Lightweight single image deraining algorithm incorporating visual saliency

mingdi hu (✉ humingdi2005@xupt.edu.cn)

Xi'an University of Posts and Telecommunications <https://orcid.org/0000-0002-5568-5321>

Jingbing Yang

Xi'an University of Posts and Telecommunications

Research Article

Keywords: Single Image Deraining, Visual Saliency, Lightweight Networks, Deep Learning

Posted Date: March 7th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1375267/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

RESEARCH

Lightweight single image deraining algorithm incorporating visual saliency

Mingdi Hu* and Jingbing Yang

*Correspondence:
mendy2013@163.com
School of Communication and
Information Engineering School of
Artificial Intelligence, Xi'an
University of Posts and
Telecommunications, Xi'an, China
Full list of author information is
available at the end of the article

Abstract

Deep learning (DL) methods have achieved excellent performance in the task of single image rain removal, however, there are still some challenges, such as artifact remnant, background over-smooth, and more and more complex and heavy-weight network architecture. Due to too heavy-weight network to suit outdoor detection devices or mobile devices, therefore, we propose a light-weight single image deraining algorithm incorporating visual attention saliency mechanisms (LDVS). The network consists of dilation convolution module, the convolutional block attention module (CBAM), and gated recursive unit module. Specifically, rain streaks feature maps are extracted by the combinations of dilated convolution with CBAM, which also facilitates accurate localisation of rain streak location, and then the three gated recursive units is cascaded to remove streaks stage by stage. The dilated convolution module and CBAM are used to reduce network's weight size and retain the rain removal result, thus our LDVS method belongs to the lightweight with only 50703 parameters. Extensive experiments on synthetic and real-world datasets have demonstrated that our method outperforms the baseline both under qualitative and quantitative analysis. Under the same rain removal result, our method is less time cost and less burden.

Keywords: Single Image Deraining; Visual Saliency; Lightweight Networks; Deep Learning

Introduction

In recent years, with the improvement of computing resources, machine vision methods have been widely applied in real world, such as industry, military, etc., including face recognition, vehicle detection and drone reconnaissance [1]. However, in bad weather, especially rainy, the original images captured by the camera are usually severely degraded by rain streaks, raindrops, rain and fog, which reduces the accuracy of recognition and detection of subsequent high-level vision tasks. So deraining is indispensable pre-processing process.

Figure 1 Example of rain removal results using our proposed rain removal method on synthetic Rain800 datasets.

Deraining includes video rain removal and single image rain removal. Single image deraining methods is more challenging because of only exploiting the spatial information of neighboring pixels and the visual properties of rain and background scenes [2]. Removing rain streaks from single rainy images is paid more and more attention in the computer vision community, and deep learning (DL) methods have

achieved excellent performance in the task of single image rain removal, however, there are still some challenges, such as artifact remnant, background over-smooth, and more and more complex and heavy-weight network architecture. Due to too heavy-weight network to suit outdoor detection devices or mobile devices, so we proposed the lightweight rain removal method incorporating visual saliency in this paper. The result of single image rain removal through our method is shown in Figure 1, and the method is also abbreviated as LDVS in the next.

The mainstream single image rain removal methods are generally fall into model-driven traditional methods and data-driven deep learning-based methods [3]. The former focus on sufficiently encoding the physical properties and prior information of rain streaks and background images into an optimization model and designing rational algorithms to recover the clean background image [4, 5]. The later learn network parameters for attaining complex rain removal functions by designing specific network architectures and precollecting rainy-clean image pairs [6, 7, 8, 9, 10, 11]. Model-based Single image de-raining methods rely more on the statistical analysis of rain streaks and background scenes, but these methods generally need time-consuming iterative computations, often with efficiency issue in real applications [12, 13, 14, 15, 16, 17, 18]. Recently based on deep learning single image deraining algorithms with superior performance have occupied the mainstream in the field. For example, Ding et al. constructed a circular iterative distributed feedback network [7], which transmits the output results to the next iteration through a feedback connection to gradually recover the clean background Image; Li et al. proposed a 2-stage network, the first stage is a physics-based backbone which estimates the rain streaks, the transmission, and the atmospheric light, and the second stage is the refinement stage which use a depth guided Generative Adversarial Networks (GAN) to recover the background details failed to be retrieved by the first stage [8]. Wang et al. constructed a deraining network combining self-attention module, scale aggregation module, with self-calibrating convolution module, called JDNet [10]. Specially, JDNet used dense connections in order to increase feature reuse. As we all known that most of the single image deraining methods based deep neural network advocate to design a complicative network by basic modules such as Multi-stage, Attention mechanism, Dense connections, Encoder-Decoder etc.. Although these methods have achieved excellent performance, they consume too time as pre-processing procedure of subsequent high-level tasks, and they are difficult to embed into the mobile devices because of the large number of parameters.

To address the limitations mentioned above, a lightweight single-image deraining algorithm incorporating visual saliency mechanisms (abbreviated as LDVS.) is proposed to improve the detection speed of the model and compress the number of parameters of the model while ensuring the rain removal performance. The main contributions of this paper can be concluded as follows:

- In order to obtain the contextual information of the input rainy image and accurately remove the rain steaks, dilated convolution [19] and CBAM [20] are leveraged to construct the backbone network of the LDVS, which can gradually increase the perceptual field and achieve extracting location information from local to global stage by stage.

- To increase the possibility of embedding into existing hardware devices, the lightweight CBAM (Convolutional Block Attention Module) is selected to guide rain streaks removal in both channel and spatial dimensions. The parameter costs of the CBAM is small according to the literature [20].
- Considering rain streaks have different directions and sizes in real rain scenes, we adapt gated recursive units (GRU) module to achieve recurrent deraining [21].
- Extensive experiments on synthetic and real-world datasets demonstrate that our method outperforms the baseline qualitatively and quantitatively. Under the same rain removal effect, our method is less time cost and less burden.

The paper is organized as follows. Sections 2, review the related works. Section 3, introduce the architecture of the proposed LVDS, and the main role of each module. Section 4, a comprehensive experiment is implemented. Conclusion is presented in section 5.

Related work

In this section, we review the current representative single-image rain removal methods and some lightweight deep neural network methods in the field of computer vision.

Single image rain removal Methods

Since Kang et al. was a pioneer in the single image deraining, and proposed image decomposition method for single image deraining [16]. Luo et al. proposed discriminative sparse coding for rain streaks removal by exploiting the sparsity of the rain streaks [17]. Recently, researchers have successively built deeper and more complex network structures to chase better performance of single-image rain removal methods. Ren et al. [22] proposed a progressive recurrent network, called PReNet. Using convolution and residual blocks was shown to be more effective in image deraining. Ni et al. [23] proposed the rain removal and generation cycle network, which consists of three main sub-networks: the Background Extraction Network (BEN), the High-frequency Rain-streak Elimination Network (HREN), and the Main Controlling Network (MCN). The BEN module extracts the background information. The HREN module removes the high frequency rain streaks, and extracts background and fog information. The MCN module is used to bi-directionally control the generation of rain of different intensities. To make the rain removal network both interpretable and flexible, many researchers have explored to combine traditional model-driven based on prior knowledge with data-driven model based on deep neural network. E.g., Zhu et al. [24] constructed a physical model-guided rain removal network consisting of rain streaks network, rain-free network and guided learning network to remove the rain streak. Wang et al. [25] proposed an interpretable single image rain removal deep neural network based on the convolutional dictionary learning mechanism, called rain convolutional dictionary (RCD) network. Yang et al. [26] proposed a context-dilated network using dilated convolution to extract multi-scale contextual information of the input rainy image, and rain streak detection module and rain streak estimation module are added to implement multi-task rain removal. Subsequently, Deng et al. [27] proposed a two-branch rain removal network, by combining

the dilation convolution with the Squeeze and excitation (SE) attention module. That is to say, the SE attention module is leveraged to remove rain streaks in the rain removal residual network branch, and the parallel dilation convolution module is leveraged to recover image details in the detail restoration network branch. Jiang et al. [28] introduced a rain removal network combined with multi-scale attention mechanism, which allows more image contextual information to interact with each other. The dilated convolution module is frequently leveraged to construct deep neural network, because more effectively contextual feature maps and contextual location relationship are extracted through increasing the perceptual field of the input rainy image.

Light-weight deep neural networks

On the one hand, subsequent high-level tasks require the assistance of lower-level tasks. On the other hand, due to the limited capacity, mobile devices suffer from to meet the storage and running space requirements of complex network modules. Thus light-weight network-based methods for low level task are focused. For example, Yang et al. [29] propose a lightweight adaptive feature fusion network (LAFFNet) to address underwater image enhancement. Adaptive feature fusion module subsumes multiple branches with different kernel sizes to generate multi-scale feature maps, channel attention is used to merge these feature maps adaptively. The method reduces the number of parameters to 0.15M. Das et al. [30] proposed a lightweight deep multi-patch hierarchical network to restore Non-homogeneous hazed images by aggregating features from multiple image patches from different spatial sections of the hazed image with fewer number of network parameters. Zhang et al. [31] proposed lightweight multi-scale end-to-end dehazing network called FAMED-Net, which comprises encoders at three scales and a fusion module to efficiently and directly learn the haze-free image. Although deep learning based single image rain removal methods have made great progress in rain streak removal and background texture maintenance, these rain removal networks are more complex in structure with more parameters and run time-cost too much. Yamamichi et al. [32] proposed a novel multi-scale context aggregation network, which exploits a lightweight residual structure subnet as the baseline architecture, to effectively solve the single image deraining problem. Fu et al. [33] proposed a lightweight rain removal network, called LPNet, which uses recursive and residual networks to construct a lightweight Gauss-Laplace pyramid rain removal network to extract multi-scale features of different rainy image. Although the method uses a low number of parameters and is easy to embed in mobile devices, its rain removal effect is poor.

Inspired by the references [34] and [20], we proposed a lightweight recurrent rain removal network (LDVS) that combines dilated convolution module with attention module not only to address above mentioned too-heavy neural network frame but also retain rain removal result. Our network belongs to light-weight because of only 50703 parameters, and our method outperforms the baseline both under qualitative and quantitative analysis. Under the same rain removal effect, our method is less time cost and less burden.

Figure 2 Rain model.

Methodology

Framework Overview

Rain in rainy image generally is considered as two forms: raindrops and rain steaks. Rain steaks are the forms presented in the rainy image when raindrops fall sharply due to the different volumes, velocities, and air resistance of raindrops. To simplify the calculation, the observed rainy image is generally considered as a direct superposition of the rain-free background image and the rain steaks, called the RSM rain model [35], which can be formulated as:

$$O = B + R. \quad (1)$$

Where O denotes the rainy image, B denotes the clean background image, and R denotes the rain steaks, as shown in Figure 2, The purpose of image rain removal is to recover a clean background image from the input rainy image.

Figure 3 Lightweight single image de-raining network framework incorporating visual saliency (LDVS). The network mainly consists of dilated convolution module, a CBAM and GRU module.

The overview of our proposed LDVS is shown in Figure 3, which comprises of backbone block and refine block. There are an encoder with five feature extracting modules and a decoder in the backbone block, and three GRU consisting of refine block, therein, each feature extracting module cascade a dilation convolution [19] with a CBAM [20]. The rainy image O is fed into the backbone to be extracted the feature maps, the output result of the first stage is rain steaks mask R' , The loss function $L_1 = \|R' - R\|_F^2$ for this stage. Rain-free image B_1 is equal to the input rain image O minus the feature map R' through the backbone block. Then, rain-free image B_1 is re-entered into the backbone network through the refine block to extract a cleaner rain-free background image B_2 , and then repeat the above steps, so the cycle rain removal begins. We chose three cycles as a trade-off for the ablation experiment, we obtained the total loss functions to $L = L_1 + L_2 + L_3$ train our network.

Basic module

Dilated convolution

Figure 4 Receptive fields under different expansion coefficients.

The traditional convolution operation could lose image information. In this paper, we use dilated convolution [19] to both progressively increase the perceptual field and obtain multi-scale contextual information without increasing the number of parameters. The perceptual field of the dilation convolution with different dilated rate shown in Figure 4.

Figure 5 The basic structure of CBAM.

CBAM

Depending on the different forms of attentional mechanisms that give weight to the network, attentional mechanisms are mainly divided into spatial, channel and hybrid domains. We use the CBAM of the hybrid domain in this paper [20]. The network architecture of this attention module is shown in Figure 5. The CBAM mainly consists of channel attention and spatial attention, where channel attention is composed of max-pool and avg-pool, multilayer perceptron (MLP), and spatial attention is composed of max-pool and avg-pool, convolution. The input feature mapping is $F \in R^{C \times H \times W}$, the one-dimensional channel attention mapping $M_c \in R^{C \times 1 \times 1}$ and the two-dimensional spatial attention mapping $M_s \in R^{1 \times H \times W}$.

Since each channel of the feature map is considered as a feature detector, and channel attention focuses on finding what is meaningful in the input rainy image. The specific process is as follows: firstly, the obtained max-pool and avg-pool pixel values are fed into a shared network consisting of a multi-layer perceptron and a hidden layer. Then, the features are fused and the sigmoid function is used to obtain the channel attention map. The spatial attention is concerned with where information is important, and is complementary to channel attention. Spatial attention mainly generates a weight mask at each pixel position and weights the output. The specific process is as follows: firstly, avg-pool and max-pool operations are performed on the input feature along the channel axis. secondly, they are stitched together by channel dimension to obtain a feature map with a channel number of 2. Finally, the spatial attention features are obtained by using a hidden layer and sigmoid function. The attention mechanism is mainly used to guide the model to focus on the key information and key location of the input image, it could extract effective feature information and improve the network performance. In simple terms, attention mechanism is to highlight important information and suppress unnecessary information. Therefore, the combined use of the CBAM, which combines channel attention and spatial attention, and dilation convolution could facilitate the network to extract more valid information and accurately remove rain steaks.

Gated Recurrent Unit

Figure 6 The basic structure of the GRU.

The internal structure of the GRU is shown in Figure 6 [21]. Firstly, two gate states (reset gate, update gate) are obtained by the state transmitted down from the previous node h_{t-1} and the input of the current node x_t , where update gate indicates whether the new information is updated to the hidden layer state, reset gate indicates whether the previous state information is forgotten. Next, the candidate hidden states are calculated. Finally, by controlling certain dimensional information transmitted down from the previous node and adding the default dimensional information entered by the current node, so the network is updated in the way shown

in Equation 1.

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}. \quad (2)$$

In equation (1), Z_t denotes the update gate, $\odot$ denotes the same or operation, $\tilde{h}_t$ denotes the hidden state calculated by the reset gate, and h_{t-1} denotes the hidden state before the moment of t. The recursive neural network (RNN) [34] usually takes the output of the previous stage as the input of the later stage without considering the feature relationship between different stages, and the RNN has the disadvantage of gradient disappearance and gradient explosion. Compared to other recurrent neural networks, the GRU has only one update gate, so it has a simple structure with fewer parameters, and it is able to achieve comparable functionality to the long short-term memory (LSTM) [36]. Therefore, in our lightweight rain removal network, the GRU is selected for multi-stage removal of rain streaks considering the computational power and time cost of hardware. In order to better use the information from the previous stage to guide the feature learning of the later stage, we use the output features of the previous part of the current layer and the features of the corresponding part of the previous stage to jointly calculate the current features.

The LDVS

Formulation of the LDVS

In stage1 of the LDVS:

The encoder part: Input rainy image x . In the first part of the first stage, the calculation process is as follows:

Firstly, the feature map extracted from the input rain map by dilated convolution is represented as:

$$g'_1 = f_{\text{sigmoid}}(f_{\text{cnn}}(x)) \times f_{\text{tanh}}(f_{\text{cnn}}(x)). \quad (3)$$

Then, the channel attention and spatial attention of CBAM are calculated as:

$$f_{\text{channel}} = f_{\text{sigmoid}}(\text{MLP}(\text{Avgpool}(g'_1)) + \text{MLP}(\text{Maxpool}(g'_1))). \quad (4)$$

$$f_{\text{spatial}} = f_{\text{sigmoid}}(f_{\text{cnn}}([\text{Avgpool}(f_{\text{channel}}); \text{Maxpool}(f_{\text{channel}})])). \quad (5)$$

Finally, the output feature g_1 of the first part is represented as:

$$g_1 = f_{\text{Leaky ReLU}}(f_{\text{spatial}}). \quad (6)$$

Where, $f_{\text{cnn}}(\cdot)$ denotes the convolution which kernel size is 3*3, f_{channel} represents the channel attention, f_{spatial} represents the spatial attention, g_i represents the output feature of each part of the first stage.

And so on, in subsequent parts of the first stage, the combination function of dilated convolution (DF=1,2,4,8) and CBAM are denoted as: $f_{d1\text{cnn-cbam}}(\cdot)$, $f_{d2\text{cnn-cbam}}(\cdot)$, $f_{d4\text{cnn-cbam}}(\cdot)$, $f_{d8\text{cnn-cbam}}(\cdot)$. Therefore, the output feature of the

second part is $g_2 = f_{d1cnn-cbam}(g_1)$; The output feature of the third part is $g_3 = f_{d2cnn-cbam}(g_2)$; The output feature of the fourth part is $g_4 = f_{d4cnn-cbam}(g_3)$; The output feature of the fifth part is $g_5 = f_{d8cnn-cbam}(g_4)$. g_5 denotes the final output of the encoder.

The decoder part: Use the output features of the encoder as input to the decoder, the calculation process is as follows:

Firstly, the input features g_5 are subjected to convolution operations and CBAM attention, which can be formulated as:

$$f_{channel} = f_{\text{sigmoid}}(\text{MLP}(\text{Avgpool}(f_{cnn}(g_5))) + \text{MLP}(\text{Maxpool}(f_{cnn}(g_5)))) \cdot (7)$$

$$f_{spatial} = f_{\text{sigmoid}}(f_{cnn}([\text{Avgpool}(f_{channel}); \text{Maxpool}(f_{channel})])). \quad (8)$$

Then, after the convolution which kernel size is 1×1 and activation function operation, the first stage of the rain streaks feature is represented as

$$F_1 = f_{1 \times 1cnn}(f_{\text{LeakyRelu}}(f_{spatial})). \quad (9)$$

where, $f_{cnn}(g_5)$ denotes the result of convolution which kernel size is 3×3 , $f_{channel}$ represents the channel attention, $f_{spatial}$ represents the spatial attention, $f_{1 \times 1cnn}$ denotes the convolution which kernel size is 1×1 , F_1 represents the rain streaks extracted in the first stage.

In stage2 of the LVDS:

The encoder part: Input rain-free image $B_1 = x - F_1$. In the first part of the second stage, we use the output features of the previous part of the current layer and the features of the corresponding part of the previous stage to jointly calculate the current features, the calculation process is as follows:

Firstly, the candidate hidden and hidden states are calculated by updating the gate and resetting the gate, which can be formulated as:

$$z_2^1 = f_{\text{sigmoid}}(f_{cnn}(B_1) + f_{cnn}(g_1)). \quad (10)$$

$$r_2^1 = f_{\text{sigmoid}}(f_{cnn}(B_1) + f_{cnn}(g_1)). \quad (11)$$

$$H_2^1 = f_{\text{tanh}}(f_{cnn}(B_1) + f_{cnn}(r_2^1 \times g_1)). \quad (12)$$

$$h_2^1 = (1 - z_2^1) \times g_1 + z_2^1 \times H_2^1. \quad (13)$$

Then, the channel attention and spatial attention of CBAM of the hidden state are calculated as:

$$f_{channel} = f_{\text{sigmoid}}(\text{MLP}(\text{Avgpool}(h_2^1)) + \text{MLP}(\text{Maxpool}(h_2^1))). \quad (14)$$

$$f_{spatial} = f_{\text{sigmoid}}(f_{cnn}([\text{Avgpool}(f_{channel}); \text{Maxpool}(f_{channel})])). \quad (15)$$

Finally, the output feature of the first part is represented as:

$$y_1 = (f_{\text{Leaky ReLU}}(f_{\text{spatial}})). \quad (16)$$

Where, $f_{\text{cnn}}(\cdot)$ denotes the convolution which kernel size is 3×3 , z_2^1 denotes the update gate, r_2^1 denotes the reset gate, H_2^1 denotes the candidate hidden states and h_2^1 denotes the hidden states, y_i represents the output feature of each part of the second stage.

And so on, in subsequent parts of the second stage, the combination function are denoted as: $y_{d1cnn-cbam}(\cdot)$, $y_{d2cnn-cbam}(\cdot)$, $y_{d4cnn-cbam}(\cdot)$, $y_{d8cnn-cbam}(\cdot)$. Therefore, the output feature of the second part is $y_2 = y_{d1cnn-cbam}(y_1)$; The output feature of the third part is $y_3 = y_{d2cnn-cbam}(y_2)$; The output feature of the fourth part is $y_4 = y_{d4cnn-cbam}(y_3)$; The output feature of the fifth part is $y_5 = y_{d8cnn-cbam}(y_4)$. y_5 denotes the final output of the encoder.

The decoder part: Use the output features y_5 of the encoder as input to the decoder, the calculation process is as follows:

Firstly, the input features y_5 are subjected to convolution operations and CBAM attention, which can be formulated as:

$$f_{\text{channel}} = f_{\text{sigmoid}}(\text{MLP}(\text{Avgpool}(f_{\text{cnn}}(y_5))) + \text{MLP}(\text{Maxpool}(f_{\text{cnn}}(y_5)))). \quad (17)$$

$$f_{\text{spatial}} = f_{\text{sigmoid}}(f_{\text{cnn}}([\text{Avgpool}(f_{\text{channel}}); \text{Maxpool}(f_{\text{channel}})])). \quad (18)$$

Then, after the convolution which kernel size is 1×1 and activation function operation, the second stage of the rain steaks feature is represented as:

$$F_2 = f_{1 \times 1 \text{cnn}}(f_{\text{Leaky ReLU}}(f_{\text{spatial}})). \quad (19)$$

Where, $f_{\text{cnn}}(y_5)$ denotes the result of the convolution which kernel size is 3×3 , f_{channel} represents the channel attention, f_{spatial} represents the spatial attention, $f_{1 \times 1 \text{cnn}}$ denotes the convolution which kernel size is 1×1 , F_2 represents the rain steaks extracted in the second stage. Here, we describe only a two-stage process.

Structural parameters of LDVS

The LDVS uses five GRU modules, each of which combines dilated convolution [19] and CBAM [20]. In the first GRU module [21], 3×3 convolution kernel and CBAM are adopted. In the second to fifth GRU module, CBAM and 3×3 convolution kernel with dilation rate is 1,2,4,8 respectively are adopted. This process is called encoder, and the number of parameters is 45018. In the decoder, the LDVS uses a 3×3 convolution kernel and CBAM. Finally, the rain steaks are extracted by activation function and 1×1 convolution kernel. The number of parameters in the decoding process is 5685.

Figure 7 Different convolution types. (a) shows the ordinary convolution, (b) shows the combination of dilated convolution and CBAM.

Dilated convolution combined with the CBAM

Since rain streaks have different orientations and sizes, in order to guide the rain removal network to pay attention to key location and key contents, each feature layer of rain streaks needs to be given different weights using the attention module. Therefore, many researchers have explored the possibility of using attention modules or utilizing more relational information to enhance the functionality of convolutional networks. We use CBAM as an enhancement to the convolutional module. The CBAM contains both global channel attention and local spatial attention, it can learn to the corresponding attention weights to adapt different rain feature layers. The CBAM assigns different weights to the different feature layers and re-weights them to obtain the new feature map. In our rain removal method, we combine the dilation convolution with CBAM in the rain removal network framework, which is shown in Figure 7 (c). Unlike the ordinary convolution operation, this combination not only applies weight to the rain image pixel-wise, but also applies weights to the rain map feature layers from the channel and spatial dimensions, which can remove the rain streaks in the image more effectively.

Loss function

To ensure the rain removal while recovering a clear and background texture, an objective function constrained network training is constructed for the real rain streak layer and the rain streak layer predicted by the network. In the proposed rain removal network, the mean squared error (MSE) [37] loss function is used to find the sum of the differences between the predicted rain layer and the real rain layer at each stage, the loss function shown in Equation 2.

$$L(\Theta) = \sum_{t=1}^T \left\| \hat{R}_t - R \right\|_F^2. \quad (20)$$

where, T denotes the number of recursive cycles, R denotes the real rain streaks layer, and $\hat{R}_t$ denotes the predicted rain streaks layer for each stage of extraction.

Experiment and discussion

In this section, we first introduction the implement Details, involving the datasets, training settings and evaluation metric. Then, we conduct a number of deraining experiments on synthetic datasets and real-world datasets compared with the state-of-the-art methods. Finally, we performed some ablation studies to verify the rationality and effectiveness of each module of our method.

Implement Details

Datasets

Since pairs of rainy images and clean background images cannot be collected in the real world. Therefore, four synthetic datasets are chosen for comparison. These synthetic datasets contain rain streaks of different sizes, directions, and shapes, among,

Rain1400 [38] contains rain images with 14 different steak orientations and magnitudes; Rain12 [15] contains 12 rain images with one type of rain streak; Rain800 [39] means that rain steaks are add into these clear images using Photoshop. Rain100L [40] has only one type of rain streak. These synthetic datasets can more accurately evaluate the performance of our proposed LDVS method. Also, we qualitatively analyze the effectiveness of our proposed rain removal method on the real-world dataset SPA-Data [41] collected by Wang et al.

Training settings

In the training process, the Adam optimizer is used to optimize the network model, the initial learning rate is 0.005. The nonsaturating activation function Leaky-ReLU is used for each convolutional layer to speed up the convergence. Randomly crop 100 patch pairs for each training rainy image pair, with each patch of size 64*64 and batch size is chosen to be 64. We adopt the pytorch implement. The experimental platform is set as follows: the operating system is Windows 10, CPU is Intel Core i7-8700K, the memory is 16GB, GPU is NVIDIA GTX 1070Ti, video memory is 8GB.

Evaluation Metric

Following the previous studies, peak signal to noise ratio (PSNR) and structure similarity (SSIM) are used to evaluate the quality of the recovered rain-free images in comparison with ground truth images. PSNR and SSIM are only computed for synthetic datasets, for the real-world datasets, they can only be evaluated by visual comparisons. PSNR value is calculated from MSE and maximum value of image colour, as shown in equation 4.

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (X(i, j) - Y(i, j))^2. \quad (21)$$

$$PSNR = 10 \times \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \sum_{j=1}^W (X(i, j) - Y(i, j))^2. \quad (22)$$

Where MSE represents the Mean Square Error of the current image X and the reference image Y , MAX_I represents the maximum value of the image colour, the 8-bit sampling point is 255. H and W are the height and width of the image respectively.

SSIM (structural similarity) is also a fully referenced image quality evaluation metric that measures image similarity in terms of brightness, contrast and structure, as shown in equation 5.

$$SSIM(X, Y) = L(X, Y) * C(X, Y) * S(X, Y) \quad (23)$$

Where $L(X, Y)$ represents brightness similarity between image X and image Y , $C(X, Y)$ represents contrast similarity between image X and image Y , and $S(X, Y)$ represents structure similarity between image X and image Y .

Ablation experiments

Ablation study on module

Table 1 Investigation of Convolution, Dilated Convolution, Dilated Convolution+CBAM. Performances are evaluated on Rain100L dataset.

Dataset	Metrics	Convolution	Dilated Convolution	Dilated Convolution+CBAM
Rain100L	PSNR	25.62	29.14	32.76
Rain100L	SSIM	0.751	0.901	0.965

We proposed feature extraction network consists of dilated convolution and CBAM. We conduct an ablation study to demonstrate the efficiency of different module. We use a single stage rain removal scheme to evaluate the rain removal performance of different modules. As shown in Table1. Simply convolution operations will obtain relatively low performances. Extraction of multi-scale rain map features using dilated convolution could improves the rain removal performance. We can further observe that the combined use of the Dilated Convolution [19] and CBAM [20] not only obtains contextual information about the input rainy image, but also accurately locates rain steaks areas, further improving de-rain performance.

Ablation study on stage

Table 2 The investigation of the number of stages. Performances are evaluated on Rain100L dataset.

Metrics	LDVS			
	Stage1	Stage2	Stage3	stage4
PSNR	32.76	32.83	32.87	32.88
SSIM	0.965	0.974	0.981	0.981
TIME	T	2T	3T	4T

To investigate the effect of the number of stages on the effectiveness of our proposed LDVS network for rain removal, our proposed LDVS with 1 to 4 stages were trained on the Rain100L dataset using the same strategy. As shown in Table 2. As the de-raining phase increases from 1 to 4, the de-raining effect increases slightly, but the training time increases exponentially, and T represents the training time of a single stage. Multi-stage can remove complex rain steaks situations and remove rain streaks efficiently by incorporating the correlations between neighboring stages. In order to reduce the risk of overfitting while balancing the quality of de-raining, maintaining background texture and improving operational efficiency, we remove rain steaks in 3 stages. The experimental results show that the number of stages of the network can be dynamically adjusted to train the network for rain removal when faced with different rain datasets.

Results on synthetic datasets

The proposed LVDS method is compared against the state-of-the-art methods qualitatively and quantitatively.

Qualitative analysis

Figure 8 Test results of different rain removal algorithms on the synthetic datasets.

Figure 8 shows the qualitative performance of different rain removal methods on the synthetic datasets (Enlarging the image will give better results). As can be

seen from Figure 8, the traditional model-driven rain removal method DSC [17] and the data-driven rain removal method CNN [42] have difficulty in distinguishing the rain streaks from the background, they cannot accurately remove the rain streaks to get a clear and clean background image. The DerainNet method [43], when facing the dense rain streaks scenes, cannot achieve the effect of complete rain streaks removal, the background is blurred after rain removal. The pyramid-based rain removal method LPNet does not maintain background detail well after rain removal, which produces artefacts and darker images [38]. The density-aware DID-MDN method, which removes rain streaks incompletely, can only partially remove the rain streaks [44]. In contrast, our proposed the LVDS method can remove more rain streaks in the rainy image and obtain better rain removal results.

Quantitative analysis

Table 3 Quantitative results of different methods on synthetic rain datasets.

Dataset	Metric	ID	DSC	CNN	DID-MDN	ID-CGAN	ours
Rain100L	PSNR	23.13	24.16	23.70	28.27	23.39	32.87
	SSIM	0.691	0.866	0.814	0.857	0.819	0.981
Rain800	PSNR	20.54	18.56	23.95	22.55	23.81	28.87
	SSIM	0.674	0.599	0.601	0.7639	0.807	0.976
Rain1400	PSNR	-	22.03	18.52	27.99	21.93	31.65
	SSIM	-	0.799	0.672	0.869	0.784	0.961
Rain12	PSNR	27.21	30.02	26.65	30.14	20.78	30.73
	SSIM	0.753	0.868	0.783	0.876	0.852	0.957

Table 3 shows quantitatively comparative results between our method and six state-of-the-art deraining methods on Rain100L [40], Rain800 [39], Rain1400 [38], Rain12 [15]. There are two conventional methods: ID [16] and DSC [17], and three deep learning-based methods: CNN [42], DID-MDN [44], ID-CGAN [39]. As can be seen from the Table 3, comparing different rain removal methods on each synthetic dataset, it can be seen that one of the model-driven rain removal methods, which relies on rain and background prior knowledge, performs much lower than the data-driven rain removal methods. Our proposed rain removal method outperforms the other rain removal methods on four datasets. Specifically, the proposed method obtains significant improvements by PSNR metric of 3.8 db, 4.92 db, 3.66 db, 0.59 db and SSIM metric of 0.5

Results on real-world datasets

Figure 9 Test results of different rain removal algorithms on the real-world dataset.

To further evaluate the generalization of our proposed LDVS method on the real-world dataset, we use the pre-trained model trained on Rain100L dataset to ensure the fairness of the evaluation results. we show the de-raining results of different rain removal methods on the real-world dataset SPA-data in Fig 9. The DID-MDN rain removal method [44] and the DualGCN rain removal method [45] do not completely remove the rain streaks. The LPNet rain removal method [33] blurs the background details after removing the rain streaks. For the first example, The LPNet method has a de-raining failure problem. Compared to other methods, our method removes almost all of the rain streaks while retaining better background detail.

Test time analysis

Table 4 The investigation of the number of stages. Performances are evaluated on Rain100L dataset.

Rain removal methods	Params	Test-time(s)
DSC	-	146.761
CNN	-	0.204
DerainNet	-	1.6991
DID-MDN	135800	0.15313
ID-CGAN	817824	0.286
SPANet	283716	2.301
LPNet	7548	0.67
Ours	50703	0.13310

Table 4 shows the number of network parameters and test times for different rain removal methods. It can be seen that the traditional model-driven DSC method requires a larger testing time [17]. Deep learning-based rain removal method, which have pre-trained models that can be used directly, have less testing time and higher rain removal efficiency than model-driven rain removal methods. Although the LP-Net rain removal network, which has the least number of network parameters, takes longer time and is less effective in removing rain streaks [33]. Our proposed method uses a CBAM and GRU module, which is lightweight network. The second lowest number of parameters and the least testing time compared to the LPNet network.

Conclusion

In this paper, we propose a lightweight single image deraining method incorporating visual saliency for the balance of de-rain performance and network parametric number in the de-rain algorithm. Specially, our method comprises of three network modules: dilated convolution, CBAM, gated recurrent network module. The dilated convolution module broaden the perceptual field gradually to extract the contextual information of the input rainy image. The CBAM belongs light-weight, so it doesn't bring too weight burden when assigning the difference weight for feature maps, and the mean while the gated recurrent network module could remove various rain streak through multi-stage iteration. Extensive experiments on synthetic and real-world datasets demonstrate that our method outperforms the baseline both under qualitative and quantitative analysis. Under the same rain removal effect, our method is less time cost and fewer parameters. Our rain removal method is more conducive to the synergistic implementation with subsequent high-level tasks.

Appendix

Acknowledgements

Thanks to previous work on open source code and datasets.

Funding

This work was supported by the Key Research and Development Project of Shaanxi Province (No. 2018KW-050), the National Natural Science Foundation of China (No. 62071378), and the Xi'an Science and Technology Plan Project (No. 21XJZZ0072).

Abbreviations

Availability of data and materials

The datasets analysed during the current study are available in the [Review on deep learning based single image rain removal] repository, [<https://github.com/mendy-2013>].

Competing interests

The authors declare that they have no competing interests.

Table 5 The investigation of the number of stages. Performances are evaluated on Rain100L dataset.

Abbreviations	Full name
DL	Deep learning
LDVS	light-weight single image deraining algorithm incorporating visual attention saliency
GAN	Generative Adversarial Networks
CBAM	Convolutional Block Attention Module
GRU	Gate recurrent unit
BEN	Background Extraction Network
HREN	High-frequency Rain-streak Elimination Network
MCN	Main Controlling Network
SE	Squeeze and excitation
RCD	Rain Convolutional Dictionary
MLP	Multilayer Perceptron
RNN	Recursive Neural Network
LSTM	Long Short-Term Memory

Authors' contributions

Conceptualization, Mingdi Hu.; methodology, Mingdi Hu.; software, Jingbing Yang.; writing—original draft preparation, Jingbing Yang.; writing—review and editing, Mingdi Hu.; All authors have read and agreed to the published version of the manuscript.

Author details

School of Communication and Information Engineering School of Artificial Intelligence, Xi'an University of Posts and Telecommunications, Xi'an, China.

References

1. Zhang, Z., Wei, Y., Zhang, H., Yang, Y., Yan, S., Wang, M.: Data-Driven single image deraining: A comprehensive review and new perspectives (2021)
2. Wang, H., Wu, Y., Li, M., Zhao, Q., Meng, D.: A survey on rain removal from video and single image. arXiv preprint arXiv:1909.08326 (2019)
3. Yang, W., Tan, R.T., Wang, S., Fang, Y., Liu, J.: Single image deraining: From model-based to data-driven and beyond. *IEEE Transactions on pattern analysis and machine intelligence* **43**(11), 4059–4077 (2020)
4. Garg, K., Nayar, S.K.: When does a camera see rain? In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2, pp. 1067–1074 (2005)
5. Park, W.-J., Lee, K.-H.: Rain removal using kalman filter in video. In: 2008 International Conference on Smart Manufacturing Application, pp. 494–497 (2008)
6. Li, M., Xie, Q., Zhao, Q., Wei, W., Gu, S., Tao, J., Meng, D.: Video rain streak removal by multiscale convolutional sparse coding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6644–6653 (2018)
7. Ding, J., Guo, H., Zhou, H., Yu, J., He, X., Jiang, B.: Distributed feedback network for single-image deraining. *Information Sciences* **572**, 611–626 (2021)
8. Li, R., Cheong, L.-F., Tan, R.T.: Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1633–1642 (2019)
9. Huang, H., Yu, A., He, R.: Memory oriented transfer learning for semi-supervised image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7732–7741 (2021)
10. Wang, C., Wu, Y., Su, Z., Chen, J.: Joint self-attention and scale-aggregation for self-calibrated deraining network. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 2517–2525 (2020)
11. Chen, J., Tan, C.-H., Hou, J., Chau, L.-P., Li, H.: Robust video content alignment and compensation for rain removal in a cnn framework. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6286–6295 (2018)
12. Ding, X., Chen, L., Zheng, X., Huang, Y., Zeng, D.: Single image rain and snow removal via guided L0 smoothing filter. *Multimedia Tools and Applications* **75**(5), 2697–2712 (2016)
13. Gu, S., Meng, D., Zuo, W., Zhang, L.: Joint convolutional analysis and synthesis sparse representation for single image layer separation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1708–1716 (2017)
14. Zheng, X., Liao, Y., Guo, W., Fu, X., Ding, X.: Single-image-based rain and snow removal using multi-guided filter. In: International Conference on Neural Information Processing, pp. 258–265 (2013)
15. Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2736–2744 (2016)
16. Kang, L.-W., Lin, C.-W., Fu, Y.-H.: Automatic single-image-based rain streaks removal via image decomposition. *IEEE transactions on image processing* **21**(4), 1742–1755 (2011)
17. 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, pp. 3397–3405 (2015)
18. Chang, Y., Yan, L., Zhong, S.: Transformed low-rank model for line pattern noise removal. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1726–1734 (2017)
19. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
20. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)

21. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014)
22. Ren, D., Zuo, W., Hu, Q., Zhu, P., Meng, D.: Progressive image deraining networks: A better and simpler baseline. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3937–3946 (2019)
23. Ni, S., Cao, X., Yue, T., Hu, X.: Controlling the rain: From removal to rendering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6328–6337 (2021)
24. Zhu, H., Wang, C., Zhang, Y., Su, Z., Zhao, G.: Physical model guided deep image deraining. In: *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6 (2020)
25. Wang, H., Xie, Q., Zhao, Q., Liang, Y., Meng, D.: RCDNet: An interpretable rain convolutional dictionary network for single image deraining. *arXiv preprint arXiv:2107.06808* (2021)
26. Yang, W., Tan, R.T., Feng, J., Guo, Z., Yan, S., Liu, J.: Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence* **42**(6), 1377–1393 (2019)
27. Deng, S., Wei, M., Wang, J., Feng, Y., Liang, L., Xie, H., Wang, F.L., Wang, M.: Detail-recovery image deraining via context aggregation networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14560–14569 (2020)
28. Jiang, N., Chen, W., Lin, L., Zhao, T.: Single image rain removal via multi-module deep grid network. *Computer Vision and Image Understanding* **202**, 103106 (2021)
29. Yang, H.-H., Huang, K.-C., Chen, W.-T.: Laffnet: A lightweight adaptive feature fusion network for underwater image enhancement. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 685–692 (2021)
30. Das, S.D., Dutta, S.: Fast deep multi-patch hierarchical network for nonhomogeneous image dehazing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 482–483 (2020)
31. Zhang, J., Tao, D.: FAMED-Net: A fast and accurate multi-scale end-to-end dehazing network. *IEEE Transactions on Image Processing* **29**, 72–84 (2019)
32. Yamamichi, K., Han, X.-H.: Lightweight Multi-Scale context aggregation deraining network with Artifact-Attenuating pooling and activation functions. *IEEE Access* **9**, 146948–146958 (2021)
33. Fu, X., Liang, B., Huang, Y., Ding, X., Paisley, J.: Lightweight pyramid networks for image deraining. *IEEE transactions on neural networks and learning systems* **31**(6), 1794–1807 (2019)
34. Li, X., Wu, J., Lin, Z., Liu, H., Zha, H.: Recurrent squeeze-and-excitation context aggregation net for single image deraining. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 254–269 (2018)
35. Li, S., Araujo, I.B., Ren, W., Wang, Z., Tokuda, E.K., Junior, R.H., Cesar-Junior, R., Zhang, J., Guo, X., Cao, X.: Single image deraining: A comprehensive benchmark analysis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3838–3847 (2019)
36. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014)
37. Shao, M.-W., Li, L., Meng, D.-Y., Zuo, W.-M.: Uncertainty guided Multi-Scale attention network for raindrop removal from a single image. *IEEE Transactions on Image Processing* **30**, 4828–4839 (2021)
38. Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3855–3863 (2017)
39. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology* **30**(11), 3943–3956 (2019)
40. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1357–1366 (2017)
41. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.H.: Spatial attentive single-image deraining with a high quality real rain dataset. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12270–12279 (2019)
42. Eigen, D., Krishnan, D., Fergus, R.: Restoring an image taken through a window covered with dirt or rain. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 633–640 (2013)
43. Fu, X., Huang, J., Ding, X., Liao, Y., Paisley, J.: Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing* **26**(6), 2944–2956 (2017)
44. 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*, pp. 695–704 (2018)
45. Fu, X., Qi, Q., Zha, Z.-J., Zhu, Y., Ding, X.: Rain streak removal via dual graph convolutional network. In: *Proc. AAAI Conf. Artif. Intell.*, pp. 1–9 (2021)

Figures

(a) Ground Truth

(b) Rainy image

(c) Rain Streak Layer

(d) Reconstruction Background

Figure 1

Example of rain removal results using our proposed rain removal method on synthetic Rain800 datasets.

Figure 2

Rain model.

Figure 3

Lightweight single image de-raining network framework incorporating visual saliency (LDVS). The network mainly consists of dilated convolution module, a CBAM and GRU module.

Figure 4

Receptive fields under different expansion coefficients.

Figure 5

The basic structure of CBAM.

Figure 6

The basic structure of the GRU.

Figure 7

Different convolution types. (a) shows the ordinary convolution, (b) shows the combination of dilated convolution and CBAM.

Figure 8

Test results of different rain removal algorithms on the synthetic datasets.

Figure 9

Test results of different rain removal algorithms on the real-world dataset.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [LVDS11.rar](#)