IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 11, 2020, accepted January 26, 2020, date of publication February 10, 2020, date of current version February 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972909

Double Recurrent Dense Network for Single Image Deraining

YANG LAN, HAIYING XIA, HAISHENG LI, SHUXIANG SONG, AND LINGYU WU
College of Electronics Engineering, Guangxi Normal University, Guilin 541004, China

Corresponding author: Haiying Xia (xhy22@mailbox.gxnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61762014 and Grant 61762012, in part by the Science and Technology Project of Guangxi under Grant 2018JJA170083 and Grant 2018JJA170089, and in part by the Innovation Project of Guangxi Graduate Education.

ABSTRACT Rain streaks can affect visual visibility, and hence disable many visual algorithms. So we present a double recurrent dense network for removing rain streaks from single image. Assume the rain image is the superposition of the clean image and the rain streaks, we directly learn the rain streaks from the rainy image. In contrast to other models, we introduce a double recurrent scheme to promote better information reuse of rain streaks and relative clean image. For rain streaks, the LSTM cascaded by DenseNet blocks is used as the basic model. The relative clean image predicted by subtracting the rain streaks from the rainy image is then input to the basic model in an iterative way. Benefiting from double recurrent schemes, our model makes full use of rain streaks and image detail information and thoroughly removes rain streaks. Furthermore, we adopt a mix of L_1 loss, L_2 loss and SSIM loss to guarantee good rain removing performance. We conduct a plenty of experiments on synthetic and real rainy images, even on similar denoise task, the results not only show our model significantly outperforms the state-of-art methods for removing rain streaks, but also exhibit our model has a high effectiveness for similar task, i.e. image denoising.

INDEX TERMS Deraining, recurrent neural network, double, dense network.

I. INTRODUCTION

Rain is a natural and common weather phenomenon and will cause the objects in an image blurred due to the influences of light refraction and scattering on rain streaks. Especially in heavy rain condition, the weather is so complex and changeable that the background is unclear and the noise distribution such as rain streaks is uneven. Since most of the computer vision methods are designed based on the assumption of clean inputs, their performances will be seriously degraded. Thus, deraining is a necessary step for computer vision applications. Although the performances of deraining methods have achieved great improvement in recent years, deraining from single image under clutter background is still a challenge task.

Up to now, more and more people are engaged in recovering the clean images from the rainy ones. For example, some traditional optimization methods [1]–[6], [1], [45] have presented to predict the rain streaks and the background from the rainy image in a separate way. However, the rainfall makes the composition of the rain layer complex and diverse, which makes the effect of these methods work not well. With the excellent results of deep learning in the field of computer vision, image deraining methods based on deep learning [7]–[10] have received extensive attention. Based on the inputs, these methods can be divided into two categories: video-based methods and single-image based methods. Since video-based methods can leverage the temporary information in adjacent frames, they are easier than the single-image based methods. More and more works by using the advantages of deep learning are proposed in recent years to remove the rain streaks from single image [8], [10], [24], [39]–[42].

The existing problem is that the above-mentioned related methods have achieved good performance in light rain, but there is still much room for improvement in heavy rain. The heavy rain image has a high density and uneven distribution of rain, which makes the rain removal task very challenging. Many models for this problem have been proposed including residual blocks [7], dilated convolution [8], [10], squeeze-and-excitation [10], and recurrent layers [10], [11], and multi-stage networks [10]. There are three limitations for these methods [7]–[11]. Firstly, according to [7], feature reused can improve network performance. These methods take the

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

context information into account but ignore feature reuse; Secondly, the loss functions of these methods [7], [8], [10] are based on the L_2 norm. L_2 norm suffers from the well-known shortcoming, that is, L_2 correlates poorly with image quality as perceived by observer [13]. A model with L_2 loss tends to result in a unnatural reconstruction. Finally, the current methods treat these rain streaks removal stages independently and do not consider their correlations.

In order to solve the above problems, we propose a double recurrent dense network for removing rain streaks from single image. The pipeline of our proposed network is shown in Fig. 2. In view of the optimization problems, we use a double recurrent scheme, which promotes better information reuse of rain streaks and relative clean image. Then, the rain streaks are removed stage by stage. Considering the relevance of each stage, we take the output of the previous stage with the original rainy image as the input for the next stage. Furthermore, we design a mix loss function of L_2 loss, L_1 loss and SSIM loss, which can guarantee good rain removing performance. A plenty of experiments has demonstrated that our proposed method outperforms the-state-of-art methods for deraining task, even for similar task, i.e. image denoising.

In sum, the contributions of this paper are listed as follows:

1) We propose a double recurrent dense network for the information reuse. The dense blocks connected with LSTM, named as RLDNet, promote the rain streaks information flow, while the cascaded RLDNet make full use of the derained result in previous stages.
2) As far as I know, this is the first study to compare the performance of various loss functions on the task of rain removal. We also propose a hybrid loss function, which proves that the loss function performs better than the existing loss function.
3) This is a general deep network model. Our network model is not only far more effective than current methods in removing light rain and heavy rain, but experiments shows that it is also superior to the existing Gaussian denoising network model in removing Gaussian noise.

II. RELATED WORK

The existing rain removal algorithms can be roughly divided into video-based methods and single image-based methods. The video-based approaches use inter-frame information between adjacent frames to identify rain areas and remove rain streaks, so it is relatively easy to remove rain from the video [18]–[22], [43].

Removing rain streaks from a single image is more challenging because there is less information available and no inter frame information between adjacent frames. Kang *et al.* [1] attempted to extract rain streaks and background details from high-frequency layer by sparse coding based dictionary learning. Luo *et al.* [4] proposed a framework of rain removal based on discriminative sparse coding. Li *et al.* [5] learned background from pre-collected natural images and rains from rainy images by utilizing two Gaussian mixture models (GMMs).

Recently, deep learning has been used in rain removal. Fu *et al.* [7], [23] first used deep learning to remove rain streaks from single image. Yang *et al.* [8] designed a deep recurrent dilated network to jointly detect and remove rain streaks. Li *et al.* [10] removed the rain streaks via multiple stages and used recurrent neural network to exchange information across stages. Zhang *et al.* [25] use the generative adversarial network (GAN) to prevent the degeneration of background when it is extracted from rain image, and utilized the perceptual loss to further ensure better visual quality. Liu *et al.* [24] proposed a novel symmetry enhanced network to explicitly remove the tilted rain streaks from rain images.

These methods based on deep learning improve the deraining performances of removing rain streaks from single image. But there is much room for improvement, especially in heavy rain condition. By far, no work has been done on studying double recurrent and hybrid loss functions for single image deraining. We empirically find that, the double recurrent scheme and a hybrid loss function can result in better deraining performance.

III. RAIN MODEL

The widely used rain model is assumed that the rain image is the superposition of the clean image and the rain streaks. Most of the deraining methods are based on this rain model:

$$R = B + S \tag{1}$$

where S is the rain streaks, B is the clean image and R is the rain image. As shown in eq. (1), the main task of the rain removal is to restore the clean image B from the rain image denoted as:

$$B = R - S \tag{2}$$

According to eq. (2), many works built the deraining network models under the assumption that the rain streaks existed in high frequency part. They decomposed the rain image into the base layer and the detail layer, and learned the differences between the detail layer with the rain streaks and the detail layer without rain streaks.

Statistically, most of the rain streaks are decomposed into the detail layer as shown in Fig. 1. It also shows obvious residuals of the rain streaks in the base layer shown in Fig. 1. This limits the improvement in deraining performance.

Instead of removing the rain streaks from the detail layer, we restore the clean image from the rain image. Suppose the deraining model is denoted by $\mathrm{H}(\theta)$, θ is input of the model, the deraining is to directly output the clean image from the inputs with the rain streaks.

$$\hat{S} = H(R) \tag{3}$$

where or indirect output, where $\hat{\mathrm{B}}$ is the prediction of the deraining model for the clean image B, and $\hat{\mathrm{S}}$ is the prediction for the rain streaks S.

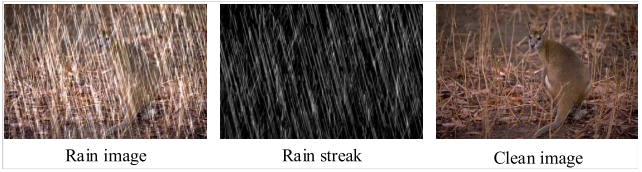


FIGURE 1. This is a pair of data images of the composite data set, Rain image is a combination of Rain streak and clean image.

In fact, we learn the prediction of the rain streaks from the model and then estimate the clean image by subtracting the predicted rain streaks from the rain image as described in the following.

$$\hat{B} = R - H(R) \tag{4}$$

The rain streak layer can be considered as a combination of rain streaks in multiple directions and other complex noises. Furthermore, it is more complicated in real weather conditions, for example, the raindrops have various appearances in the air. Cameras from different distances will also get irregularly distributed rain stripes. Especially in the case of heavy rain, the accumulation of rain streaks in the atmosphere will cause attenuation and scattering of light, further increasing the diversity of rain streaks in brightness. Although the rain streak layer has complicated textures, it is obviously shown in Fig. 1 than the complexity of the clean image is far more than the rain streak layer. Therefore, learning the rain streaks is simpler than learning the clean image directly from the rain image.

IV. THE PROPOSED DERAINING METHOD

Based on the rain model presented above, we present the deraining model, i.e. RDNet, and extend it to several varieties for handling the deraining task. In addition, we also directly apply our proposed model for image denoising task and validate the model universality for solving similar challenges. The architecture of our proposed network is shown in Fig. 2. It involves three steps: (1) the design of RDNet, (2) the design of RLDNet and (3) the loss function design of our proposed RLDNet. Based on the thought of the progressive removal of rain streak, we designed a recurrent dense network model (RDNet). To further improve RDNet's ability to remove rain streak, we integrate the LSTM with the RDNet to make it suitable for predicting the rain streaks. The integration of the LSTM and the RDNet benefit from each other and can obviously improve the performance of our proposed deraining model, we named it RLDNet. The RLDNet provides a possible way of using information in previous outputs to guide the later model learning. To guarantee the performance of the deraining model, we also redesign the loss function by using a mix of L_2, L_1 and L_{ssim} to train the model. The loss function combing L_2, L_1 with L_{ssim} ensures the improvement of our proposed model both in evaluation index and visual effects. Finally, we will discuss the advantages of our proposed deraining model and extend the RLDNet for the similar task, i.e. image denoising.

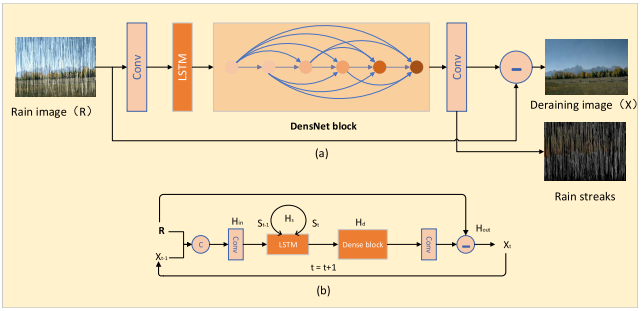


FIGURE 2. The subnetwork architecture of our proposed model consists solely of LSTM [16] and a DenseNet [14] block.

A. THE DESIGN OF RDNET

The RDNet we proposed is shown in Fig. 2. It can be divided into two steps. First, the de-raining image (X) is extracted through the network in Fig. 2 (a). Second, the rainy image (R) and the de-raining image(X) are recurrently input to the network to remove the rain streak as shown in Fig. 2 (b). Note that the RDNet network does not include the LSTM model, the LSTM model should be ignored in Fig. 2. The RDNet is mainly composed of three parts: (1) the input layer H_{in}, (2) the DenseNet layer H_d, and (3) the residual output layer H_{out}.

The input layer H_{in} is mainly used to perform a convolution operation to extract 32 rain streak information, and the rain streak information will be transmitted to DenseNet layer H_d. The DenseNet layer H_d is then placed to further encourage feature reuse for better information flow between layers. It is the most critical part to extract the rain streaks noise layer.The structure of DenseNet Block is shown in Fig. 2. It contains 5 layers: layer I, layer II, layer III, layer IV and layer V. Each layer has 32 filter maps followed by a non-linear ReLU function. The details of the DenseNet are also listed in Fig. 3.

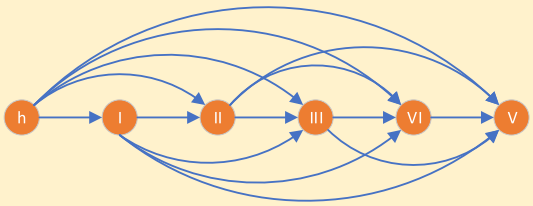
The output layer H_{out} completes a residual operation by subtracting the rain streaks from the rainy input. The relative clean image X_t can be computed by the following equation.

$$X_t = H_{out}(H_d(H_{in}(X_{t-1}, inputs))) \tag{5}$$

where the inputs are the original rainy image, X_{t-1} and X_t respectively represents the outputs of the RDNet in the $t-1^{th}$ and t^{th} iterations.

B. RLDNET ARCHITECTURE

LSTM is explicitly designed to remember information for a long period. And it has achieved incredible success in the tasks related to sequences and lists, such as speech recognition, image captioning, language translation. But the works in the fields unrelated to sequences or lists has rarely been reported. For solving the problems in image processing, iteration is a common method to improve the performances of the Network models. In the iterative schemes, the information in previous iterations is also very useful for current iteration. Hence, the LSTM model can be exploited to pass the useful information in previous iterations to the current iteration for better performance of single image deraining.



Layer	Operation	Feature map
I	Conv2d(32, 32, 3, 1, 1) + Relu()	Input: (b, 32, h, w) Output: (b, 32, h, w)
II	Conv2d(64, 32, 3, 1, 1) + Relu() Conv2d(32, 32, 3, 1, 1) + Relu()	Input: (b, 64, h, w) Output: (b, 32, h, w)
III	Conv2d(96, 32, 3, 1, 1) + Relu() Conv2d(32, 32, 3, 1, 1) + Relu()	Input: (b, 96, h, w) Output: (b, 32, h, w)
IV	Conv2d(128, 32, 3, 1, 1) + Relu() Conv2d(32, 32, 3, 1, 1) + Relu()	Input: (b, 128, h, w) Output: (b, 32, h, w)
V	Conv2d(160, 32, 3, 1, 1) + Relu() Conv2d(32, 32, 3, 1, 1) + Relu()	Input: (b, 160, h, w) Output: (b, 32, h, w)

FIGURE 3. The Dense network block with layer I, layer II, layer III, layer IV and layer V.

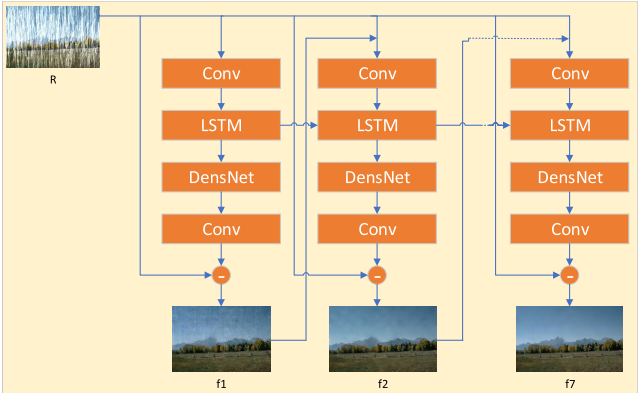


FIGURE 4. The framework of RLDNet. t1 represents the first iteration, f1 represents the output image of the rain streaks removal for the first iteration, f7 is the output image of the 7th iteration, and their weight values are shared.

Meanwhile, since the rain streaks have complicated textures caused by light scattering, light attenuation, irregular rain stripes and so on, it is a great challenge to remove the rain streaks from single rainy image. Although the integration of the LSTM model and the DenseNet block benefits from each other and achieves the-state-of-the-art performance, the performance of single image deraining still needs further improvement to satisfy the requirements in real world applications. So we further integrate the recurrent structure into the RDNet to remove the rain streaks in multiple stages, named as RLDNet.

The structure of the RLDNet is shown in Fig. 2(a) and (b). It adds the LSTM model on the basis of the RDNet. As shown in Fig. 2(b), it is mainly composed of four parts: (1) the input layer H_{in}, (2) the LSTM layer H_s, (3) the DenseNet layer H_d, and (4) the residual output layer H_{out}. The input layer H_{in} translates the input rainy image and the output X_{t-1} in previous stages into 32 feature maps by using 32 filters with size of 3*3, which are used as the input of the LSTM in later stages.

The LSTM layer H_s works with loops in them, which can pass the previous information of rain streaks to the next step. The structure of the LSTM can be found in [16]. It contains the state of the cell, which is a bit like a convey or belt. It runs through the entire model. We call it as S_t. This enriches the information needed for extracting rain streaks. The hidden outputs S_t is computed by Eq. (6), The relative clean image X_t can be computed by Eq. (7).

$$S_t = H_s(H_{in}(X_{t-1}, inputs), S_{t-1}) \quad (6)$$

$$X_t = H_{out}(H_d(H_s(H_{in}(X_{t-1}, inputs), S_{t-1}))) \quad (7)$$

where S_{t-1} and S_t respectively denote the hidden information of rain streaks calculated by the LSTM model in the $t-1^{th}$ and t^{th} iterations.

The complete recurrent structure of RLDNet is shown in Fig. 4, the recurrent structure can be regarded as the multiple copies of the same LDNet (As shown in the upper part of Fig. 2, the LSTM model and DenseNet block, named as LDNet, are connected in series, and there is no recurrence in the network framework.), each passing the current information to its successor. Different from the recurrent structure, the RLDNet has two information flow channels. The one introduced by the LSTM model mainly pass the information related to the rain streaks to its following stage. The other one introduced by the recurrent structure is responsible for delivering the outputs after rain removal in previous stage to the next stage. The information transmitted by the two channels is just complementary, one for the rain streaks, the other for the relative clean image. The complementary information flow further contributes to the improvement of the performance for single image deraining task.

Intuitively, the more times it iterates, the better performance of the rain streaks removal it achieves. Actually, the improvement of the performance is negligible when the number of iterations reaches a certain threshold as shown in Table 5.

By comparison between RLDNet and RDNet in Table 2, the average PNSR and SSIM values of RLDNet's is better than RDNet's under the same experiment setting and network architecture. It is because that RLDNet combines the feature reuse of dense networks and recurrent network memory units to improve the performance of single image deraining and the DenseNet and the LSTM can benefit from each other for the image deraining task. This view can be further verified by the following analyzes.

On the one hand, the DenseNet block benefits from the LSTM model. Although the DenseNet block can pass all the preceding information to the subsequent layers, the information flow only limits in the DenseNet block. In the iterative scheme, it still lack the useful information delivered from the previous iterations. With the LSTM model, the DenseNet block implicitly keeps the useful information for rain streaks removal in the hidden states. Hence, the LSTM model just

complements the missing of the useful information coming from the previous iterations.

On the other hand, the LSTM model benefits from the DenseNet block. The LSTM model can make full use of the information in previous iterations, but its capability of feature extraction is much weaker than the DenseNet block. Obviously, the capability of feature extraction directly affects the accuracy of the extraction of rain streaks. So the RLDNet with the DenseNet block has enough capability to exploit the necessary information for extracting the rain streaks from the rainy image.

To sum up, the integration of the LSTM model and the DenseNet block takes full use of their advantages. It not only can guarantee the converge in network training process, but also improve the performance for single image deraining.

C. RGDNET ARCHITECTURE

RGDNet architecture is a variant of the RLDNet. Instead of using the LSTM model, the RGDNet model combines the GRU unit with the RDNet. The difference between LSTM and GRU: First, the GRU parameter is less than LSTM, so it is easy to converge. When the data set is large, the performance of LSTM is better than that of GRU. Second, the performance of GRU and LSTM is similar on a general data set. Third, structurally speaking, the GRU has only two gates (update and reset), the LSTM has three (forget, input, output), the GRU directly passes its hidden state to the next unit, and the LSTM uses the storage unit to the hidden state is wrapped up.

D. LOSS FUNCTION

Suppose we have a training set $(B_i, R_i)_{i=1,2,\cdots,n}$, where n is the number of the training samples, R_i is the rainy image and B_i is the corresponding rain-free image for R_i. Then, the mean squared error (MSE), that is L_2 loss,is widely used to train a model with the least error between the rainy images and their ground truths. The L_2 loss leveraged as follows:

$$L_2 = \frac{1}{N}\sum_{i=1}^{N}(B_i - H(R)_i)^2 \tag{8}$$

Generally, the L_2 norm works well for image processing problems measured by the peak single to noise ratio (PSNR). But it correlates poorly with image quality as perceived by a human observer, that is, the high PSNR values can't guarantee a good visual effect. In fact, L_2 norm tends to generate the over-smoothed results due to the squared penalty at edges. The smoothed edges are unacceptable for the tasks related to image quality, such as image deraining, image denoising, and image deblurring. Thus, we explore the combinations of more different loss functions to alleviate the over-smoothed problem. Following [13], we add two L_1 norm and structural similarity index (SSIM) item into our loss function.

The L_1 norm is significantly different from the L_2 norm, it prefers to narrow some weights of the features to zero rather than penalizing the large errors as L_2 norm does. So, their convergence curves have much difference. The L_1 loss function can be simply expressed as follows:

$$L_1 = \frac{1}{N}\sum_{i=1}^{N}|B_i - H(R)_i| \tag{9}$$

SSIM [17] is an index to measure the measures structural similarity of two images. It is sensitive to the local structure changes and hence closely related to image quality perceived by an observer. The loss function of SSIM is:

$$SSIM(B, H(R)) = \frac{2\mu_B\mu_{H(R)} + C_1}{\mu_B^2\mu_{H(R)}^2 + C_1} * \frac{2\sigma_{BH(R)} + C_2}{\sigma_B^2\sigma_{H(R)}^2 + C_2} \tag{10}$$

where μ_B denotes the pixel mean value of the rain-free image, $\mu_{H(R)}$ is the pixel mean value of the predicted image, $\sigma_{BH(R)}$ is the covariance of the rain-free image B and the predicted image H(R). σ_B and $\sigma_{H(R)}$ are the standard deviations of B and H(R), respectively, $\sigma_{BH(R)}$ is the covariance of the image B and H(R). C_1, C_2 and C_3 are constants, $C_1 = (K_1 * L)^2$, $C_2 = (K_2 * L)^2$, generally $K_1 = 0.01$, $K_2 = 0.03$, $L = 255$. More details can be found in reference [17]. In general, the larger the SSIM value is, the better the visual effect of the predicted image is. If the SSIM value is 1, it indicates that the two images are identical. So, maximizing the SSIM can be converted to a minimization problem by the following function:

$$L_{ssim} = 1 - SSIM(B, H(R)) \tag{11}$$

To clearly show the improvement by the combinations of different loss functions, we adopt three combinations: the combination of L_2, and loss function and SSIM loss function, the combination of L_1 loss function and SSIM loss function and the combination of L_1 plus L_2 loss function and SSIM loss function. The three combinations are listed as follows:

$$\begin{aligned} L_{mix_1} &= \alpha * L_{ssim} + (1-\alpha) * L_2 \\ L_{mix_2} &= \alpha * L_{ssim} + (1-\alpha) * L_1 \\ L_{mix_3} &= \alpha * L_{ssim} + (1-\alpha) * (L_1 + L_2) \end{aligned} \tag{12}$$

L_{mix_1} is mixed with L_2 and L_{ssim}, L_{mix_2} is combined with L_1 and L_{ssim}, and L_{mix_3} is mixed with L_1, L_2 and L_{ssim}, where $\alpha = 0.85$.

Table 1 summarizes the results by six different loss functions in terms of PNSR and SSIM. The results are obtained by our proposed RLDNet with six iterations on Rain100H dataset. It can be seen that among the SSIM loss function, the L_1 loss function and the L_2 loss function, the SSIM loss function perform be stand the L_2 loss function perform worst measured by PNSR and SSIM. Perhaps surprisingly, the L_1 loss function has a significant improvement over the L_2 loss function. Motivated by this observation, we consider the introduction of the L_1 loss function to the design of the loss function for the proposed RLDNet. For the three combinations, they averagely achieve better PNSR results than the single loss functions. Concretely, L_{mix_3} increases by 0.24dB compared with L_{mix_1} and increases by 0.2dB

TABLE 1. The results obtained by using six different loss functions on the proposed RLDNet.

Loss function	PSNR	SSIM
L_2	27.69	0.856
L_1	28.25	0.871
L_{ssim}	28.59	0.891
L_{mix_1}	28.61	0.891
L_{mix_2}	28.65	0.891
L_{mix_3}	28.85	0.891

compared with L_{mix_2}. Especially, the performance obtains an obvious improvement by introducing the L_1 loss function and L_2 loss function together. By introducing the SSIM function in the three combinations, they all achieve high SSIM values. For our deraining task, we adopt L_{mix_3} to conduct the following experiments. performs best, and followed by.

V. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed RLDNet, we compare it with the state-of-the-art deraining methods on synthetic datasets and real-world datasets. In addition, we also compare the denoising results of our model with several state-of-art methods to validate the high performance of our proposed for handling similar problems. The training and all the subsequent experiments are conducted on a PC with Intel Core i3 CPU 8100, 16GB RAM and NVIDIA Geforce GTX 2070.

A. EXPERIMENT SETTINGS

1) TRAINING SETTINGS

For training, we crop plenty of training image patches (the size is 100*100) from the training images. The proposed network is trained on Pytorch platform. We use a batch size of 5 and set the stage of RLDNet as t $=$ 7. Leaky ReLU [31] is adopted to carry out the nonlinear operation and the ADAM algorithm [32] is taken with a start learning rate 0.0001 to optimize the model parameters. During the training, the learning rate is divided by 10 at 10000th, 15000th and 20000th iterations. As for the parameters of the state-of-art methods, we use the default parameters reported in the references [10].

2) QUALITY MEASURES

To evaluate the performance on synthetic image pairs, we adopt two widely used metrics, including peak single to noise ratio (PSNR) [27] and structure similarity index (SSIM) [17]. Since there are no ground truth for real-world rainy images, the performance on the real-world dataset can only be evaluated in terms of visual effect. We compare our proposed approach with seven state-of-the-art methods, including image decomposition (ID) [1], discriminative sparse coding (DSC) [4], A directional global sparse model for single image rain removal [45], layer priors (LP) [5], DetailsNet [7], joint rain detection and removal (JORDER) [8], Single Image Rain Removal via a Simplified Residual Dense Network (SRDN) [38], Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal [23], Density-aware single image de-raining using a multi-stream dense network (DID-MDN) [12], Recurrent Squeeze-and-Excitation Context Aggregation Net(RESCAN) [10] and S-DSEN [24]. As for the task of image denoising, we also adopt PNSR and SSIM to compare the performances of our proposed model with six representative methods, including image denoising by sparse 3-d transform-domain collaborative filtering (BM3D) [33], Weighted nuclear norm minimization with application to image denoising (WNNM) [36], Image denoising: Can plain neural networks compete with BM3D (MLP) [37], A flexible framework for fast and effective image restoration (TNRD) [29], Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising (DnCNN) [35], Toward a fast and flexible solution for cnn based image denoising. IEEE Transactions on Image Processing (FFDNet) [34].

B. DATASET

1) SYNTHETIC DATASET

It is difficult to get a large amount of rainy and rain-free pairs in the real world. So we use five synthetic datasets to train our network: Rain1000, Rain800, Rain100H, Rain100L and Rain12. Rain1000 is a large dataset provided by method [7], which contains two types of images: 1000 clean images and 14000 synthetic rainy images. Each clean image is used to generate 14 rainy images with different orientations and magnitudes. Meanwhile, the training set of this data set has 900 pairs of images, and its test set has 100 pairs of images. Zhang *et al.* [25] synthesize 800 rain images (Rain800) from randomly selected outdoor images, and split them into a test set with 100 image pairs and a training set with 700 image pairs. Yang *et al.* [8] collects and synthesizes 2 datasets, Rain100L and Rain100H. Rain100L is the synthesized data set with only one type of rain streak and Rain100H is synthesized data set with five streak directions. According to [8], The rain streaks are synthesized in two ways: (1) the photorealistic rendering techniques proposed by [46]; (2) the simulated sharp line streaks along a certain direction with a small variation within an image as shown in Fig. 1. Among the two datasets, Rain100H contains the most image pairs synthesized with the combination of rain streaks in various directions, which makes it hard to effectively remove all rain streaks. Hence, it is mainly chosen to validate the performance of our model, 1700 synthetic image pairs for training and 100 pairs for testing. Rain12 [5], which uses the photo-realistic rendering techniques proposed by Garg and Nayar [46], includes 12 synthesized rain images with only one type of rain streak. Among the five datasets, the Rain100H and the Rain800 are heavy rain, and the Rain1000, the Rain100L and the Rain12 are light rain.

2) DENOISING DATASET

To train a denoising model based on RLDNet, we prepare a dataset with image pairs $\{C_i, B_i\}_{i=1}^{N}$, where C_i is generated by adding AWGN with specific noise levels to the latent clean

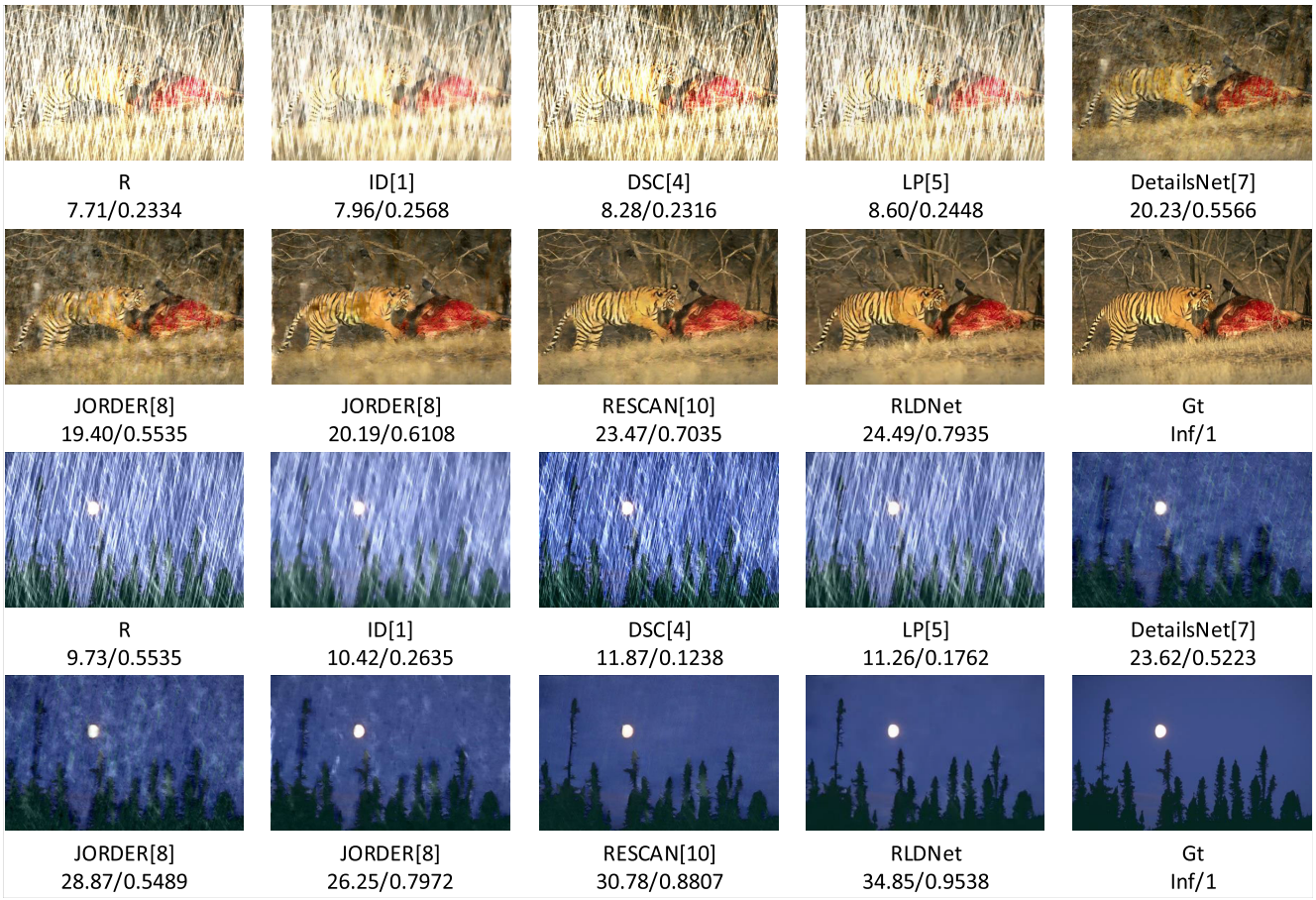


FIGURE 5. Results of various methods on synthetic images.

image B_i, that is, $C_i = B_i + n$. Following [28], we consider three noise levels: $\sigma = 15$, 25 and 50. we follow [29] to use 400 images of size 180x180 for training. For test, we choose two different test datasets for thorough evaluation, one is a test dataset containing 68 natural images from Berkeley segmentation dataset (BSD68) [30] and the other one contains 12 images as Set12. Note that all the test images are widely used for the evaluation of Gaussian denoising methods and they are not included in the training dataset.

3) REAL-WORLD DATASET

Zhang *et al.* [25] and Yang *et al.* [8] also provide a real-world dataset containing 92 images collected from the real-world. These images are diverse in terms of content as well as intensity and orientation of rain streaks. We use these images for objective evaluation.

C. RESULTS ON SYNTHETIC DATASET FOR SINGLE IMAGE DERAINING

Fig. 5 shows the results of synthetic images with heavy rain. As can be seen, ID, DSC and LP fail to remove the rain streaks in heavy rain. DetailsNet, JORDER and JORDER-R are able to remove most of the heavy rain streaks while also tend to generate obvious artifacts. Our RLDNet achieves comparable visual effects with RESCAN and outperforms the others. To clearly show the improvement, we also list the PNSR and SSIM values under each image. Compared with RESCAN, our RLDNet has the PNSR gains of 1.02dB for the first heavy image, 4.07dB for the second and the SSIM gains of 0.09dB for the first, 0.07dB for the second. The large gains of our model over the others demonstrate the effectiveness of the proposed RLDNet for removing heavy rain streaks on synthetic images.

We also adopt PNSR and SSIM to conduct quantitative evaluations on two synthetic datasets: Rain800 and Rain100H. Except for comparing with several state-of-art deraining methods, we evaluate the performances of the three variants of our proposed methods: RDNet, RLDNet and RGDNet. As shown in Table 2, ID, DSC and LP perform poorly both in PNSR and SSIM due to the limited capability of feature exploitation. The following five competing models built on CNN are all superior to the first three methods (ID, DSC and LP) without using CNN. Compared with over the best values among the competing methods, our RDNet obtains an average PNSR gain of 1.27dB on Rain800 dataset, an average PNSR gain of 0.3dB on Rain100H, an average SSIM gain of 0.01dB on Rain800 dataset, and an average SSIM gain of 0.02dB on Rain100H dataset. That's means

TABLE 2. Quantitative experiments evaluated on two synthetic datasets (Heavy rain). Best results are marked in bold.

Dataset	Rain800		Rain100H	
Measure	PSNR	SSIM	PSNR	SSIM
Rainy images	21.26	0.6479	12.23	0.3477
Method[45]	21.73	0.7035	13.50	0.4005
ID[1]	18.88	0.5832	14.02	0.5239
DSC[4]	18.56	0.5996	15.66	0.4225
LP[5]	20.46	0.7297	14.26	0.5444
DetailsNet[7]	21.16	0.7320	22.26	0.6928
JORDER[8]	22.24	0.7763	22.15	0.6736
JORDER-R[8]	22.29	0.7922	23.45	0.7490
RESCAN[10]	24.09	0.8410	26.45	0.8458
S-DSEN[24]	23.64	0.8379	27.16	0.8379
RDNet	25.36	0.8504	27.46	0.8686
RGDNet	25.74	**0.8530**	28.70	0.8900
RLDNet	**26.47**	0.8468	**28.96**	**0.8922**

our RDNet defeats all the competing methods by only combing the recurrent with the DenseNet block. Furthermore, by add the information flow of rain streaks in iterations, the RLDNet and RGDNet both further improve the performance of our proposed methods. Especially on Rain100H dataset, the RLDNet and RGDNet both gain over RDNet more than 1.5dB.

Through quantitative and qualitative analysis discussed above, our proposed models including RDNet, RGDNet and RLDNet effectively remove the rain streaks with various directions while promising more natural and realistic luminance. Even in heavy rain, out model still shows a good performance, which significantly improves the subjective effects and greatly surpasses other methods in terms of clarity and visibility.

As shown in Table 3, our proposed method can also achieve good results in removing light rain. Especially on the Rain1000 dataset, the average PSNR is 3.52dB higher than the SRDN. The Rain1000 dataset contains 14 different rainfall with different rain streak directions, which poses a huge challenge to the robustness of the model, which also shows that our proposed method is more robust than existing methods [5], [7], [8], [38], [23], [45]. Compared to Rain12 and Rain100L, they only have one type of rain pattern. The method [8], [38] has good rain removal performance, but our method performs better in single image rain removal. As shown in Fig. 11 and Fig. 12, in contrast to above six methods, our proposed RLDNet is more capable of removing rain streaks while preserving image details.

D. RESULTS ON REAL-WORLD DATASET FOR SINGLE IMAGE DERAINING

Since the real-world rainy image may contains the rain streaks with different scales, the degradations of the real-world rainy image are complex. In this section, we test the superiority of our RLDNet, which is trained on the synthetic rainy images but still performs well on the real-world rainy images.

Fig. 6 list three examples for demonstrating the performances of different methods on the real-world rainy image. As observed, our RLDNet obtains consistently promising deraining performances on the real-world images with different scales of rain streaks. It shows better visual effects than all the competitors on the three examples, especially in preserving the local structures and edges of the images.

TABLE 3. Quantitative experiments evaluated on three synthetic datasets (Light rain). Best results are marked in bold.

Rain1000		
Measure	PSNR	SSIM
Rainy images	21.34	0.7046
LP[5]	23.75	0.7844
Method[23]	21.97	0.8333
Method[45]	24.84	0.7935
DetailsNet[7]	27.31	0.8703
JORDER-R[8]	26.59	0.8605
SRDN[38]	27.72	0.8819
RLDNet	**31.24**	**0.9226**
Rain100L		
Measure	PSNR	SSIM
Rainy images	25.52	0.8255
Method[45]	27.14	0.8503
LP[5]	28.36	0.8712
Method[23]	28.17	0.9134
DetailsNet [7]	31.39	0.9161
JORDER-R [8]	35.21	0.9696
DID-MDN [12]	30.22	0.8259
SRDN[38]	37.28	0.9704
RLDNet	**37.82**	**0.9788**
Rain12		
Measure	PSNR	SSIM
Rainy images	28.82	0.8371
Method[45]	31.93	0.9083
LP[5]	30.70	0.8928
Method[23]	29.42	0.9033
DetailsNet [7]	30.68	0.8947
JORDER-R [8]	34.49	0.9447
DID-MDN [12]	28.90	0.8750
SRDN [38]	34.41	0.9470
RLDNet	**35.67**	**0.9558**

E. RESULTS ON IMAGE DENOISING

To further demonstrate the capability of our proposed method for handling similar problems, we compare the proposed RLDNet with several state-of-the-art denoising methods, including two non-local similarity based methods (i.e., BM3D [33] and WNNM [36]), two discriminative training based methods (i.e., MLP [37] and TNRD [29]). Note that TNRD is highly efficient by GPU implementation while offering good image quality. Two methods are based on deep learning (i.e., DnCNN [35] and FFDNet [34]), and DnCNN [35] is one of the most representative deep learning network models for image denoising with Gaussian noise.

The average PSNR results of different methods on the BSD68 and Set12 dataset are shown in Table 4. Our method is best for qualitative assessment of PSNR at three noise levels. Specifically, our model gains an improvement of 0.02dB by comparison of DnCNN [35] both on the Set12 and BSD68 datasets with $\sigma = 15$. Note that our model is trained for only 45 epochs. When the noise level is $\sigma = 25$ and the trained epochs are 80, the PSNR values obtained by the RLDNet are increased by 0.07dB and 0.06dB on the Set12 and BSD68, respectively. When the noise level is $\sigma = 50$ and the trained epochs are still 80, the PSNR gains increase by 0.15dB and 0.09dB on the Set12 and BSD68, respectively.

FIGURE 6. Visual quality comparison on the real-world rainy image.

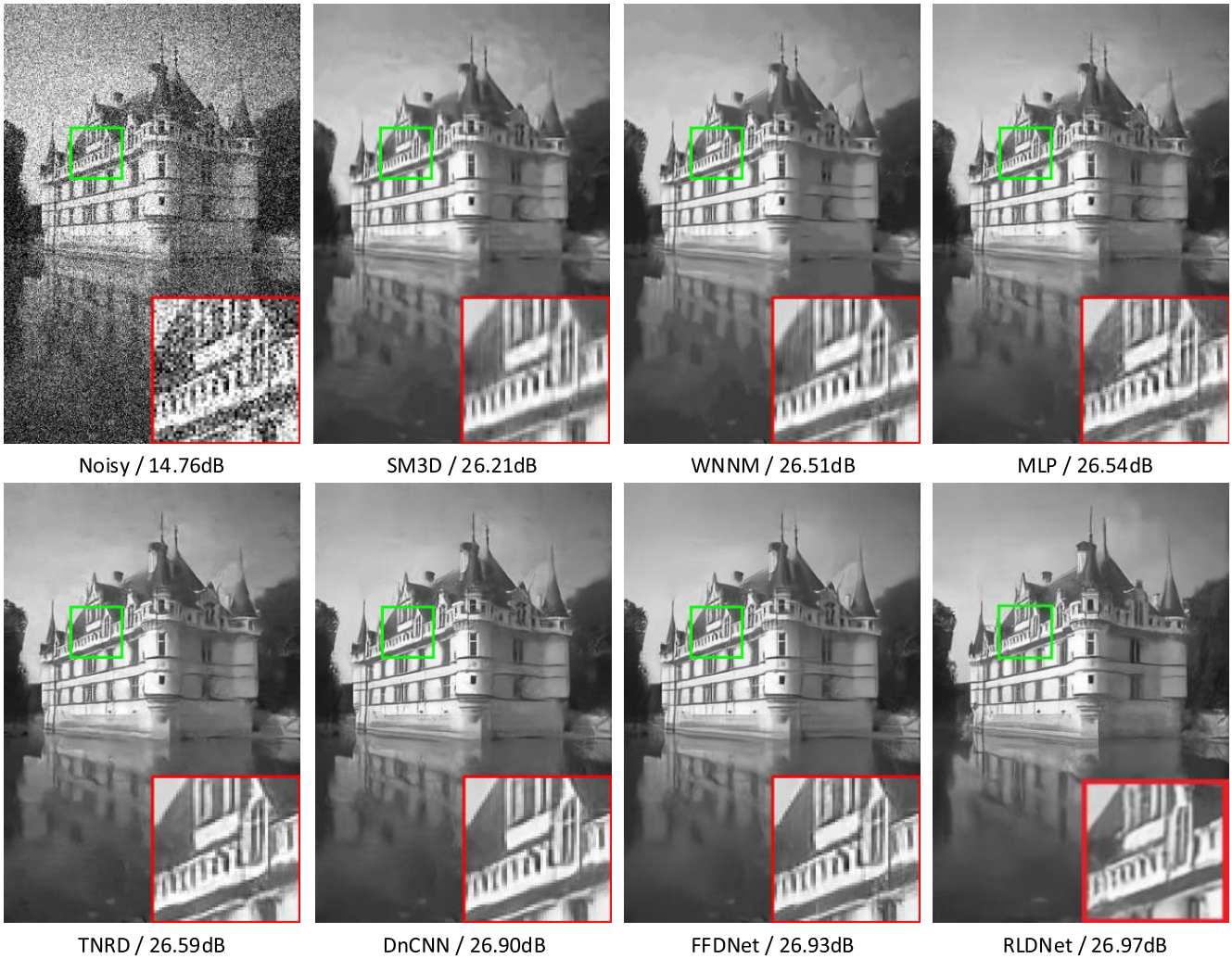


FIGURE 7. Denoising results by different methods on one image from BSD68 with noise level 50.

As discussed above, the PNSR gains by the proposed RLD-Net are become bigger with the increase of the noise levels. Note that the PNSR results obtained by DnCNN are all trained for 180 epochs. If we continue to train our model, the performance becomes better. More exploring experiments are left for further study.

FIGURE 8. RLDNet removes rain streaks by stages.

TABLE 4. Average PSNR values obtained by different methods under noise level 15, 25 and 50 on Set12 and BSD68. The best results are highlighted in red.

Dataset	σ	BM3D[33]	WNNM[36]	MLP[37]	TNRD[29]	DnCNN[35]	FFDNet[34]	Our(RLDNet)
Set12	15	32.37	32.70	-	32.50	32.86	32.75	32.88
	25	29.97	30.28	30.02	30.05	30.44	30.43	30.51
	50	26.72	27.05	26.78	26.82	27.18	27.32	27.33
BSD68	15	31.08	31.37	-	31.42	31.73	31.63	31.75
	25	28.57	28.83	28.96	28.92	29.23	29.19	29.29
	50	25.60	28.87	26.03	25.97	26.23	26.29	26.32

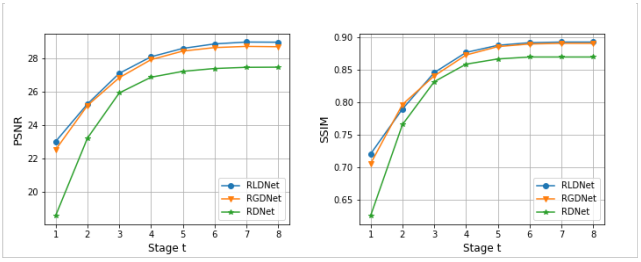


FIGURE 9. Average PSNR and SSIM values of RLDNet, RGDNet and RDNet at stage t = 1, 2, 3, 4, 5,6,7,8.

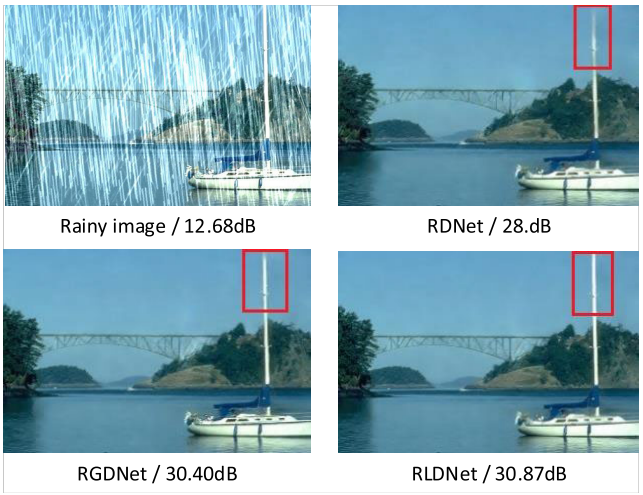


FIGURE 10. The visual results obtained by RDNet, RLDNet and RGDNet.

Fig. 7. shows the denoising visual results obtained by different denoising methods at a noise level of $\sigma = 50$. Obviously, RLDNet produce the best PNSR value among the competing denoising methods. As observed in Fig. 7, BM3D [33], WNNM [36], MLP [37] and TNRD [29] generate obvious visual artifacts on the sky area. Although DnCNN and FFDNet achieve the visual results almost comparable to our proposed model. By analyzing the visual quality in zoom area, we still can see that the local structures and the edges obtained by our proposed model are more natural and smooth, in agreement with the visual results perceived by a human observer.

VI. ANALYSIS ON OUR PROPOSED MODEL

According to the experimental analysis of the network model in [10], it has been proved that the model with iterations can improve the deraining performance from single image stage by stage. In the following, we also validate this improvement again by experiments. As can be seen in Figure 8, the RLDNet shows a poor visual effect of rain streaks removal when t = 1 (that is LDNet). The visual effect obviously gets better when t = 3, but there exists clear residues of the rain streaks in the image. The visual results obtained under t = 5 and t = 7 are quite similar except for the details in local structures and edges, which can be seen clearly by zoom. The PNSR values are also compared under different number of iterations. They are 23.53dB for t = 1, 27.17dB for t = 3, 29.99dB for t = 5, and 30.89dB for t = 7. The margin between t = 1 and t = 7 is 7.36dB. This fully illustrates the contributions introduced by recurrent to the performance improvement of our proposed model.

As shown in Table 2 and Table 5, the PNSR and SSIM values of our proposed RLDNet are better than the results of RESCAN. Note that our model only iterates 3 times to generate the results on the Rain100H. when our model cycles 4 times, the performance is better than S-DSEN. The backbone of S-DSEN also adopts the feature reuse scheme, and exceeds the other competing methods for rain removal. By combining the advantages of S-DSEN and RESCAN, we design the backbone of our proposed model: RLDNet with 5-layer dense blocks shown in Fig. 3. Both on Rain800 and Rain100H datasets, the performances of RLDNet are higher than RESCAN and S-DSEN demonstrated in Table 2. As can be seen in Table 5, the results of RDNet on Rain800 dataset

FIGURE 11. Deraining results of different methods on Rain1000. (a) Rainy images (b) Result by the method [5] (c) Result by the method [23] (d) Result by the method [7] (e) Result by the method [12] (f) Result by the method [45] (g) Result by the method [38] (h) Result by our proposed RLDNet (i) Rainy images.

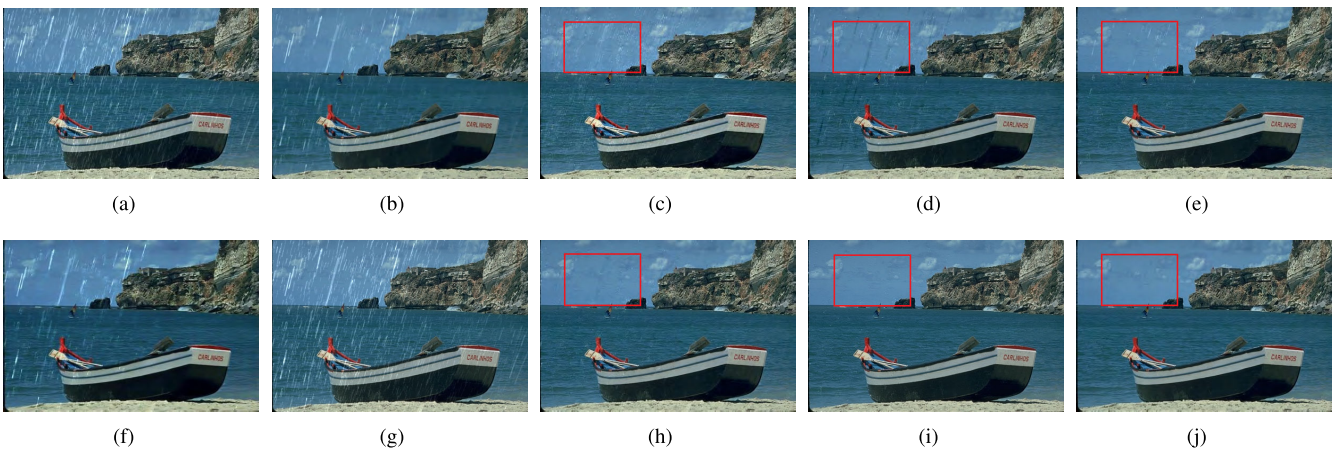


FIGURE 12. Deraining results of different methods on Rain12. (a) Rainy images (b) Result by the method [5] (c) Result by the method [23] (d) Result by the method [7] (e) Result by the method [8] (f) Result by the method [12] (g) Result by the method [45] (h) Result by the method [38] (i) Result by our proposed RLDNet (j) Rainy images.

are superior to the results achieved by RESCAN and S-DSEN in terms of both PSNR and SSIM when stage t = 2. When stage t = 5, the results of our proposed RLDNet on Rain100H dataset is higher than RESCAN and S-DSEN.

In order to improve the performance of the RDNet network, we also add state units to RDNet according to the state unit of RNN. We added the LSTM module to the RDNet to form RLDNet. GRU is a variant of LSTM that exploits a single update gate by combining the forgotten gate and the input gate. Meanwhile the cell state and the hidden state are also grouped together. With some other changes, the final model is simpler and more popular than the standard LSTM. We add the GRU network module to the RDNet to form RGDNet. We verified the rain-removal effects of RLDNet and RGDNet on the Rain100H and Rain800, as shown in Table 5. When the stage t = 7, RLDNet and RGDNet respectively achieved the best results of the model, but RLDNet performed better than RGDNet. On the Rain100H test set, the average PSNR was

TABLE 5. The results of RLDNet, RGDNet and RDNet on Rain800 and Rain100H datasets.

RDNet									
Stage(t)		1	2	3	4	5	6	7	8
Rain100H	PSNR	18.59	23.22	25.92	26.86	27.21	27.38	27.45	27.46
	SSIM	0.625	0.765	0.831	0.858	0.866	0.869	0.869	0.869
Rain800	PSNR	23.17	24.23	24.88	25.21	25.34	25.36	25.32	25.25
	SSIM	0.745	0.799	0.828	0.842	0.847	0.850	0.850	0.850
RGDNet									
Stage(t)		1	2	3	4	5	6	7	8
Rain100H	PSNR	22.53	25.17	26.83	27.91	28.42	28.63	28.70	28.68
	SSIM	0.705	0.796	0.840	0.872	0.885	0.889	0.890	0.890
Rain800	PSNR	23.43	24.26	24.86	25.29	25.59	25.74	25.67	25.38
	SSIM	0.75	0.798	0.824	0.840	0.849	0.853	0.852	0.848
RLDNet									
Stage(t)		1	2	3	4	5	6	7	8
Rain100H	PSNR	23.02	25.26	27.09	28.08	28.58	28.85	28.96	28.95
	SSIM	0.720	0.789	0.845	0.876	0.887	0.891	0.892	0.892
Rain800	PSNR	23.27	23.88	24.21	24.95	25.82	26.38	26.47	26.24
	SSIM	0.700	0.7723	0.804	0.820	0.836	0.841	0.847	0.845

0.26dB higher, indicating that the RDNet added LSTM model is more effective in rain removal than the added GRU model.

Fig. (9) and (10) demonstrates the visual results obtained by the three proposed models: RDNet, RLDNet and RGDNet. The performance of RDNet was significantly improved after adding the memory unit. However, As observed in the red boxes in images, RDNet performs poorly at edges compared with the other two methods. RLDNet and RGDNet gain almost the same visual effects except for a slight difference at edges. The detail preservation of RLDNet is better than RDGNet.

VII. CONCLUSION

In this paper, we propose a double recurrent dense network for single image deraining. Our double recurrent scheme has two information flow, one for rain streaks and the other for the background. By using the double recurrent framework and the hybrid loss function, our model provides the capacity to remove the rain streaks with various directions and intensities. Extensive experiments validate that our proposed RLDNet is better than the state-of-the-art methods in the task of removing rain, and it is also better than the most classic denoise methods in removing Gaussian noise.

REFERENCES

[1] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[2] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1968–1975.

[3] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. and Wang, "Context-aware single image rain removal," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 164–169.

[4] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE ICCV*, Dec. 2015, pp. 3397–3405.

[5] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2736–2744.

[6] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1717–1725.

[7] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE CVPR*, Jul. 2017, pp. 1715–1723.

[8] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE CVPR*, Jul. 2017, pp. 1357–1366.

[9] J. Pan, S. Liu, D. Sun, J. Zhang, Y. Liu, J. Ren, Z. Li, J. Tang, H. Lu, and Y.-W. Tai, "Learning dual convolutional neural networks for low-level vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3070–3079.

[10] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 262–277.

[11] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2482–2491.

[12] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2018, pp. 695–704.

[13] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

[14] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[15] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078

[16] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[18] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, nos. 2–3, pp. 256–274, Jan. 2010.

[19] J. Bossu, N. Hautière, and J. P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, 2011.

[20] K. Garg and K. Shree Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2004, p. 1.

[21] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.

[22] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *Int. J. Comput. Vis.*, vol. 112, no. 1, pp. 71–89, 2015.

[23] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Apr. 2017.
[24] H. Liu, H. Ye, X. Li, W. Shi, M. Liu, and Q. Sun, "Self-refining deep symmetry enhanced network for rain removal," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2018, pp. 2786–2790.
[25] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," 2017, *arXiv:1701.05957*. [Online]. Available: http://arxiv.org/abs/1701.05957
[26] H. Kaushal, V. Jain, and S. Kar, "Free-space optical channel models," in *Free Space Optical Communication*. New Delhi, India: Springer, 2017, pp. 41–89.
[27] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *IET Electron. Lett.*, vol. 44, no. 13, pp. 800–801, 2008.
[28] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 773–782.
[29] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2016.
[30] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.
[31] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, vol. 30, 2013, p. 3.
[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Jul. 2007.
[34] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.
[35] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Feb. 2017.
[36] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.
[37] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.
[38] H. Xia, R. Zhuge, H. Li, S. Song, F. Jiang, and M. Xu, "Single image rain removal via a simplified residual dense network," *IEEE Access*, vol. 6, pp. 66522–66535, 2018.
[39] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2019.2895793.
[40] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2019.2920407.
[41] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2019, doi: 10.1109/tnnls.2019.2926481.
[42] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proc. CVPR*, 2019, pp. 3877–3886.
[43] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4057–4066.
[44] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "FastDeRain: A novel video rain streak removal method using directional gradient priors," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 2089–2102, Apr. 2019.
[45] L.-J. Deng, T.-Z. Huang, X.-L. Zhao, and T.-X. Jiang, "A directional global sparse model for single image rain removal," *Appl. Math. Model.*, vol. 59, pp. 662–679, Jul. 2018.
[46] K. Garg and S. K. Nayar, "Photorealistic rendering of rain streaks," *TOGACM Trans. Graph.*, vol. 25, no. 3, p. 996, Jul. 2006.
[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

YANG LAN is currently pursuing the master's degree with the School of Electronic Engineering, Guangxi Normal University. His research interests include image restoration, medical image segmentation, and deep learning.

HAIYING XIA received the M.S. and Ph.D. degrees from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2011, respectively. She worked as an Associate Professor with Guangxi Normal University. Her current research interests include pattern recognition, medical image analysis, and neural networks.

HAISHENG LI received the M.S. degree in computer science from Chongqing University, Chongqing, China, in 2004, and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China. He is currently an Associate Professor with the College of Electronic Engineering, Guangxi Normal University, China. His research interests include quantum information processing, quantum neural networks, and image processing.

SHUXIANG SONG received the Ph.D. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China. He is currently a Full Professor with Guangxi Normal University. His current research interests include intelligent detection, automatic control, and signal and image processing.

LINGYU WU is currently pursuing the master's degree with the School of Electronic Engineering, Guangxi Normal University. Her research interests include image registration, medical image segmentation, and deep learning.