



# Dual-attention U-Net and multi-convolution network for single-image rain removal

Ziyang Zheng<sup>1</sup> · Zhixiang Chen<sup>1</sup> · Shuqi Wang<sup>2,3</sup> · Wenpeng Wang<sup>1</sup>

Accepted: 18 October 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

Images taken on rainy days have rain streaks of varying degrees of intensity, which seriously affect the visibility of the background scene. Aiming at the above problems, we propose a rain mark removal algorithm based on the combination of dual-attention mechanism U-Net and multi-convolution. First, we add a double attention mechanism to the encoder of U-Net. It can give different weights to the rain mark features that need to be extracted in different channels and spaces so that sufficient rain mark features can be obtained. With different dilation factors, we can obtain rain mark characteristics of different depths. Secondly, the multi-convolutional channel integrates the characteristics of rain streaks and prepares sufficient rain mark information for the task of clearing rain streaks. By introducing a cyclic rain streaks detection and removal mechanism into the network architecture, it can achieve gradual removal of rain streaks. Even in the case of heavy rain, our algorithm can get good results. Finally, we tested on both synthetic and real datasets to obtain subjective results and objective evaluations. Experimental results show that for the rainy day image de-rain task with different intensities of rain streaks, our algorithm is more robust. Moreover, the ability of our algorithm to remove rain streaks is better than that of the other five different classical algorithms. The de-raining images produced by our algorithm are visually sharper, and its visibility enhancements are effective for computer vision applications (Google Vision API).

**Keywords** Image processing · Dual-attention mechanism · U-Net · Single-image de-rain · Feature extraction · Convolutional neural networks

## 1 Introduction

A variety of complex rainfall environments can affect the capture, resulting in images that may have reduced visibility, which can seriously affect the performance of outdoor

computer vision systems such as object tracking [1], video surveillance [2], and pedestrian detection [3]. As shown in Fig. 1, the presence of rain marks greatly affects the visual effect of the image. Thereby, it poses a greater challenge to vehicle identification and verification algorithms in rainfall weather. In addition, in this era of widespread cell phone use, images taken by cell phone cameras in adverse weather conditions can degrade. The visual quality of the images can be greatly affected, resulting the images cannot be shared or used. To restore all aspects of the quality of these degraded images while ensuring the enhanced performance of the vision algorithms for better visualization, these negative smudges due to the aforementioned poor weather conditions must be automatically removed. Therefore, stripe removal from rain images is an important pre-processing task, and this research [4, 5] topic attracted extensive research concern in recent years.

From a mathematical point of view, one rainy day image can be decomposed into two independent images. As shown in Fig. 2, one is a striped image with rain and the other is

✉ Zhixiang Chen  
zxchenphd@163.com

Ziyang Zheng  
eva.0311@foxmail.com

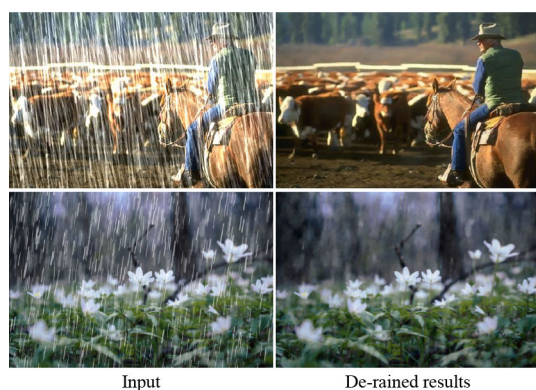
Shuqi Wang  
347423159@qq.com

Wenpeng Wang  
wpp\_9527@163.com

<sup>1</sup> School of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, Fujian, China

<sup>2</sup> School of Computer, Minnan Normal University, Zhangzhou 363000, Fujian, China

<sup>3</sup> Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou 363000, Fujian, China



**Fig. 1** Sample results of the proposed AUCNN method for single-image de-raining

a clean background image. Therefore, the entered image of the rainy day can be shown as:

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

where  $O$  represents the rainy day image,  $R$  represents the rain streaks, and  $B$  represents the clean background image. Thus, image rain removal can be seen as a problem of separating two components from a rainy image, which is similar to the problem of image denoising [6] and image separation [7].

A common strategy to solve (1) in the rain streak removal task for video is to exploit additional temporal information as proposed in [8, 9]. However, there is no temporal information that can be used in the process of dealing with the individual image rain removal problem. In previous work in solving (1), researchers design appropriate priors such as sparse priors [10, 11], Gaussian mixture model (GMM) priors [12], and base order priors [13]. In recent years, convolutional neural networks (CNNs) have also been successfully applied to solve individual image rain removal problems as researchers have released various large-scale synthetic training samples [14, 15]. By directly learning the nonlinear mapping between the source rainfall image and its respective rain removal image, the CNN-based approach achieves superior visual performance. While the methods mentioned above have met with varying degrees of success, most of the available methods have several limitations:

- Due to the inherent overlap between rain streaks and the background image in rainy day images, the majority of

methods try to remove the details of the texture in the non-rain streak areas, resulting in excessive local detail smoothing.

- It is complex to recover the quality of various aspects of rainfall images. Most of the existing methods apply previous models that are insufficient to cover some of the important factors in the real images of rainfall, such as the veil of the atmosphere due to the accumulation of rainfall marks and the different shapes or directions of the rainfall marks.
- Many existing methods work only on local image patches or limited receptive fields to eliminate rain streaks. Therefore, those methods that have proven useful for removing rain marks are rarely used in the case of larger regions of spatial background information [16].

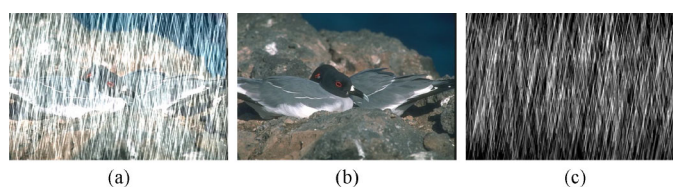
Considering these limitations, our goal is to work on the development of a rain streak removal algorithm that can acquire more and more accurate information about various rain streak features in real scenes, which include rain streak accumulation and rain storms, and then strip and remove them from the rain images.

First, we propose a contextual information expansion network combining a dual-attention mechanism and U-Net [17] to expand the receptive domain. This network not only inherits the advantages of U-Net for multi-scale feature recognition of images but also incorporates the characteristics of spatial attention and channel attention for weight assignment of useful information. Enhance useful information and weaken useless information while deepening the network. Thus, rain mark areas are automatically detected and more and more accurate image features and rain mark details are extracted. This information is used to constrain rain removal to achieve the effect of rain mark detection and removal. Our algorithm can perform adaptive operations on rain-streaked areas and clean background areas, which preserves richer details after removing rain streaks.

Secondly, to extract more information about rain streak features, we propose a multi-convolution channel improved by the discriminator of CycleGAN [18]. To remove rain marks to provide more and more accurate image features and rain mark details, we perform further refinement of features and details extraction on rain mark features in this module.

Finally, to recover images captured in environments with rainfall accumulation and different rain trace directions, we

**Fig. 2** Rain streak removal from a single image. A rainy image (a) can be viewed as the superposition of a clean background image (b) and a rain streak image (c)



propose a recurrent network for rain trace detection and removal, which contains a specific loss function at each stage of the network to make corresponding feedback and adjustment for that stage. It enables the gradual removal of rain marks while making a substantial improvement in the model effect. Extensive experimental and evaluation results show that our algorithm has better restoration capabilities on both synthetic and real data. Our algorithm is more robust and produces visually clearer de-rain images. In particular, our algorithm achieves pretty well for some images taken under heavy rainfall conditions.

In summary, this paper makes the following contributions:

1. A rain streak removal algorithm combining a dual-attention mechanism U-Net and multi-convolution is proposed. It can effectively identify and extract rain streaks from rain images and remove them, restoring the details of the images and making them visually clearer.
2. Further extraction of rain features using multiple convolution channels to obtain more image features and rain details while retaining rich local details.
3. By introducing a cyclic rain streak detection and removal mechanism and using different loss functions at each stage of the network, it can gradually remove the rain streaks and generate a clean image without rain streaks. We were able to get pretty good results even with images that were taken under heavy rainfall conditions.

The paper is organized as follows. Related rain removal efforts are reviewed in Sect. 2. Section 3 introduces the AUCNN model for removing rain marks and the details of the designed algorithm. The experimental results of the synthetic and real images and the quality indicators are given in Sect. 4. Finally, Sect. 5 summarizes and discusses the paper briefly.

## 2 Related work

In this section, we briefly review the knowledge and related literature on existing single-image rain removal methods. These rain removal methods can be broadly classified into two categories: a priori-based methods and deep learning-based methods.

### 2.1 Prior-based methods

A difficult point of single-image de-raining is that it does not lack temporal information, so most early de-raining methods tried to explore additional a priori information to overcome this challenge. Kim [19] assumes that the rain streaks are elliptical, as a way to recover the detected rain streaks through a non-local averaging filter for the regions with rain. Luo

[20] proposed a discriminative sparse coding framework, which is capable of separating image patches from background images that do not contain rain streaks to obtain clean images. Wang et al. [21] extracted non-rain details by rain streak orientation and variance sensitivity of color channels. Li et al. [12] separated the rain streak layer from the background image by a patch-based a priori GMM. The above a priori-based rain removal method can remove some rain streaks, but the modeling process is complicated and the rain removal effect is average. These limit their application to practical tasks.

### 2.2 Deep-learning-based methods

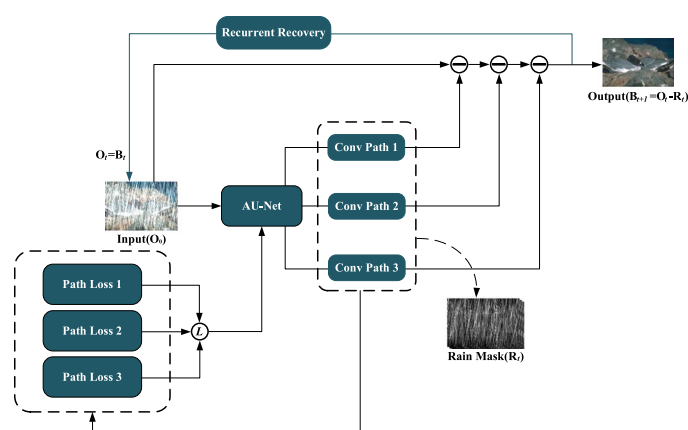
In recent years, single-image rain removal methods based on deep networks have achieved good performance. Fu et al. [16] first applied convolutional neural networks to the field of single-image de-raining. He used deep learning techniques to learn the nonlinear mapping between clean and rain images and to remove rain streaks from a single image. Li et al. [22] used a pixel-level attention mechanism for rain image recovery. Ren et al. [23] proposed a progressive recursive network. He used the stepwise results and the original rain image as input to progressively generate clean output images. Pan et al. [24] proposed a method capable of learning rain streaks and rain-free images together, which was implemented by a pairwise convolutional network. Chen et al. [25] proposed a single-image de-raining method based on the feedback mechanism in control theory from the perspective of error detection and error compensation. Zhang et al. [26] synthesized another dataset and proposed a multi-stream dense network based on rain-strip density classification. Wang et al. [27] proposed a spatial attention network to eliminate rainfall streaks in a local-to-global manner. Although the above approaches have made considerable improvements, they remain with many shortcomings. For example, they cannot recover the details of images very well.

## 3 Proposed method

To further limit the ability to remove rain streaks during the recovery of rain images while utilizing more background without losing the loss of local details, we construct a combined dual-attention mechanism U-Net and multi-convolution algorithm for rain streaks detection and removal. Our algorithm can solve the inverse problem in (1) and achieve rain streak detection and removal for a single rain image.

As shown in Fig. 3, the proposed rain trace removal algorithm (AUCNN) consists of three phases: (1) The rain streak recognition feature extraction stage of the dual-attention

**Fig. 3** An overview of the proposed AUCNN method for single-image de-raining



mechanism U-Net. (2) Refinement of rain streak features and detail re-extraction stages for multiple convolution channels. (3) Repeat the previous two stages to achieve layer-by-layer removal of rain streaks and finally generate a clean image with rain streaks removed. The modules involved in these three phases are described in detail in the following subsections.

### 3.1 Contextual information expansion network combining U-Net and dual attention

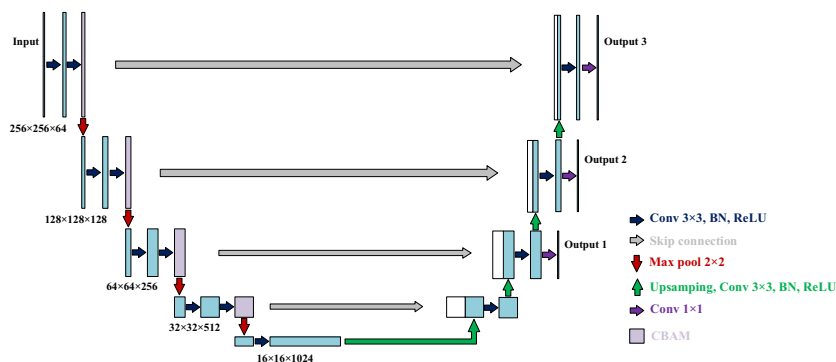
In the rain streak removal task, contextual information from the input image is very important for the automatic recognition and removal of rain streaks. U-Net is an image segmentation feature extraction network with an encoder-decoder structure. The encoder captures the contextual feature information, and the decoder reduces the abstracted features to the dimensions of the original image. The presence of a jump connection between the encoder and decoder helps the decoder to better repair the details of the target. It is widely used in image segmentation tasks because of its ability to obtain sufficiently detailed features without losing edge features. The rain streak detection module of our algorithm is improved from the base U-Net by introducing the convolutional block attention module (CBAM) [28] in its encoder part, and the framework of this network is shown in Fig. 4.

In this paper, we propose a convolutional neural network for rain streak detection, called AU-Net. AU-Net provides an increasingly large receptive domain for subsequent layers through a cyclic structure, and it can use contextual information to obtain adequate rain streak features. AU-Net includes the encoder part and the decoder part, and we have improved the encoder part mainly. AU-Net has five downsampling and five upsampling layers, and the total number of convolutional layers is eighteen, and it uses a fully convolutional neural network instead of fully connected layers. As the number of convolutions increases, the extracted features become more effective and more abstract. The input volume of AU-Net

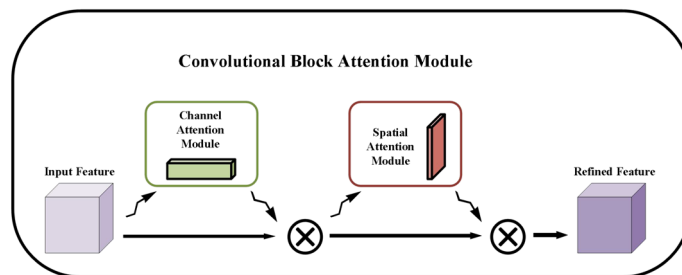
is a color image of size  $256 \times 256$ . The addition of the CBAM after each convolutional layer in the encoder part makes AU-Net reassign weights to image features based on downsampled feature extraction. It can emphasize useful feature information and weaken useless interference information, which facilitates bringing up more and more useful features and details. The convolutional layers of both the contracting path and the expanding path consist of convolutions with a kernel size of  $3 \times 3$ , and a ReLU is added after each convolutional layer. The number of convolution kernels in the first to the fifth layer in the shrinkage path is 64, 128, 256, 512, and 1024, where the size of the feature maps is  $256 \times 256$ ,  $128 \times 128$ ,  $64 \times 64$ ,  $32 \times 32$ , and  $16 \times 16$ , respectively. The contracting path and expanding path are symmetric, with a twice contracted downsampling between every two convolutional layers in the contracting path and a twice expanded upsampling between every two convolutional layers in the expanding path. The fused feature information is spliced between the same layers using jump connections to stitch the features together in the channel dimension to form thicker features. The upsampled image is a highly efficient abstract image obtained by undergoing multiple convolutions, and finally, it is joined with the low abstraction high-resolution feature image obtained by downsampling on the left to obtain an image with more features and more details. As shown in Fig. 4, we output the data of the last three layers of AU-Net separately and use them as input for the next module. We parse the features of different depths as a way to obtain richer information about the rain streak features. We hope to maximize the use of rain streak features and thus improve the performance of the model.

The CBAM in AU-Net is a convolutional block-based attention module, which combines spatial attention mechanism and channel attention mechanism, and it can significantly improve the correct rate of image classification and target detection. As shown in Fig. 5, the CBAM contains two independent sub-modules, which are channel attention module (CAM) and spatial attention module (SAM). Each of them

**Fig. 4** The overview of AU-Net. The architecture of U-Net and CBAM convergence. Each blue box corresponds to a multi-channel feature map. The  $x$ - $y$ -size is provided at the lower left edge of the box. White boxes represent copied feature maps. Purple boxes represent feature maps after processing by the CBAM module. The arrows denote the different operations



**Fig. 5** The overview of CBAM. The module has two sequential sub-modules: channel and spatial. The intermediate feature map is adaptively refined through CBAM at every convolutional block of deep networks



performs the attention weight assignment on the channel and space, which can extract the image feature information more fully.

The channel attention module focuses on the important content information in the input image with constant channel dimension and compressed spatial dimension. The pooling layer is set up to compress the features while removing redundant information. The average pooling layer assigns weights to each pixel point, and the maximum pooling layer takes the maximum weight value within a certain range as the weight of the field. As shown in Fig. 6, first of all, *Input Feature*  $F$  performs the maximum pooling and global pooling operations. After that, the image information is compressed by shared multilayer perceptron (MLP) for the feature compression operation of dimensionality reduction and then dimensionality enhancement. Finally, the resulting features are subjected to a summation operation and then a sigmoid activation operation to generate the channel attention weights, namely  $M_c$ . The  $M_c$  is multiplied with the *Input Feature*  $F$ , and finally, the output feature map of the channel attention is derived, namely *Channel-refined feature*  $F'$ .

The spatial attention module focuses on the important location information in the input image with constant spatial dimensionality and compressed channel dimensionality. The *Channel-refined feature*  $F'$  is used as input for spatial attention, with global maximum pooling and global average pooling operations on the channel. After that, the two obtained feature maps are merged and passed through a convolutional layer consisting of  $7 \times 7$  convolutional kernels. Finally, the compressed single-channel feature map is

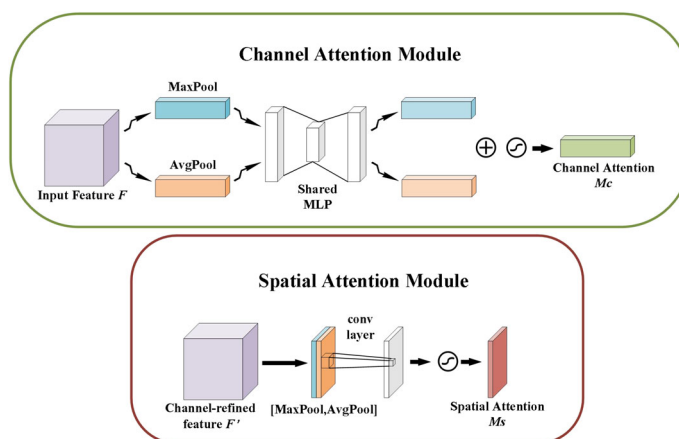
subjected to a sigmoid activation operation to generate spatial attention weights, namely  $M_s$ . Multiplying  $M_s$  with the *Channel-refined feature*  $F'$ , the output feature map of spatial attention is finally obtained, namely  $F''$ .

The CBAM combines channel attention and spatial attention to increase the weight of important features in the channel and space. The mixed pooling of global average and global maximum on both space and channel enables AU-Net to reduce the amount of redundant information computed and reduce memory overhead. It can significantly improve the accuracy of feature detection and feature extraction and provide a large amount of feature data for the subsequent rain removal work to achieve more accurate rain removal work and obtain clean images with clearer visual effects.

### 3.2 Enhanced multi-convolution path

To do further feature extraction on the rain streak feature information output from the contextual information expansion network, we propose a multi-convolutional channel improved by CycleGAN discriminator to achieve learning rain streak features of different degrees and depths on a multi-scale and multi-channel basis. Based on the original discriminator, we preserve the convolutional layer and remove the average pooling layer so that the output value of the discriminator is the weight instead of the discriminator result. It can also determine the current progress of rainfall work based on the loss function of the CycleGAN discriminator while optimizing the relevant parameters to improve the rain removal effect and rain removal efficiency. As shown

**Fig. 6** Diagram of each attention sub-module. As illustrated, the channel sub-module utilizes both max-pooling outputs and average pooling outputs with a shared network. The spatial sub-module utilizes similar two outputs that are pooled along the channel axis and forward to a convolution layer



in Fig. 3, the rain streak feature information of three different paths in the contextual information expansion network is used as the input of three multi-convolutional channels. In each recursion, the multi-convolutional channel refines these features to obtain different degrees of rain trace feature information for different degrees of rain removal, achieving layer-by-layer rain removal. Each multi-convolutional channel consists of five convolutional layers with a  $4 \times 4$  convolutional kernel, each followed by a normalization layer and a LeakyReLU activation function.

### 3.3 Loss function

Our cyclic framework is obtained by improving CycleGAN. There are two adversarial losses in the original CycleGAN. The loss function of the contextual information expansion network in our algorithm plays one of the loss-fighting roles. The other loss-fighting role is played by the aggregation of the loss function of the modified multi-convolutional channel and the exclusion loss.

#### 3.3.1 The loss function of the contextual information expansion network.

The main body of our proposed contextual information expansion network consists of U-Net, so we choose the loss function of U-Net as the loss function of the contextual information expansion network. The U-Net uses a loss function with boundary weights, so we define the loss function of the AU-Net as:

$$E = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x)) \tag{2}$$

where  $p_k(x) = \frac{\exp(a_k(x))}{\sum_{n=1}^K \exp(a_n(x))}$ ,  $a_k(x)$  denotes the activation in feature channel  $k$  at the pixel position  $x \in \Omega$  with  $\Omega \in \mathbb{Z}^2$ .  $K$  is the number of classes, and  $p_k(x)$  is the approximated

maximum function. The  $l : \Omega \rightarrow \{1, \dots, K\}$  is the true label of each pixel, and  $w : \Omega \rightarrow \mathbb{R}$  is a weight map that we introduced to give some pixels more importance in the training.

We segment the weight maps based on each rainy day image to compensate for the different frequencies of a certain class of pixels in the training dataset and force the network to learn the small separation boundaries we introduce between contact units. Separation boundaries were calculated using morphological operations. We define the weight map calculation formula as:

$$w(x) = w_c(x) + w_0 \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right) \tag{3}$$

where  $w_c : \Omega \rightarrow \mathbb{R}$  is the weight map to balance the class frequencies,  $d_1 : \Omega \rightarrow \mathbb{R}$  denotes the distance to the border of the nearest rain trace, and  $d_2 : \Omega \rightarrow \mathbb{R}$  is the distance to the border of the second nearest rain trace. In our experiments, we set  $w_0 = 10$  and  $\sigma \approx 5$  pixels.

In context expansion networks, if similar rain marks are close to each other, it may increase the difficulty of training and reduce the accuracy. The convolution will only consider the local features around that pixel point, and it is easy to misjudge if two similar rain marks are pasted together. So we give a larger weight to this kind of two similar pasted together rain trace boundaries. As shown in (3), the U-Net loss function assigns weights to pixels and then weights them. The  $d_1(x)$  denotes the distance from a background pixel point in the graph to the nearest rain trace boundary, and the  $d_2(x)$  denotes the distance to the second closest rain trace to this pixel point. The pixel points near the rainfall trace boundary are given larger weights, and the pixel points farther away from the rainfall trace are given smaller weights, which makes the rainfall recognition feature extraction more accurate after training. The role of the loss function of the contextual information expansion network is to adjust the network and the optimizer by calculating the loss of this

epoch image  $B_{t+1}$  after processing and image  $B_t$  before processing. It enables AU-Net to segment images better and obtain richer details of rain traces.

### 3.3.2 The loss function of the improved multi-convolution channel

We propose a multi-convolutional channel improved by CycleGAN discriminator to achieve learning rain trace features of different degrees and depths on a multi-scale and multi-channel basis. Each convolutional path is improved by a separate CycleGAN network discriminator, so each multi-convolutional channel with its loss function. The effectiveness of layer-by-layer rain removal and the progress of the work are judged based on the values obtained from the loss function. It can optimize the relevant parameters to improve the accuracy of rain trace feature recognition extraction and make rain removal more efficient, resulting in a clean image with clearer visual effects.

$$\mathcal{L}_{AU-Net}(U, D_B, X, B) = \mathbb{E}_{b \sim p_{data}(b)} [\log D_B(b)] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D_B(U(x)))] \quad (4)$$

where  $U$  tries to generate images  $U(x)$  that look similar to images from domain  $B$ , while  $D_B$  aims to distinguish between translated samples  $U(x)$  and real samples  $b$ .  $U$  aims to minimize this objective against adversary  $D$  which tries to maximize it. The purpose of our proposed mapping function is to optimize the rain removal effect and task efficiency through the loss function of AU-Net and the loss function of multiple channels.

### 3.3.3 Exclusion loss

To better separate the rain trace layer from the background layer, we explore the relationship between the two layers by analyzing the edges of the two layers. Our observation is that there is little probability that the edges of the rain trace layer overlap with the edges of the background layer. Therefore, we minimize the correlation between the predicted rain trace layer and the background layer. We express the exclusion loss as the product of the normalized gradient fields of the two layers at multiple spatial resolutions, so the loss function is defined as:

$$L_{\text{excl}}(\theta) = \sum_{I \in D} \sum_{n=1}^N \left\| \psi \left( f_T^{\downarrow n}(I; \theta), f_R^{\downarrow n}(I; \theta) \right) \right\|_F \quad (5)$$

$$\psi(T, R) = \tanh(\lambda_T \nabla T) \odot \tanh(\lambda_R \nabla R) \quad (6)$$

where  $f_T(I; \theta)$  and  $f_R(I; \theta)$  are derived from the decomposition of  $f(I; \theta) = (f_T(I; \theta), f_R(I; \theta))$ ,  $I$  is the input images,  $\theta$  is the network weights,  $D = \{(I, T, R)\}$ ,  $T$  is

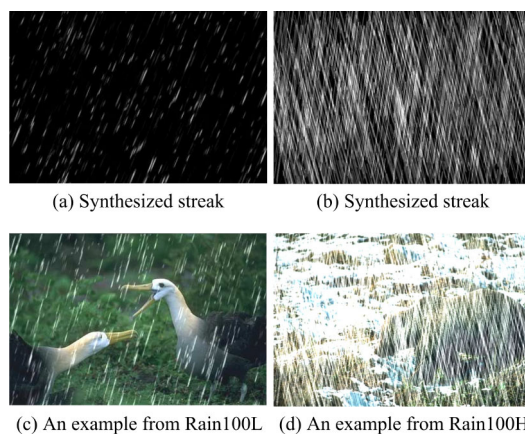


Fig. 7 Sample images from real-world rainy dataset

the transmission layer of  $I$ , and  $R$  is the reflection layer of  $I$ .  $\lambda_T$  and  $\lambda_R$  are normalization factors,  $\|\cdot\|_F$  is the Frobenius norm,  $\odot$  denotes element-wise multiplication, and  $n$  is the image downsampling factor: The images  $f_T$  and  $f_R$  are downsampled by a factor of  $2^{n-1}$  with bilinear interpolation.

## 4 Experiments and results

In this section, we detail the experimental and quality metrics used to evaluate the proposed multi-channel single-image rain removal network. We also discuss the dataset and training details and then compare the proposed method with recent approaches.

### 4.1 Experimental details

#### 4.1.1 Synthetic dataset

We compared our approach with state-of-the-art methods on several benchmark datasets: (1) Rain100L, which is the synthetic data set with only one rain pattern (Fig. 7c). (2) Rain100H, which is the dataset we synthesized for the five directions (Fig. 7d). Please note that although real rainfall images rarely contain rain traces in many different directions, synthesizing such images for training can increase the capacity of the network.

Rain streaks are synthesized in two ways: (1) The realistic rendering technique proposed in [29], as shown in Fig. 7a. (2) Simulation of rain traces along a certain direction, with the disadvantage that there is less variation within the image, as shown in Fig. 7b.

#### 4.1.2 Real-world rainy images dataset

To demonstrate the effectiveness of the method on real data, we created a dataset consisting of 300 images of rainy days

**Fig. 8** Sample images from real-world rainy dataset



downloaded from the Internet. In creating the dataset, we were as careful as possible to ensure that the images collected were different in terms of content, the intensity of rain pixels, and orientation. Some sample images from this dataset are shown in Fig. 8. This dataset was used for evaluation purposes only.

#### 4.1.3 Model details and parameters

We trained and implemented our network on an NVIDIA TITAN Xp GPU using the torch framework [30]. We use the Adam optimizer [31] with a batch size of 16. The initial learning rate is  $1 \times 10^{-3}$ , and the total epoch is 100, which is the total number of cycles.

#### 4.1.4 Quality measures

The performance metrics of different methods include peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [32]. Similar to the previous method [12], all these quantitative measurements are calculated using the luminance channel. Since we do not have ground truth reference images of the real dataset, the performance of our proposed algorithm and other methods on the real dataset is evaluated by vision.

## 4.2 Comparison with state-of-the-art methods

### 4.2.1 Evaluation of synthetic dataset

In the first set of experiments, we evaluate the proposed method. We compared its quantitative and qualitative performance with several state-of-the-art methods from synthetic datasets. Since the ground truth of these test images is available, we calculate quantitative metrics such as PSNR and SSIM. The comparison of the results based on these indicators is shown in Table 1. The table clearly shows that our proposed AUCNN method can obtain superior quantitative performance compared to recent methods in terms of all the metrics described above.

Figures 9 and 10 show the qualitative improvements achieved on the sample images from two synthetic datasets of different degrees due to the use of the proposed method. Note that we have selectively sampled complex images to show that our method performs well under complex conditions. In the tests on the synthetic dataset Rain100L, although JORDER\_E [35] and LPNet [37] can reduce the rain streak

density or remove some of the streaks, they also cannot completely remove the rain streaks. As shown in Fig. 9, there are still some rain remnants in the sky in the fourth sample. In tests with the synthetic dataset Rain100H, PReNet [23] was able to remove rain marks, but it produced blurred results that were not visually appealing. As shown in Fig. 10, the iceberg in the third sample and the roof part in the fourth sample lose some details and appear to be over-smoothed. The other methods are significantly less effective than PReNet [23] in removing rain under heavy rainfall conditions. Compared with other methods, our proposed AUCNN method can remove most of the rain marks while preserving the rich details of the image.

### 4.2.2 Evaluation of real rainy images

We evaluate the performance of our proposed method and recent methods on real-world rain test images. The rain removal results of all methods for the rain images of the four sample inputs are shown in Fig. 11. For a better visual comparison, we selected several specific regions of interest for comparison in the results with the rain streaks removed. By looking at these regions of interest, we can observe that the DSC [20] tends to add artifacts to the images with the rain streaks removed. Although PReNet [23], JORDER\_E [35], DiG-CoM [36], and LPNet [37] can achieve good visual performance, rain streaks are still visible in the selected region of interest. In contrast, our proposed method can remove most of the rain streaks while maintaining the background image details. We can observe that our proposed method misses some rain streaks in the output image. This is because these few image samples represent relatively complex images of rainy days. However, the method was able to achieve better results compared to the existing methods.

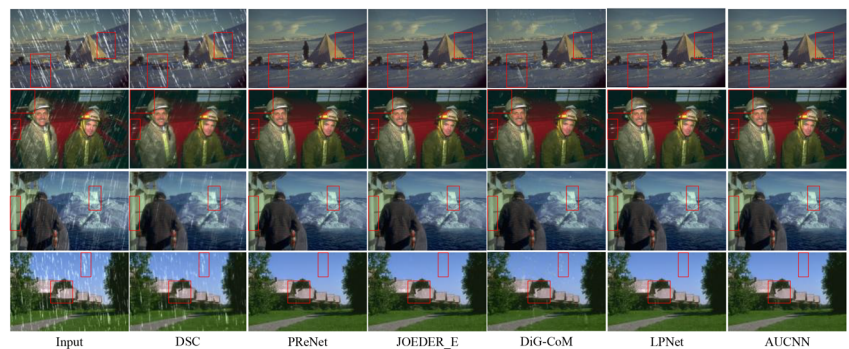
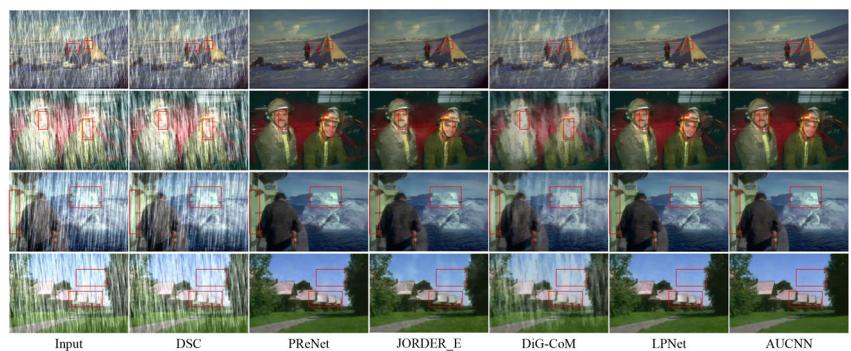
### 4.3 Running time comparison

We compare the running time of the proposed method with other state-of-the-art methods. As shown in Table 2, both the DSC [20] and GMM [29] methods inevitably take a lot of time because traditional model-based methods require several iterations to find the optimal solution. As can be observed in Table 2, our method, PReNet [23], and DerainGAN [38] take less time. However, neither PReNet [23] nor DerainGAN [38] can effectively remove rain streaks (see Sect. 4.2 for details). In short, our method achieves a better balance between performance and time.



**Table 1** PSNR and SSIM comparisons on two benchmark datasets

Datasets	Year	Rain100H		Rain100L	
		PSNR	SSIM	PSNR	SSIM
DSC [20]	2015	13.75	0.313	27.32	0.838
GMM [12]	2016	14.44	0.392	27.76	0.867
DDN [33]	2017	22.26	0.690	34.72	0.950
RESCAN [34]	2018	29.23	0.878	37.77	0.961
PRNet [23]	2019	29.10	0.895	37.45	0.960
SPANet [27]	2019	25.11	0.832	35.33	0.961
JORDER_E [35]	2019	28.75	0.877	38.59	0.963
DiG-CoM [36]	2020	22.40	0.705	33.43	0.937
LPNet [37]	2020	27.03	0.800	36.45	0.964
DerainGAN [38]	2021	29.03	0.856	38.60	0.965
Ours		<b>29.80</b>	<b>0.899</b>	<b>38.70</b>	<b>0.968</b>

**Fig. 9** Qualitative comparison of rain streak removal on four sample images from Rain100L**Fig. 10** Qualitative comparison of rain streak removal on four sample images from Rain100H**Fig. 11** Qualitative comparison of rain streak removal on four sample real images

**Table 2** Average runtime on  $512 \times 512$  images. The best results are bolded

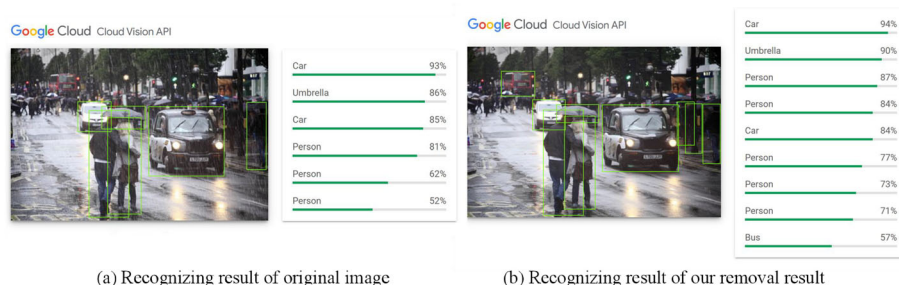
Method	Year	Computing devices	Inference time (s)
DSC [20]	2015	CPU	99.2
GMM [12]	2016	CPU	371.2
DDN [33]	2017	CPU	0.6
RESCAN [34]	2018	CPU	0.6
PReNet [23]	2019	CPU	<b>0.3</b>
SPANet [27]	2019	CPU	0.4
JORDER_E [35]	2019	CPU	3.1
DiG-CoM [36]	2020	CPU	2.8
LPNet [37]	2020	CPU	0.4
DerainGAN [38]	2021	CPU	<b>0.3</b>
Ours		CPU	<b>0.3</b>

## 4.4 Application in high-level computer vision tasks

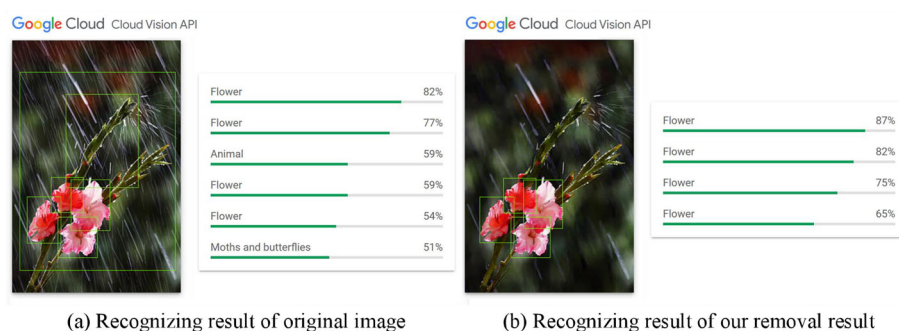
### 4.4.1 Evaluation of object detection results

To further demonstrate that the visibility enhancement of our algorithm works for computer vision applications, we use the Google Vision API to test to prove whether using our clean images with rain marks removed improves recognition performance. We evaluated the real rainfall image dataset. The computational results are shown in Fig. 12. The clean images generated by our algorithm with the rain marks removed

**Fig. 12** A sample of improving the result of Google Vision API. Our method increases the scores of main object detection as well as the number of objects recognized



**Fig. 13** A sample of improving the result of Google Vision API. Our approach effectively corrects the results of Google Vision API recognition errors



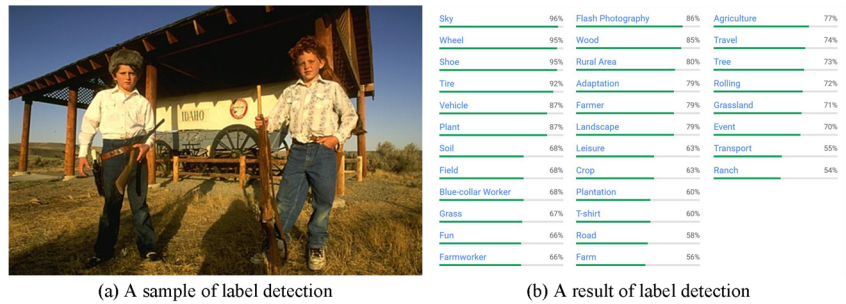
not only have high target detection metrics but also have an improved number of recognized objects. Not only that, but as shown in Fig. 13, the clean images generated by our algorithm with the rain marks removed effectively correct the results of the Google Vision API recognition errors. Experimental results show that the visibility enhancement of clean images with rain streaks removed generated by our algorithm is effective, and it provides a significant improvement in the target recognition performance of computer vision applications.

### 4.4.2 Evaluation of label detection results

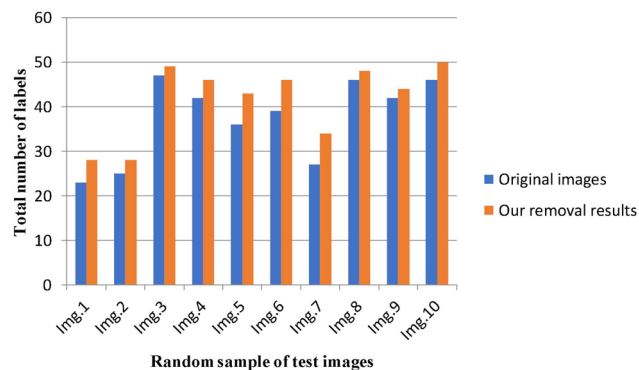
We evaluate the label detection separately for the rainy day images of the real rainfall image dataset and the clean images generated by our algorithm (Fig. 14). As shown in Fig. 15.a, we use it as a sample for label detection evaluation. The result of the label detection evaluation consists of the label name and the labeled percentage. We can determine the contents of the image and the proportion of the contents occupying the whole image based on the results of the label detection evaluation. As shown in Fig. 15.b, we derive thirty-two labels and their respective percentages.

As the statistics in Fig. 16 show, we randomly selected ten real rainfall images (Fig. 15) as statistical samples. In terms of the number of labels for recognized content, the number of labels for clean images generated by our algorithm is improved, and the image visibility enhancement is significant.

**Fig. 14** A demonstration example of Google Vision API experiments



**Fig. 15** Ten random samples of Google Vision API experiments. We randomly selected ten real rain images as experimental samples



**Fig. 16** Statistical chart of the number of labels that identify the content. We randomly selected ten real rain images as statistical samples

## 5 Conclusions

This paper deals with a rain trace removal algorithm based on a combination of a dual-attention mechanism U-Net and multiple convolutions. Channel attention and spatial attention in the network enable the network to detect and extract more rain streaks for better clearance. The multi-convolution channel further refines the rain streak feature to ensure coverage of most of the rain streaks in the rainy day image. By

introducing a cyclic rain streak detection and removal mechanism and embedding specific loss functions at each stage of the network, our algorithm can achieve progressive removal of rain streaks. Our algorithm is trained on different synthetic datasets and the resulting network can perform effective rain removal on real rain image datasets. Although our proposed method cannot completely remove the fog and some rain patches from rainy day images, the visibility enhancement of our algorithm is demonstrated to be effective for computer vision applications as evaluated by Google Vision API. Our algorithm not only improves target detection metrics, but also the number of objects recognized and the number of labels counted in advanced computer vision tasks, and the image information visibility enhancement is significantly better than other methods. Even on images obtained in heavy rain, our algorithm can remove most of the rain streaks and obtain good results while maintaining the background image details. It shows that our algorithm is more capable and robust in removing rain streaks, and it can generate clean images with clearer visual effects.

**Acknowledgements** This work was supported by the Natural Science Foundation of Zhangzhou under Grant ZZ2020J33 and the Natural Science Foundation of Fujian under Grant 2023J01924.

**Author's contribution** ZZ contributed to conceptualization, methodology, data curation, writing—original draft, and writing—review and editing, and provided software. ZC was involved in conceptualization, methodology, supervision, and writing—review and editing. SW contributed to writing—review and editing. WW was involved in writing—review and editing.

**Data availability** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Ethical approval** I testify on behalf of all the co-authors: This material has not been published in whole or in part elsewhere. The manuscript is not currently being considered for publication in another journal. All authors have been personally and actively involved in substantive work leading to the manuscript and will hold themselves jointly and individually responsible for its content.

## References

1. Hang, Y., Derong, C., Jiulu, G.: Object tracking using both a kernel and a non-parametric active contour model. *Neurocomputing* **295**, 108–117 (2018)
2. Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., et al.: A system for video surveillance and monitoring. VSAM final report 2000(1–68), 1 (2000)
3. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 743–761 (2011)
4. Luo, Y., Wu, M., Huang, Q., Zhu, J., Ling, J., Sheng, B.: Joint feedback and recurrent deraining network with ensemble learning. *The Vis. Comput.* **268**, 1–11 (2022)
5. Yang, H., Zhou, D., Li, M., Zhao, Q.: A two-stage network with wavelet transformation for single-image deraining. *Vis. Comput.* **39**(9), 3887–3903 (2023)
6. Fan, L., Zhang, F., Fan, H., Zhang, C.: Brief review of image denoising techniques. *Vis. Comput. Ind. Biomed. Art* **2**(1), 1–12 (2019)
7. Zhang, L., Li, Y., Wang, P., Wei, W., Xu, S., Zhang, Y.: A separation–aggregation network for image denoising. *Appl. Soft Comput.* **83**, 105603 (2019)
8. Garg, K., Nayar, S.K.: Detection and removal of rain from videos. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2004*, vol. 1, IEEE (2004)
9. Wei, W., Yi, L., Xie, Q., Zhao, Q., Meng, D., Xu, Z.: Should we encode rain streaks in video as deterministic or stochastic? In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2516–2525 (2017)
10. Zhang, H., Patel, V.M.: Convolutional sparse and low-rank coding-based rain streak removal. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1259–1267, IEEE (2017)
11. Zhu, L., Fu, C.-W., Lischinski, D., Heng, P.-A.: Joint bi-layer optimization for single-image rain streak removal. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2526–2534 (2017)
12. 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)
13. He, X., Feng, Y., Xie, Q., Wang, J.: Low-rank appearance-preserving rain streak removal from single images. *Commun. Inf. Syst.* **22**(1), 79–102 (2022)
14. Jayaraman, T., Chinnusamy, S.: Performance analysis of a dual stage deep rain streak removal convolution neural network module with a modified deep residual dense network. *Int. J. Appl. Math. Comput. Sci.* (2022). <https://doi.org/10.34768/amcs-2022-0009>
15. Zheng, Z., Chen, Z., Wang, S., Wang, W., Wang, H.: Memory-efficient multi-scale residual dense network for single image rain removal. *Comput. Vis. Image Underst.* **235**, 103766 (2023). <https://doi.org/10.1016/j.cviu.2023.103766>
16. Fu, X., Huang, J., Ding, X., Liao, Y., Paisley, J.: Clearing the skies: a deep network architecture for single-image rain removal. *IEEE Trans. Image Process.* **26**(6), 2944–2956 (2017)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 234–241. Springer (2015)
18. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232 (2017)
19. Kim, J.-H., Lee, C., Sim, J.-Y., Kim, C.-S.: Single-image deraining using an adaptive nonlocal means filter. In: *2013 IEEE International Conference on Image Processing*, pp. 914–917. IEEE (2013)
20. 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)
21. Wang, Y., Liu, S., Chen, C., Zeng, B.: A hierarchical approach for rain or snow removing in a single color image. *IEEE Trans. Image Process.* **26**(8), 3936–3950 (2017)
22. Li, G., He, X., Zhang, W., Chang, H., Dong, L., Lin, L.: Non-locally enhanced encoder-decoder network for single image deraining. In: *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 1056–1064 (2018)
23. 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)
24. Pan, J., Liu, S., Sun, D., Zhang, J., Liu, Y., Ren, J., Li, Z., Tang, J., Lu, H., Tai, Y.-W., et al.: Learning dual convolutional neural networks for low-level vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3070–3079 (2018)
25. Chen, C., Li, H.: Robust representation learning with feedback for single image deraining. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7742–7751 (2021)
26. 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)
27. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: 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)
28. 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)
29. Garg, K., Nayar, S.K.: Photorealistic rendering of rain streaks. *ACM Trans. Graph. (TOG)* **25**(3), 996–1002 (2006)

30. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: a matlab-like environment for machine learning. In: *BigLearn, NIPS Workshop* (2011)
31. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
32. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
33. 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)
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. 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 Trans. Pattern Anal. Mach. Intell.* **42**(6), 1377–1393 (2019)
36. Ran, W., Yang, Y., Lu, H.: Single image rain removal boosting via directional gradient. In: *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE (2020)
37. Fu, X., Liang, B., Huang, Y., Ding, X., Paisley, J.: Lightweight pyramid networks for image deraining. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(6), 1794–1807 (2019)
38. Yadav, S., Mehra, A., Rohmetra, H., Ratnakumar, R., Narang, P.: Deraingan: Single image deraining using wasserstein gan. *Multimed. Tools Appl.* **80**(30), 36491–36507 (2021)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



**Ziyang Zheng** is a graduate student studying in the School of Physics and Information Engineering at Minnan Normal University in Fujian, China. His main research interest is digital image processing.



**Zhixiang Chen** is a professor in the School of Physics and Information Engineering, Minnan Normal University, and received his Doctor of Engineering in 2010. He is a part-time senior engineer of the National Internet of Things, national information engineer, chief engineer of the British Institute of Engineers (IET), examiner of China Network Security Review Technology and Certification Center (CCRC) and obtained the industrial control system of the Ministry of Human Resources and Social Security. His main research interests are industrial Internet information security and digital image processing.



**Shuqi Wang** is a graduate student studying in the School of Computer at Minnan Normal University in Fujian, China. Her main research direction is digital image processing.



**Wenpeng Wang** is a graduate student studying in the School of Physics and Information Engineering at Minnan Normal University in Fujian, China. His main research interest is industrial Internet information security.