A pyramid non-local enhanced residual dense network for single image de-raining

Minghua Zhao\textsuperscript{1,2} | Hengrui Fan\textsuperscript{1} | Shuangli DU\textsuperscript{1} | Li Wang\textsuperscript{3} | Peng Li\textsuperscript{1} | Jing Hu\textsuperscript{1}

\textsuperscript{1} School of Computer Science and Engineering, Xi’an University of Technology, Xi’an 710048, China
\textsuperscript{2} Shaanxi Key Laboratory of Network Computing and Security Technology, Xi’an, Shaanxi, China
\textsuperscript{3} Personnel Department, Xi’an University of Technology, Xi’an 710048, China

Abstract

Single image de-raining based on convolutional neural network (CNN) has made considerable progress in recent years. However, usually the de-rained result has dark artifacts and image textures tend to be over-smoothed. In this paper, a pyramid non-local enhanced residual dense network is proposed to reduce such distortion. Firstly, the down-sampled images are input into the Laplacian pyramid, which can extract the overall and partial texture clues, and subsequently a set of images of different scales are produced. Secondly, these images are fed into a non-local enhanced residual dense block, which can not only capture long-distance dependencies of feature maps, but also fully utilizes the hierarchical features in every dense block, leading to high accuracy of rain streaks extraction and better preservation of image edge detail. Finally, the de-rained image is gradually restored by Gaussian reconstruction pyramid. Experimental results on both synthetic data and real-world data show that the artifacts distortion is obviously reduced by the proposed network. And the quality of de-rained image is significantly improved compared with the state-of-the-art methods.

1 | INTRODUCTION

Rain not only affects human visual perception, but also outdoor vision systems, i.e. self-driving cars and surveillance systems. Usually, for these systems, clear image and video are used as input. Thus, their robustness is easy to be reduced during rainy day. In recent years, de-raining has become an important research topic, and many approaches have been proposed. They can be classified into two types: traditional methodology-based approaches [1–9] and deep learning-based approaches [10–23]. Traditional methodology-based approaches focus on analysing the appearance of single rain streak and global rain distribution in image space: such as supposing rain streaks following ellipse shape [1–4], low rank pattern [5, 7, 8] and Gaussian mixture model (GMM) [6]. However, due to the complex imaging scenes (e.g., wind and fog), unknown camera parameters (e.g., focal length, exposure time), rain streak shape is complex, changeable and thus does not follow a pre-designed rain model. As a result, the traditional methods cannot obtain satisfactory results in real scenes. In addition, such methods require complex optimization and thus are time-consuming. In contrast, the deep-learning based methods can perform better in real scenes and are more efficient since training images are diverse, and the pre-trained model can be directly used in the rain removal stage. At present, the following deep learning methods are widely used in the de-raining task: supervised learning [10–17], semi-supervised learning [18, 19], unsupervised learning [20–23]. They have made a great progress, and the de-raining performance is significantly improved. However, black artifacts along rain streaks and over-smoothed texture edges still are the touchy issues. In this paper, we propose a pyramid non-local enhanced residual dense network, which consists of a series of non-local enhanced residual dense blocks and Gauss-Laplacian pyramid network to efficiently learn the mapping function between image pairs for accurately modelling the rain streaks while perfectly preserving...
the texture information of the image. The contributions of this work are twofold:

1. A novel CNN-based model named a pyramid non-local enhanced residual dense network is proposed, which can obtain superior performance on removing rain streaks compared to other methods.

2. We integrate non-local enhanced residual dense block (NEDB) and Gauss-Laplacian pyramid. The de-raining advantages of the two modules are fully utilized. NEDB can help our network capture the long-distance dependencies of feature maps and make full use of hierarchical features of convolution layers. Moreover, the multi-scale decomposition of the Gaussian–Laplacian pyramid is adopted in our network, which makes use of multi-task supervision in the training stage. Thereby, the problem of black artifacts and over-smoothed edges in the image is better avoided.

2 RELATED WORK

Image de-raining approaches can be divided into two categories: single image-based methods and video-based methods. Here, only single image-based methods are discussed. Sparse coding and dictionary learning are two popular techniques in traditional methodology-based approaches, which are usually used to divide the rain image into a background layer and a rain layer [1–9]. Chen et al. [5] proposed a low-rank pattern to model rain distribution. Li et al. [6] use GMM to model rain streak with different directions and different scales. The model constructed by the existing traditional methods have limited capacity in de-raining. In recent years, many methods based on deep learning have been proposed, including supervised learning [10–17], semi-supervised learning [18, 19], un-supervised learning [20–23]. Supervised learning approaches tend to learn the mapping between ground-truth clear image and synthetic rainy image. Fu et al. [10] first proposed a CNN-based de-raining network, which cannot effectively remove rain streaks because only three convolutional layers are used. Subsequently, residual block is widely used in de-raining task [11–13]. To better remove rain streaks, researchers focus on developing new network architectures. A fully convolutional network based on region-dependent rain image model was proposed by Yang et al. [14] to jointly detect and remove rain. The restored image has over-smoothed edges. Then, the authors propose a higher version of JORDER-E to recover image details. Considering the fog-like visual effect caused by rain accumulation, an end-to-end neural network was designed by Hu et al. [15] to learn the depth attention features. With the improvement of de-raining performance, the learning network becomes more and more complex and requires a lot of parameters. To reduce the number of parameters, Fu et al. [16] proposed a lightweight LPnet. It does not make full use of the hierarchical features of the original image, resulting in unsatisfactory results. A non-local enhanced encoder-decoder network (NLEDN) was proposed by Li et al. [17] to restore detail texture. However, supervised learning based methods have weak generalization ability to handle real rainy images. To solve this problem, a semi-supervised de-raining model was first introduced by Wei et al. [18], this is realized by calculating the residual between the input image and the expected clean image to simulate the rain pattern distribution. Moreover, Yasarla et al. [19] proposed a semi-supervised learning network based on Gaussian processes, which can model latent space vectors of unlabelled, then was used to compute pseudo-ground-truth for supervising the network.

Compared with semi-supervised learning, unsupervised learning is more conducive to practical applications. Zhu et al. [20] proposed a Rain Removal-GAN (RR-GAN), which mainly defines a novel multiscale attention memory generator and a new multiscale deeply supervised discriminator. Meanwhile, an Unsupervised De-raining GAN (UD-GAN), using a new self-supervised learning method was proposed by Jin et al. [21]. Rain guidance module and background guidance module is designed to take advantage of rainy image characteristics. To generate better de-raining results, Wei et al. [22] proposed an attention guided de-raining network by constraining the CycleGAN of [23], they utilize two cycle consistency branches to extract rain streak by paying attention to the relationship between rainy image and clean image. Although these methods achieve significant success, there exist two touchy issues: (1) Dark artifacts along with the rain streaks; (2) Image textures tend to be over-smoothed. To reduce such distortion, in this paper, a pyramid non-local enhanced residual dense network is proposed.

3 METHOD

For the first time, a pyramid non-local enhanced residual dense network is adopted in the field of image de-raining. The overview architecture of our proposed network is shown in Figure 1. Interestingly, this combination of non-local enhanced block and pyramid has been exploited by [24] for de-noising and high-resolution reconstruction, which is sufficient to prove that our method is feasible. The design of the Multi-scale model (e.g., image pyramid) can obtain more powerful feature expression and reduce the number of parameters. Moreover, non-local enhanced residual dense block (NEDB) is introduced into our network. The NLB and RDB of NEDB are adopted in each layer of the pyramid to improve the de-raining effect of our proposed method.

Specific steps are as follows, first, a rainy image is decomposed into its Laplacian pyramid by five down-sampling operations, then two convolutional layers are used for extracting shallow features. Second, the five layers singly pass through the two convolutional layers to extract shallow features, and then the feature maps are fed into the non-local enhanced block. The weighted summation method is used to improve the receptive field of the neural network and enhance long-distances of feature maps, thereby providing rich information for the following layer. Subsequently, the feature maps are input into a dense
block with residual mapping for learning rain streaks, which can make full use of hierarchical features of convolution layers. Then it passes through residual blocks to obtain the de-rained result.

In the following sections, each part of the network architecture is introduced in detail.

### 3.1 Gaussian–Laplacian pyramid

The pyramid network for rain removal is first proposed by Fu et al. [16], which can get a satisfactory result with fewer parameters. The sub-network of Lpnet is shown in Figure 2. However, the de-rained result still contains dark artifacts due to the following reasons. First, Lpnet does not fully use hierarchical features, which makes the edge information hard to maintain during the process of de-raining. Second, the sub-network of Lpnet adopts a shallow CNN structure, the size of the network receptive field is limited. Therefore, the modelling of rain patterns has low accuracy, and the ability of de-raining performs worse. Finally, Resnet [25] is used to remove rain streaks in Lpnet, it is problematic to deal with black artifacts in heavy rainy images. The Resnet combine feature maps through summation before passing them to the layer, it will affect the
information flow of the feature maps. In this paper, a pyramid non-local enhanced residual dense network is introduced to improve it in several ways: (1) In non-local enhanced block, the weighted summation method is used to improve the receptive field of the neural network for the following layer. (2) To accurately model rain streaks, a residual dense network consisting of five densely connected convolutional layers is used to obtain the feature maps with rich details. (3) Compared with the LPnet, the NEDB combines feature maps by cascading features. Therefore, the feature maps are passed to all subsequent layers. Figure 3 demonstrates the comparison of LPnet results and our network results.

As the method described in [16], we learn that the background information can be completely extracted at the top of the Laplacian pyramid, other layers contain rain streaks and detailed information at different spatial scales. Each subnet only needs to deal with a single scale of high-frequency components. The following equation is used to calculate the \( i \)-th Laplacian pyramid:

\[
L_i(r) = G_i(r) - \text{upsample}(G_{i+1}(r)) \quad (i \in [0, n])
\]  

where \( r \) is the input rainy image, \( n \) is the number of pyramid layers, \( L_i(r) \) is the \( i \)-th Laplacian pyramid layer, \( G_i(r) \) is the \( i \)-th Gaussian pyramid layer, and \( G_{i+1}(r) \) is computed by down-sampling \( G_i(r) \) using Gaussian filter kernel, \( G_1(r) = r \) and \( L_0(r) = G_n(r) \).

The Gaussian pyramid reconstruction is the layer restored by the residual blocks (RBS), and the bottom layer \( n \) is our final output:

\[
G_i(r) = \sigma(L_i(r) + \text{upsample}(G_{i+1}(r))) \quad (2)
\]

\[
G_n(r) = \sigma(L_n(r)) \quad (3)
\]

where \( \sigma \) is an activation function. The reconstruction layer \( G_i(r) \) is obtained through the activation function leaky ReLU (LReLU) [26], and the \( G_i(r) \) is used to guide the output result of the \( G_{i+1}(r) \) through the up-sampling operation. The final result is the bottom layer \( G_n(r) \) of the Gaussian pyramid.

### 3.2 Non-local enhanced residual dense block

According to [17], non-local enhanced block with dense block is combined to obtain a more accurate modelling of rain streaks. In order to solve the problem of increased computational burden due to the expansion space of the feature maps at the bottom layers of the pyramid, grid divisions are used for the feature maps before inputting into a non-local enhanced block (NLB). For higher layers, fewer parameters are used. Thus, the number of grid divisions are set from low to high levels as 8, 6, 4, 2, and 1 respectively.

Compared with the traditional convolution layer with local operations, region non-local enhanced residual dense block (RNEDB) has a larger receptive field of the neural network, thereby maintaining the edge while effectively removing the rain band. Before the \( L_n(r) \) is input into non-local enhanced residual dense blocks, two convolutional layers are first adopted to extract features, which can combine both non-local and local information to build a richer hierarchy in our network. After the first convolutional layer, skip-connection is used to reduce the problem of information loss, which can not only provide...
Figure 3: The comparison of de-rained results Lpnet and our network in some layers of pyramid. From left to right, each image represents $G_1(r)$, $G_2(r)$, $G_3(r)$. (a–c) The results of sub-network in Lpnet. (d–f) The results of our proposed method.

Information compensation in the network, but also help gradient backpropagation and pixel-level prediction by adding residual learning.

3.2.1 On-local enhanced block (NLB)

The principle of non-local blocks is to calculate a position response as a weighted sum of all position features, which has been applied to the video classification task [27]. A non-local enhanced block is shown in Figure 4. The non-local operation is defined as:

$$y_{k,i} = \frac{1}{\varepsilon(P)} \sum_{j} f(P_{k,i}, P_{k,j}) g(P_{k,j})$$

where $P_{k,i}$, $P_{k,j}$ denote the feature map $P_k$ at position $i$, $j$ respectively. A pairwise function $f$ computes a scalar between $i$ and all $j$. The unary function $g$ computes a representation of the input feature at the position $j$. The output feature map is normalized by a factor $\varepsilon(P)$. Equation (4) can fit various sizes of input images, which can be embedded in many networks.

3.2.2 Residual dense block (RDB)

The enhanced feature maps are fed into the residual dense block, a residual dense block is shown in Figure 5. It consists of five closely connected convolution layers.
where \([D_0, \ldots, D_{k-1}]\) represent a series of feature maps generated by densely connected layers. \(H_k\) is a comprehensive function of two continuous operations: ReLU and a \(3 \times 3\) convolution layer. The local residual learning is introduced to improve the network representation ability, which can learn more abstract feature representation. The residual dense block has two advantages.

1. The Gaussian pyramid, using the up-sampling operation, contains the advantages of the dense network, namely making full use of the hierarchical characteristics and alleviating gradient disappearance.
2. The image of each layer in the pyramid is sparse, which does not obey Gaussian distribution. Thus, dense blocks can remove the batch normalization (BN) layer \([28]\) by using image pyramid technology, to reduce the use of GPU memory.

### 3.3 Loss function

#### 3.3.1 Pixel-wise loss

Given a training set \(\{R_m, \hat{R}_m\}_{m=1}^N\), \(R\) is an input rainy image, \(G(R)\) and \(\hat{R}\) denotes the de-rained result and corresponding ground truth respectively. The \(L_1\) norm loss over \(N\) samples can be written as:

\[
L_1 = \sum_{m=1}^N || G(R_m) - \hat{R}_m ||_1
\]

The distortion is measured between de-rained image and the ground-truth in image pixel space.

#### 3.3.2 SSIM loss

The SSIM loss is proposed to measure structural similarity between de-rained image and ground-truth. SSIM is defined as

\[
\text{SSIM}(G(R_m), \hat{R}_m)
\]
where $\mu_r$, $\sigma_r^2$ represent the mean and the variance of $y$. $\sigma_{G(R^n)}(\hat{y}_r)$ is the covariance of $G(R^n)$ and $\hat{y}_r$. For maintain stability of SSIM loss, the constants $C_1$ and $C_2$ are used. Different from the SSIM evaluation index, the SSIM loss is range from 0 to 1. Thus, The SSIM loss function defined as

$$L_{SSIM} = 1 - SSIM(G(R), \hat{R})$$

$$L_T = \frac{1}{N} \sum_{n=1}^{N} \left( \lambda_1 \sum_{u=1}^{P} L_1 + \lambda_2 \sum_{u=1}^{2} L_{SSIM} \right)$$

where $\lambda_1, \lambda_2$ are positive weights. Pyramid level $P = \{1, 2, 3, 4, 5\}, N$ is the number of training data, $L_1$ and SSIM are adopted in our loss functions. Specifically, in levels $\{1, 2\}$, they contain finer details and more complete rain streaks. According to [29], the $L_1$+SSIM is adopted as the loss function to better preserve detailed information. The loss function $L_1$ can be used as smooth background area and crude structural information exists in levels $\{3, 4, 5\}$.

### 4 | EXPERIMENT

Our proposed network is compared against several state-of-the-art single image de-raining methods, including Gaussian mixture models (GMMs) of [6], Deep detail network (DDN) of [11], Discriminative sparse coding (DSC) of [9], lightweight pyramid network (Lpnet) of [16], and no-locally enhanced encoder-decoder network (NLEDN) of [17]. For fair comparison, all CNN-based methods are retrained on the same dataset.

#### 4.1 | Comparison with the state-of-the-art methods

The visual results in each dataset are shown in Figures 6–12. The following points can be seen. First, GMM [6] and DSC [9] fail to remove rain streak from heavy rainy images, although they try to model complex rain streaks by diverse well-designed priors, they are only applicable to specific patterns instead of irregular in different datasets. Second, under the condition of heavy rain, DDN [11] and Lpnet [16] are able to remove rain streak but tend to generate obvious artifacts. Compared with the DDN and Lpnet, our network takes into account the importance of the long-distance dependencies and the hierarchical features, which can learn abstract feature representation, that is, our method can remove more rain streaks while perfectly preserving the image detail. Third, under the condition of light rain, our method has comparable visual results with NLEDN [17] and outperforms other methods. Fourth, the edges of the region produced by the proposed method are clearer than the other methods. Moreover, the PSNR and SSIM values of our proposed method are higher than the other five methods.

Compared with NLEDN, our method has better edge-preserving capability in the heavy rainy images. From Table 1, we can note that our network contains fewer parameters. The number of parameters is reduced in our method by removing the BN layer. Even with fewer parameters, our method still has a better performance.

Since real-word rainy images contain the irregular distribution of rain streaks, the scenarios of these images are very complex. Figure 12 shows the visual comparison of state-of-the-art methods on real-world data. As can be seen, our method has superior performance. This is because the Laplacian pyramid technology is used to decompose rainy images, each layer of the pyramid only needs to deal with high-frequency components at a single scale, it can help our network to deal with complex scenarios in real-world data.

#### 4.2 | Datasets and evaluation criteria

The performance of our proposed network is evaluated on both synthetic datasets and real data. For real-world data, some of the images are collected from the Internet and some from the real dataset provided by [16]. For synthetic data, the following four datasets are adopted.

1. The first dataset was proposed by [6]. Figure 6 shows visual comparison of de-rained results with state-of-the-art methods on the rain12 dataset.
2. The second dataset is Rain100L, which consists of 200 image pairs for training and the other 100 images for testing. All images contain light rain streaks, so the edge information can be easily maintained. Figure 7 demonstrates visual comparison of de-rained results with state-of-the-art methods on the rain100L dataset.
3. The third dataset is Rain100H, which contains 1800 images for training and 100 images for testing. There is a big gap between images in Rain100H and real rain images, as the rain streaks of images in Rain100H is more like noise. The robustness of our proposed network is enhanced by training in this dataset. Figure 8 shows visual comparison of de-rained results with state-of-the-art methods Both Rain100L and Rain100H were proposed by [14].
4. The DIDMDN dataset consists of 12,000 image pairs, each image is labelled for its rainfall intensity, and there are only three types of labels for the entire dataset (e.g. light, medium and heavy rain streak). Follow Zhang et al. [30], the 9100 image pairs are selected for training and the remaining 2900
FIGURE 6 Visual comparison of de-rained results with state-of-the-art methods on Rain12 [6]

TABLE 1 Performance comparisons of state-of-the-art rain removal methods

| Method    | Rain100L | Rain100H | Rain12 | DID-MDN |
|-----------|----------|----------|--------|---------|
| Metric    | PSNR     | SSIM     | NIQE   | PSNR    | SSIM    | NIQE   | PSNR    | SSIM    | NIQE    | Parameters (K) |
| DSC[9]    | 23.37    | 0.868    | 4.3236 | 17.75   | 0.320   | 11.7264| 30.07   | 0.866   | 4.6428  | —        |
| GMM[6]    | 28.71    | 0.876    | 3.8549 | 15.03   | 0.434   | 6.3816 | 32.02   | 0.912   | 4.1564  | —        |
| DDN[11]   | 25.99    | 0.814    | 3.9273 | 16.02   | 0.726   | 5.2573 | 31.41   | 0.904   | 4.6781  | 57K      |
| Lpnet[16] | 33.40    | 0.962    | 2.9156 | 24.48   | 0.826   | 3.2780 | 34.72   | 0.913   | 4.3236  | 7.5K     |
| NLEDN[17] | 36.57    | 0.974    | 2.5690 | 30.38   | 0.893   | 3.2403 | 35.79   | 0.957   | 3.4023  | 1519K    |
| Our       | 36.64    | 0.974    | 2.4891 | 30.46   | 0.896   | 3.1285 | 35.81   | 0.968   | 3.0525  | 670K     |

TABLE 2 Comparison of running time (in s)

| Image size | GMM with CPU | DSC with CPU | DDN | Lpnet | NLEDN | Our |
|------------|--------------|--------------|-----|-------|-------|-----|
| 250 × 250  | 167.9        | 45.9         | 0.22| 0.10  | 0.27  | 0.19|
| 500 × 500  | 584.7        | 178.6        | 0.31| 0.12  | 0.83  | 0.36|
| 750 × 750  | 1467.4       | 383.9        | 0.50| 0.16  | 1.60  | 0.72|
image pairs for evaluation. A visual comparison is provided in Figure 9.

The performance of synthetic data using three metrics, including peak signal-to-noise ratio (PSNR) [31], structure similarity index (SSIM) [32] and natural image quality evaluator (NIQE) [33]. PSNR and SSIM are adopted as full reference image quality assessment, and NIQE is used as no-reference image quality assessment.

### Table 3: PSNR and SSIM comparison for different loss functions

| Dataset  | MSE | $l_1$ | SSIM | SSIM | MSE + SSIM + $l_1$ |
|----------|-----|-------|------|------|-------------------|
| DID-MDN  | PSNR| 32.51 | 0.820| 0.806| 33.49             |
|          | SSIM| 0.820 | 0.806| 0.817| 0.933             |
|          | PSNR| 30.62 | 0.806| 0.817| 0.933             |
|          | SSIM| 0.817 | 0.806| 0.817| 0.933             |
|          | PSNR| 31.73 | 0.817| 0.817| 0.933             |
|          | SSIM| 0.817 | 0.806| 0.817| 0.933             |
|          | MSE | 33.25 | 0.933| 0.933| 0.902             |
|          | SSIM| 0.933 | 0.933| 0.933| 0.902             |

### Table 4: Ablation study on different components of our proposed network

| Pyramid module | Non-local enhance block | Residual dense block | DID-MDN dataset | PSNR | SSIM |
|----------------|-------------------------|---------------------|-----------------|------|------|
| No             | No                      | No                  | DID-MDN         | 26.14| 0.806|
| Yes            | No                      | No                  | DID-MDN         | 27.75| 0.873|
| Yes            | Yes                     | No                  | DID-MDN         | 33.23| 0.918|
| Yes            | Yes                     | Yes                 | DID-MDN         | 33.49| 0.933|

### 4.3 Parameter settings

Our network is implemented using TensorFlow, all experiments were performed on a desktop PC with I7-10700 CPU and NVIDIA GTX 2080Ti GPU. According to [16], combined with our own experiment, kernel numbers $k$ are set to 16, 8, 4, 2, and 1 from low to high layers. The fixed smoothing kernels [0.0625, 0.25, 0.375, 0.25, 0.0625], which are used to decompose the RGB image and reconstruct the Gaussian pyramid, the loss weights $\lambda_1, \lambda_2$ are set to 2,1. The Leaky ReLU is
adopted as activation function $\sigma$. The whole training process is conducted under multi-task supervision. We set the learning rate at 0.001. The input image size is resized to $256 \times 256$, the batch size is 128, and the training ends after 10,000 epochs. During training, the Adam optimizer is adopted to update the parameters of network.

4.4 Ablation study

4.4.1 Different components of our proposed network

In this section, the different components are tested to analyse their impact on de-raining performance. To facilitate experimental comparison, this experiment is conducted on the DIDMDN-data.

Pyramid module. The Gauss-Laplacian pyramid is introduced into our network. The results are shown in Table 4. As can be seen, the value of SSIM and PSNR are improved by using pyramid module. We can note that the PSNR and SSIM have about 1.61 db, 6.7% improvement, respectively.

Non-local enhance block: We add the non-local enhance block into the Gauss-Laplacian pyramid network. From Table 4, we can note that the PSNR and SSIM have about 5.48 db, 4.5% improvement, respectively.

Residual dense block: We add the residual dense block into our network. From Table 4, we can note that the PSNR and SSIM have about 0.26 db, 1.5% improvement, respectively.

4.4.2 Loss function

The loss function of our method is mentioned in Section 3.3. To verify the effectiveness of the loss function, we select the different components for test. Table 3 shows the PSNR and SSIM comparison for different loss functions. It can be seen that incorporating SSIM in loss function can clearly improve the performance since SSIM measures the local image feature, which is consistent with human visual perception characteristics. The performance is clearly improved.
FIGURE 9  The visual comparison of de-rained results with state-of-the-art methods on DIDMDN [30]

FIGURE 10  Visual comparison of de-rained results with state-of-the-art methods on three synthesized benchmark datasets
FIGURE 11  The comparison results of our proposed method with [16, 17] on Rain100H [14]

FIGURE 12  Visual comparison of de-rained results generated from state-of-the-art methods on real-world data
4.5 Running time

The comparison of run time in seconds is shown in Table 2. To provide a fair comparison, all methods are executed on the same machine, and we follow the original setting of all the released codes. On average, our method takes about 0.19 s, 0.36 s, 0.72 s to obtain de-rained images of size 250 × 250, 500 × 500, and 750 × 750, respectively.

5 CONCLUSION

In this paper, a new pyramid non-local enhanced residual dense network is proposed to reduce the following distortions: dark artifacts along with the directions of rain streaks and over-smoothed de-rained results. Specifically, in the first step, a rainy image is decomposed into its Laplacian pyramid, which can use multi-scale decomposition technology to extract overall and partial information, and each layer only needs to deal with a single high-frequency component. In the second step, a series of non-local enhanced residual dense blocks (NEDB) are adopted to each layer of the Laplacian pyramid, it is designed as a concatenation of non-local enhanced block and residual dense block. The non-local enhanced block is used to enhance long-distance dependencies of feature maps, and the residual dense block is used to fully exploit the hierarchical features. Finally, the result of our method is reconstructed at the bottom of the Gaussian pyramid. Experimental results on real-world data and synthetic data show that our network can effectively remove rain streaks with maintaining edge texture information. Compared with the existing methods, the obtained results are more natural.

However, the restoration of background details in real rainy images and heavy rainy images are not clean enough. In future research work, we will focus on the development of an unsupervised method to obtain optimal de-raining results.

ACKNOWLEDGMENTS

The authors wish to thank the editor-in-chief, associate editor and reviewers for their insightful comments and suggestions. This work was supported by National Key Technology Research and Development Program of China (2017YFB1402103-3), National Natural Science Foundation of China (61901363, 61901362) and Natural Science Foundation Technology Research and Development Program of China (61901363, 61901362) and Natural Science Foundation of Shaanxi province, China (2017JQ-648, 2019JM-381, 2019JQ-729) and Key Laboratory Foundation of Shaanxi Education Department (20JS086).

ORCID

Minghua Zhao https://orcid.org/0000-0001-8062-2982

REFERENCES

1. Xu, J., et al.: Removing rain and snow in a single image using guided filter. In: 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, pp. 304–307 (2012).
2. Zheng, X., et al.: Single-Image-Based Rain and Snow Removal Using Multi-Guided Filter. In: Lee M., Hirose A., Hou Z. G., Kil R. M. (eds) Neural Information Processing. ICONIP 2013. Lecture Notes in Computer Science, vol. 8228, pp. 258–265, Springer, Berlin, Heidelberg (2013).
3. Ding, X., et al.: Single image rain and snow removal via guided L0 smoothing filter. Multimedia Tools Appl. 75(5), 2607–2712 (2016).
4. Kim, J.-H., et al.: Single-image deraining using an adaptive nonlocal means filter. In: 2013 IEEE International Conference on Image Processing, Melbourne, pp. 914–917 (2013).
5. Chen, Y.-L., Hsu, C.-T.: A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: Proceedings of the IEEE International Conference on Computer Vision, Sydney, pp. 1968–1975 (2013).
6. Li, Y., et al.: Rain streak removal using layer priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, pp. 2736–2744 (2016).
7. Du, S., et al.: Single image deraining via decorrelating the rain streaks and background scene in gradient domain. Pattern Recognit. 79, 303–317 (2018).
8. Du, S., et al.: Rain streaks removal from single image based on texture constraint of background scene. Neurocomputing 419, 224–238 (2021).
9. Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, pp. 3397–3405 (2015).
10. Fu, X., et al.: Clearing the skies: A deep network architecture for single-image rain removal. IEEE Trans. Image Process. 26(6), 2944–2956 (2017).
11. Fu, X., et al.: Removing Rain from single images via a deep detail network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, pp. 1715–1723 (2017).
12. Fan, Z., et al.: Residual-guide feature fusion network for single image deraining. arXiv preprint arXiv:1804.07493 (2018).
13. Wang, H., et al.: Structural residual learning for single image rain removal. Knowledge-Based Syst. 213, 106595 (2020).
14. Yang, W., et al.: Deep joint rain detection and removal from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, pp. 1685–1694 (2017).
15. Hu, X., et al.: Depth-attentional features for single-image rain removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Long Beach, CA, pp. 8022–8031 (2019).
16. Fu, X., et al.: Lightweight pyramid networks for image deraining. IEEE Trans. Neural Networks Learn. Syst. 31(6), 1794–1807 (2019).
17. Li, G., et al.: Non-locally enhanced encoder-decoder network for single image de-raining. In: Proceedings of the 26th ACM international conference on Multimedia. Seoul, pp. 1056–1064 (2018).
18. Wei, W., et al.: Semi-supervised transfer learning for image rain removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Long Beach, CA, pp. 3872–3881 (2019).
19. Yasarla, R., Sindagi, V., Patel, V.: Semi-supervised image deraining using Gaussian processes. arXiv preprint arXiv:2009.13075 (2020).
20. Zhu, H., et al.: Singe image rain removal with unpaired information: A differentiable programming perspective. In: Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, HI, pp. 9332–9339 (2019).
21. Jin, X., et al.: Unsupervised single image deraining with self-supervised constraints. In: 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 2761–2765 (2019).
22. Wei, V., et al.: Derainyclegan: An attention-guided unsupervised benchmark for single image deraining and rainmaking. arXiv preprint arXiv:1912.07015 (2019).
23. Zhu, J.-Y., et al.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy, pp. 2242–2251 (2017).
24. Zhu, F., Fang, C., Ma, K.-K.: Pnen: Pyramid non-local enhanced networks. IEEE Trans. Image Process. 29, 8831–8841 (2020).
25. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, pp. 770–778 (2016).
26. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th International Conference on Machine Learning. Atlanta, GA (2013).
27. Wang, X., et al.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, pp. 7794–7803 (2018)
28. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
29. Zhao, H., et al.: Loss functions for image restoration with neural networks. IEEE Trans. Comput. Imaging 3(1), 47–57 (2016)
30. Zhang, H., Patel, V. M.: Density-aware single image de-raining using a multi-stream dense network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, pp. 695–704 (2018)
31. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of Psnr in image/video quality assessment. Electron. Lett. 44(13), 800–801 (2008)
32. Wang, Z., et al.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13(4), 600–612 (2004)
33. Mitral, A., Soundararajan, R., Bovik, A.C.: Making a “Completely Blind” image quality analyzer. IEEE Signal Process. Lett. 20(3), 209–212 (2012)