springer, 2018.

[18] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016.

[19] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3233–3242, 2018.

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

[21] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

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

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Sonia Starik and Michael Werman. Simulation of rain in videos. In *Texture Workshop, ICCV*, volume 2, pages 406–409, 2003.

[25] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W.H. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

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

[28] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2018.

[29] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018.

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

[31] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017.