See clearly on rainy days: Hybrid multiscale loss guided multi-feature fusion network for single image rain removal

Huiyuan Fu¹ (✉), Yu Zhang¹, and Huadong Ma¹

© The Author(s) 2021.

Abstract The quality of photos is highly susceptible to severe weather such as heavy rain; it can also degrade the performance of various visual tasks like object detection. Rain removal is a challenging problem because rain streaks have different appearances even in one image. Regions where rain accumulates appear foggy or misty, while rain streaks can be clearly seen in areas where rain is less heavy. We propose removing various rain effects in pictures using a hybrid multiscale loss guided multiple feature fusion de-raining network (MSGMFFNet). Specially, to deal with rain streaks, our method generates a rain streak attention map, while preprocessing uses gamma correction and contrast enhancement to enhanced images to address the problem of rain accumulation. Using these tools, the model can restore a result with abundant details. Furthermore, a hybrid multiscale loss combining L_1 loss and edge loss is used to guide the training process to pay attention to edge and content information. Comprehensive experiments conducted on both synthetic and real-world datasets demonstrate the effectiveness of our method.

Keywords single image rain removal; multiple feature fusion; deep learning; hybrid multiscale loss

1 Introduction

Images captured by cameras are one of the most important information sources for intelligent transportation systems, video surveillance systems, self-driving systems, etc. Bad weather such as heavy

rain leads to severe degradation of image quality and poor performance of downstream high level vision tasks like object detection, object tracking, traffic flow monitoring, and so on. De-raining algorithms which can remove the resultant artifacts as a preprocessing step can help to ensure the performance of following high level vision tasks which rely on high-quality input.

However, rain removal is an ill-posed problem due to the variety of rain streaks, and because it is hard to determine where rain streaks occur. Different camera angles, wind direction, and light intensity result in different densities, directions, and shapes of rain streaks in captured pictures. In recent decades, rain removal from a single image has aroused the enthusiasm of many researchers due to its practical application value [1–15]. Traditional optimization based approaches [3, 4, 7, 9, 13] assume that a rainy image is made up of a rain streak layer and a clean background layer and treat it as a decomposition problem. These methods find it difficult to select effective features. Recently, deep learning methods have greatly boosted the performance of computer vision tasks. As a result, several rain removal methods based on convolutional neural networks have been proposed [1, 2, 5, 6, 8, 10, 12, 15, 16]. Compared to traditional optimization methods, the quality of images predicted by deep learning methods has greatly improved. However, the results still have several deficiencies, as can be observed in Figs. 1(b) and 1(c). Most approaches tend to mistake background texture details for rain streaks or retain some rain streaks in rainy regions, as most existing single image de-raining approaches are not designed to fully consider the complexity of rain streaks and lack the ability to handle different types of rain streaks. These issues affect the features extracted by high

¹ Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: H. Fu, fly@bupt.edu.cn (✉); Y. Zhang, yzhang@bupt.edu.cn; H. Ma, mhd@bupt.edu.cn.

Manuscript received: 2021-01-06; accepted: 2021-01-30

Fig. 1 Rain removal results. (a) Input. (b, c) DID-MDN [1] and PReNet [2] methods fail to completely remove rain, resulting in restored images containing apparent rain streaks; details are often lost by DID-MDN. (d) Our method, MSGMFFNet, achieves the best results.

level vision algorithms such as object detection, and degrade the final performance.

In this work, we propose a hybrid multiscale loss guided multiple feature fusion de-raining network (MSGMFFNet) to tackle these issues. During the rain removal task, rain streak regions and rain accumulation regions should be given more attention. However, we cannot simply treat them uniformly, because the characteristics of the two areas differ. Hence, a rain streak map is generated by the attention mechanism to locate the regions with rain streaks, while image enhancement algorithms are used to coarsely eliminate the influence of rain accumulation. We use gamma correction and contrast enhancement to pre-process the original input, as contrast enhancement can improve global visibility, but may lose details in bright areas (see Fig. 2(b)) while gamma correction can preserve details in bright

regions (see Fig. 2(c)). We then fuse the de-rained results of the two enhanced inputs and original input to generate the restored image under guidance of the rain streak map and a feature-fusion confidence score. We also introduce a hybrid multiscale loss made up of L_1 loss and edge loss to focus on texture detail information during the training process. Specifically, we extract features from different upsampled layers to compute multiscale outputs, and a convolutional Scharr filter layer is built to flexibly extract edge information. We calculate L_1 loss and edge loss on the multiscale outputs and their corresponding edge maps during the training process, not only to boost the performance of rain removal, but also to better preserve image texture details.

MSGMFFNet consists of an attention map learning module, a multi-input feature extraction module, a multiple feature fusion module, and a reconstruction

Fig. 2 Multiple inputs and their weight maps. (a) Input image. (b) Contrast enhanced. (c) Gamma corrected. (d) Our de-rained result. (e–g) Generated weight maps for (a–c), respectively. Green areas represent high attention; blue areas, low attention.

module. The attention map learning module generates a rain streak map to distinguish between rainy and non-rainy regions. Under the guidance of the attention map, the multi-input feature extraction module pays more attention to rain streaks and texture information around them. To fully blend multiple features, the fusion module computes weight maps of the characteristics of each input. The reconstruction module restores the original resolution and produces the final de-rained output. The input of MSGMFFNet is a single image containing rain streaks without any other knowledge.

In conclusion, the key contributions of this work are:

- MSGMFFNet, a novel single image de-raining network which extracts and fuses multiple features derived from an original rainy image by contrast enhancement and gamma correction, to better predict de-rained output;
- a new hybrid multiscale loss function based on L_1 loss and edge loss to focus on multiscale texture information, to improve rain removal and preserve details, and
- comprehensive experiments on four challenging

datasets, both synthetic and real-world, to show the superiority of our method, and an ablation study to demonstrate the effectiveness of our method.

2 Related work

The single image de-raining problem can be seen as removing rain streaks from a rainy image to recover the clean background layer. A rainy image O is abstracted as a combination of a clean background layer B and a rain streak layer R . A widely used mathematical formulation is thus:

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

This is a highly ill-posed problem. Hence, various approaches use prior information to constrain the solution space, and pay more attention to the desired characteristics of the restored image. Kang et al. [7] tackled single image rain removal as a decomposition problem. They used a bilateral filter to decompose the rainy image into high frequency and low frequency information, respectively. They then utilized dictionary learning and sparse coding to further decompose the high frequency part to produce a de-rained result. Chen and Hsu [4] designed a low-rank model to find spatio-temporally correlated rain streaks, which is less time-consuming. A Gaussian mixture model (GMM) is applied in Ref. [9] to provide patch-based priors which can accommodate diverse rain streaks. These methods can remove some rain streaks in captured images, but generally fail to remove all rain.

In recent years, a number of deep learning based single image de-raining methods have been proposed, achieving a better performance than prior-based de-raining methods. Ref. [5] proposed a deep detail network which uses a guided filter to get a high frequency detail layer as input and learns the negative residual to reduce the mapping between input and output. Yang et al. [12] designed a multi-task recurrent architecture to handle the problem of rain streak accumulation. Zhang and Patel [1] used estimated rain-density information to guide rain removal at different rain densities and scales. Ren et al. [2] proposed a simpler and more efficient rain removal network composed of residual blocks, a convolutional long short-term memory (LSTM) [17] unit. The recurrent architecture is used to

reduce parameters and maintain good performance. Yasarla and Patel [10] designed an uncertainty guided multiscale residual learning (UMRL) network to compute confidence maps to guide a de-raining network. Tan et al. [15] proposed a multiscale attentive residual network (MSAResNet) which is both location-aware and density-aware for detection and removal of rain streaks. Recently, Zhang et al. [18] designed a paired rain removal network for binocular cameras. This model can effectively extract and exploit semantic information and multi-view information. In comparison to single image de-raining methods, our proposed method can extract and aggregate features from multiple inputs to get more texture detail information.

We also now briefly review multiscale learning methods which inspired us to leverage multiple input features and our design of the hybrid multiscale loss. Previous de-raining methods [1, 12] use parallel dilated convolution with different strides to provide multiscale features and enlarge the receptive field. These methods utilize multiscale information at the feature level. In other image restoration tasks such as de-hazing [19, 20], coarse-to-fine networks exploit multiscale knowledge. Ren et al. [19] proposed a multiscale fusion network and derived three inputs from the original hazy image using white balance, contrast enhancement, and gamma correction. In Ref. [20], a multiscale network and a multiscale loss function were designed for image de-hazing. Motivated by this work, we designed a hybrid, multiscale loss guided, multiple feature fusion de-raining network. We not only leverage multiscale information at the feature level, but also focus on multiscale outputs by optimizing the hybrid loss function. Taking advantage of multiple inputs, our network is able to recover images with rich details.

3 Multiscale loss guided multi-feature fusion network

3.1 Overview

Considering the variety of rain streaks, we propose a hybrid multiscale loss guided multiple feature fusion de-raining network (MSGMFFNet) to use features from multiple inputs to facilitate rain streak removal. The architecture of MSGMFFNet is illustrated in Fig. 3(a); it consists of four modules:

(1) a learning module to generate an attention map which focuses on the rain streaks and texture detail information around them, (2) a multiple input feature extraction module to extract features from three inputs separately guided by the attention map, (3) a multiple feature fusion module to compute weight maps for blending the features of multiple inputs, and (4) a reconstruction module to restore the original resolution and predict the final de-rained image. The attention map learning module produces an attention map which shows the significance of different regions. Then we use contrast enhancement and gamma correction to provide two further inputs with better global visibility. To utilize the information concerning distribution of rain streaks, we concatenate the three inputs and the attention map predicted by the attention map learning module as inputs, and feed them into the multiple input extraction module to obtain features from the multiple inputs. The multiple feature fusion module estimates weights for the three inputs for blending the multiple features automatically. The reconstruction module is designed to remove rain streaks and to better restore the

original clean image from a rainy image. A hybrid multiscale loss composed of L_1 loss and edge loss is added to further enhance the performance of de-raining and ensure good detail preservation.

3.2 Attention map learning module

As explained in Section 1, the quality of images taken on rainy days is degraded by rain streaks and rain accumulation. To deal with rain streaks, we should pay different attention to rainy and non-rainy regions, to avoid over de-raining and loss of background texture information. On the other hand, treating rainy areas the same as non-rainy areas would result in rain streaks remaining in the recovered images. Thus, learning the distribution of rain streaks is important. Visual attention mechanisms [21] can automatically learn areas of interest; under their guidance, local features are more effectively extracted for downstream tasks. Thus, we design an attention map learning module which can automatically locate regions covered by rain: see Fig. 3(a). A recursive computation [2, 22, 23] utilizing the same module is introduced to reduce the number of parameters while

Fig. 3 Architecture of MSGMFFNet. (a) Key stages are an attention map learning module, a multiple input feature extraction module, a multiple feature fusion module, and a reconstruction module. The attention map learning module predicts an attention map showing the significance of different regions. Guided by the attention map, the multiple input feature extraction module obtains multiple features. Weight maps are computed in the multiple feature fusion module to fuse these features automatically and boost the performance of rain removal in the reconstruction module. (b) Architecture of the MSABlock. (c) Architecture of the MSRBBlock. DF: dilation factor. @: concatenation.

maintaining excellent performance.

Thus, the proposed attention map learning module is a recursive network formed by unfolding the same architecture. A convolution-ReLU layer is designed to extract shallow features. As different images have different rain streaks, and even the same image can contain various kinds of rain streak, as shown in Fig. 4, we build a multiscale aggregation block (MSABlock) to extract multiscale features to better capture different rain streaks. Figure 3(b) shows details of the MSABlock. It is made up of three dilated convolution layers and two normal convolution layers. We set the dilated factors to 1, 2, and 3 respectively to enlarge the receptive field. To further boost the flow of different scale features and reduce the loss of information, we concatenate all outputs of the dilated convolution layers and feed them into the next convolution layer. A convolutional gated recurrent unit (GRU) [24] follows link features in different stages. Finally, we build a convolution-sigmoid layer to generate the attention map with value in $[0, 1]$, larger values indicating greater importance. Figure 5 shows some generated attention maps. As can be seen,

they focus on the rain streaks. Under the guidance of the distribution of rain streaks, our method restores images with rich detail.

3.3 Acquisition of multiple inputs

Due to rain streaks and accumulation of rain, images captured in the rain face severe degradation. Inspired by de-hazing methods [25, 26], we find contrast enhancement can improve the global clarity of rainy images and coarsely eliminate the effect of rain accumulation. We take a linear combination of the input image I_{in} and average luminance of the image $\tilde{I}$:

$$\mathcal{I}_{\text{out}} = (\tilde{I} + 0.5)(1 - \alpha) + I_{\text{in}}\alpha \quad (2)$$

We set $\alpha = 1.5$ in practice. As shown in Fig. 2(b), this can remove some rain streaks and make the image globally clearer. However, some texture details are blurred in bright areas.

Because of the loss of texture detail in bright areas, we obtain another input by using gamma correction (a non-linear transformation):

$$\mathcal{I}_{\text{out}} = \beta I_{\text{in}}^{\gamma} \quad (3)$$

We set $\beta = 1$ and $\gamma = 1.5$; the gamma corrected image contains rich details in bright areas: see Fig. 2(c). In

Fig. 4 Rain streaks with appearances. In different images, rain streaks can be quite different in shape, and even in one image, they can have diverse densities, shapes, and scales. (a) Various rain streaks in different images. (b) Diverse rain streaks in one image.

Fig. 5 Attention maps generated by the attention map learning module. (a, c) Rainy images. (b, d) Corresponding attention maps.

summary, we use the original rainy image to ensure the restored image has correct colors, while the two derived inputs provide texture detail for further de-raining processes.

3.4 Multi-input feature extraction module

The multiple input feature extraction module aims to fully capturing the characteristics of the original rainy image and the two other inputs obtained by contrast enhancement and gamma correction. The three inputs are separately concatenated with the attention map generated by the attention map learning module and fed into three branches having the same weights. As shown in Fig. 3(a), each branch is made up of three parts: (1) a single convolution-instance normalization-ReLU layer to extract shallow features, (2) two strided convolution-instance normalization-ReLU layers to downsample the feature maps, to reduce computation, and (3) four multiscale residual blocks (MSRBlock) to enlarge the receptive field and capture multiscale characteristics to better extract rain streaks. Details of MSRBlocks are shown in Fig. 3(c). The dilation factors of the three dilated convolution layers are set to 1, 2, and 3 respectively. To further boost the flow of feature information, we add the input and output of the current layer and feed it into the next layer. A skip connection between the input to the MSRBlock and the output of the last layer is also provided. Outputs of the three branches are later fed into the multiple feature fusion module.

3.5 Multi-feature fusion module

We denote the above features extracted from the rainy image, contrast enhanced image, and gamma corrected image F_r , F_c , and F_g respectively. We follow Ref. [27], in which a gate structure is utilized to automatically calculate confidence scores of features for further feature fusion. In our multiple feature fusion module, firstly, we compute weight maps (W_r , W_c , W_g) which correspond to the importance of F_r , F_c , and F_g respectively. Then a linear combination is used to combine all products of these weight maps with the features of the multiple inputs as follows:

$$\mathcal{F}_f = W_r \otimes F_r + W_c \otimes F_c + W_g \otimes F_g \quad (4)$$

where $\mathcal{F}_f$ is the final fused result, further fed into the reconstruction module to predict the de-rained image. In practice, three inputs respectively are fed into a convolution-instance normalization-ReLU layer with a 1×1 filter. Then the outputs are concatenated as

input and fed in a convolution layer with a 3×3 filter to compute the weight maps. Finally, we multiply the weight maps and corresponding input features and use linear combination to get the final fused features $\mathcal{F}_f$. Example weight maps for F_r , F_c , and F_g are shown in Fig. 2. The weight map of F_c (Fig. 2(f)) shows that the region of interest is the global image, which is consistent with our intention to utilize contrast enhanced images to improve the global visibility. The F_g weight map (Fig. 2(g)) pays more attention in light areas which meets our idea of using gamma corrected images to provide more details in bright areas.

3.6 Reconstruction module

The multiple feature fusion module produces the fused features $\mathcal{F}_f$ and feeds them into the reconstruction module to increase the spatial resolution and generate the final de-rained image. The reconstruction module consists of two ResBlocks [28], two deconvolution-instance normalization-ReLU layers and a convolution-tanh layer: see Fig. 3(a). During training, a convolution-tanh layer is used after the first upsampling operation to obtain an image at 0.5 scale of the input size for further computing the hybrid multiscale loss.

3.7 Loss for the multiscale loss guided multi-feature fusion network

In the attention map learning module, we use mean squared error (MSE) to compute the difference between the attention maps predicted by different stages and the binary mask M . As the attention map learning module is a recurrent architecture, we denote the attention map produced at stage t by M_{A_t} . We calculate M based on a widely used mathematical representation of rainy images [9, 13, 29]. Firstly, the ground truth B is subtracted from the rainy image R . Secondly, the experience threshold is set; we use 30. Values larger than the threshold belong to the rainy region, others to the non-rainy region. The loss function of the attention map learning module is

$$\mathcal{L}_A = \sum_{t=1}^{N_1} t\theta \mathcal{L}_{\text{MSE}}(M_{A_t}, M) \quad (5)$$

where N_1 represents the number of recurrences. $t\theta$ indicates the importance of different stages. In practice, we set the N_1 to 4 and set θ to 0.2.

As Fig. 6 shows, rain streaks and texture details can be clearly observed in the edge map, and non-

Fig. 6 Comparison between original images and edge maps. (a) Original rainy image. (b) Ground truth. (c, d) Edge maps of (a, b) respectively.

rainy and rainy regions can be easily distinguished by edge information. Original images contain a wealth of information, but edge maps focus on texture detail. Thus, we design a new hybrid multiscale loss which consists of L_1 loss and edge loss for the purpose of restoring the content of original images and preserving the texture detail. To generate the corresponding edge map, we build a convolutional Scharr filter layer, denoted $S(\cdot)$. We get O_1 (0.5 scale of the input) and O_2 (full scale of the input) from the reconstruction module. The hybrid multiscale loss is

$$\mathcal{L}_H = \sum_{i=1}^{N_2} \theta_i \left(\|O_i - B\|_1 + \|\nabla_x(S(O_i)) - \nabla_x(S(B))\|_1 + \|\nabla_y(S(O_i)) - \nabla_y(S(B))\|_1 \right) \quad (6)$$

where B is the ground truth. N_2 is the number of different scale outputs and θ_i is weights the importance of different scales. $\nabla_x(\cdot)$ and $\nabla_y(\cdot)$ are the derivative operations along the horizontal (x) and vertical (y) directions. We set N_2 to 2 and θ_1, θ_2 to 0.2, 0.8, respectively.

The other two loss functions utilized are perceptual loss [30] and SSIM loss [31] to further ensure visual quality. A pre-trained VGG-19 model is applied to capture features in the restored image and the corresponding ground truth. We denote the features captured from the i th ReLU layer by $P_i(\cdot)$ and we linearly combine outputs of different ReLU layers. The perceptual loss is defined as

$$\mathcal{L}_P = \sum_{i=1}^m \theta_i \|P_i(O_2) - P_i(B)\|_1 \quad (7)$$

where m is the number of ReLU layers and θ_i is the importance of the corresponding layer. Here, we set m to 5, and $\theta_1, \dots, \theta_5$ to $1/32, 1/16, 1/8, 1/4, 1$, respectively. The SSIM loss is used to ensure

structural similarity between the ground truth and the de-rained image, and is calculated as follows:

$$\mathcal{L}_{SSIM} = 1 - \text{SSIM}(O_2, B) \quad (8)$$

$$\text{SSIM}(O_2, B) = \frac{2\mu_{o_2}\mu_b + C_1}{\mu_{o_2}^2 + \mu_b^2 + C_1} \frac{2\sigma_{o_2b} + C_2}{\sigma_{o_2}^2 + \sigma_b^2 + C_2} \quad (9)$$

where μ_{o_2} and μ_b are the mean value of the restored image O_2 and the ground truth B , respectively. $\sigma_{o_2}^2$ and σ_b^2 are the variances of O_2 and B , respectively. σ_{o_2b} represents the covariance. The two small constants C_1 and C_2 are used to prevent division by zero.

The overall objective function during training is

$$\mathcal{L} = \lambda_A \mathcal{L}_A + \lambda_H \mathcal{L}_H + \lambda_P \mathcal{L}_P + \lambda_{SSIM} \mathcal{L}_{SSIM} \quad (10)$$

where each λ sets the importance of each loss term. In practice, the parameters $\lambda_A, \lambda_H, \lambda_P, \lambda_{SSIM}$ are set to 1, 8, 8, 12, respectively.

4 Experiments

4.1 Approach

In this section, we describe experiments on our proposed MSGMFFNet method with three synthetic rainy datasets, a real-world rainy dataset and a task-driven evaluation dataset. Using the synthetic datasets, comparative qualitative and quantitative evaluations are conducted. Because each rainy image has a corresponding ground truth clean image in the synthetic datasets, PSNR [32] and SSIM [33] are used as full-reference image quality assessment metrics for quantitative rain removal performance comparison. On account of the difficulty of obtaining ground truth for real world images, we only show de-rained images for qualitative evaluation. A task-driven evaluation dataset is utilized to compare performance of an object detection task before and after rain removal.

Following Ref. [34], we use mean average precision (mAP) results to compare de-raining methods. We compare our proposed MSGMFFNet to four state-of-the-art methods.

4.2 Datasets

Previous de-raining work has published some synthetic datasets [1, 12, 14, 34]. In this paper, we chose the DID-MDN dataset provided by Ref. [1] as our training dataset, Train. The Train dataset contains 12,000 synthetic rainy images with diverse shapes, scales, and densities of rain streaks. Our evaluation uses three synthetic test sets provided by Refs. [1, 14]. The DID-MDN test dataset is composed of two subsets with 1000 and 1200 images, which we denote Test1 and Test2, respectively. The Rain800 test dataset from Ref. [14], denoted Test3 here, was chosen as our third test dataset; it has 100 synthetic rainy images with rain streaks of differing shape and density.

Real-world rainy datasets are provided by Refs. [12, 14]; we use them as our real-world rainy dataset when evaluating the performance of our proposed method. Most images of this real-world rainy dataset came from the Internet. Images of different scenes again contain diverse rain streaks.

Many high-level computer vision tasks are severely affected by rain. The RIS dataset captured by traffic surveillance cameras in rainy weather [34] was used to study the benefit of de-raining algorithms in an objection detection task. The RIS set contains 2048 relatively low resolution images taken by traffic cameras; it is annotated with object bounding boxes. Since 24 images lack label information, we used the remaining 2024 images as our test set to study the effectiveness of rain removal algorithms in a real task.

4.3 Training

During training, we randomly cropped 256×256 image patches from the original rainy images and corresponding clean background images. Gamma correction and contrast enhancement were used to get the other two inputs. The Adam [35] optimizer was utilized to train the model. We set momentum parameters β_1, β_2 and weight decay to 0.5, 0.999, and 5×10^{-5} , respectively. The batch size was set to 4. We initialized the learning rate to 2×10^{-4} , and multiplied it by 0.5 and 0.2 at epochs 30 and 35, respectively. The training process ended after 46 epochs.

4.4 Comparison on synthetic datasets

Qualitative and quantitative evaluations on Test1, Test2, and Test3 were conducted to assess the performance of our proposed MSGMFFNet and the following single image rain removal algorithms: JORDER [12], DID-MDN [1], PReNet [2], and UMRL [10]. All methods predicted de-rained results during inferencing in the same test environment. We used published pre-trained models for the selected methods for fairness.

4.4.1 Qualitative evaluation

Figure 7 shows results for different methods on the synthetic dataset Test1. We can clearly observe that some methods such as Refs. [2, 12] do not remove rain streaks completely, while some approaches like Refs. [1, 10, 12] tend to over smooth texture details and reduce the clarity of restored images. In contrast, our proposed method removes rain better and retains the texture detail information better than other methods. For example, our method can completely remove rain streaks in shadowed areas of vegetation.

Figure 8 presents de-rained images for different methods tested on Test1, Test2, and Test3. It can be clearly seen that our method outperforms other methods in dealing with diverse rain streaks of different shapes, directions, and densities. A visual comparison demonstrates that our proposed method can remove rain streaks effectively as well as preserving contextual details well.

4.4.2 Quantitative evaluation

Table 2 lists the average PSNR and SSIM for different methods on Test1, Test2, and Test3. As can be observed, our approach outperforms the peer methods in terms of these indicators. This demonstrates the effectiveness of our proposed method which utilizes multiple input features and multiscale information.

4.5 Real-world rainy dataset

To verify the effectiveness of rain removal in real-world rain environments, we conducted experiments on the real-world rainy dataset. We present some results for visual comparison. Figure 9 shows restored results from images with diverse rain streaks. In particular, our method removes rain streaks on the black dress marked with a box in the second row of Fig. 9 while other methods leave some rain streaks

Fig. 7 Visual quality comparisons on images from Test1, showing restored results and local areas for two scenes: in each case, in order: input, and results from JORDER [12], DID-MDN [1], PReNet [2], UMRL [10], and our proposed MSGMFFNet.

there. The tattoo in the third row of Fig. 9 can be seen clearly after using our method while other approaches produce artifacts or leave rain streaks in restored images. The other state-of-the-art methods either tend to over de-rain causing image blurring, or under de-rain resulting in rain streaks remaining, while our method both removes rain streaks and preserves texture details.

4.6 Task-driven evaluation dataset

To further study the performance of de-raining methods, we ran our proposed algorithm and four other algorithms to preprocess the task-driven dataset taken on rainy days. Then, three state-of-the-art object detection models, Faster R-CNN [36], SSD-512 [37], and RetinaNet [38], were utilized detect objects: our goal was to use general detection

Fig. 8 Visual quality comparisons on images from Test1, Test2, and Test3. (a) Input. (b) DID-MDN [1]. (c) PReNet [2]. (d) UMRL [10]. (e) MSGMFFNet. (f) Ground truth.

methods not designed for rainy weather, to test detection performance after rain removal, not to retrain an object detection method suitable for rainy days. We used models trained on MS COCO provided

by MMLab Detection Toolbox [39]. We used mean average precision (mAP) to compare the object detection results.

Table 1 presents the mAP detection results for

Table 1 Detection results (mAP) on task-driven evaluation dataset

Method	Rainy	JORDER [12]	DID-MDN [1]	PReNet [2]	UMRL [10]	MSGMFFNet
Faster R-CNN [36]	15.5	15.3	14.7	16.3	16.1	16.3
SSD-512 [37]	13.3	13.6	13.7	13.7	13.3	13.9
RetinaNet [38]	15.3	15.1	13.9	15.0	14.8	15.6

Table 2 Quantitative results (PSNR, SSIM) for different methods on the synthetic dataset

Method	Test1		Test2		Test3	
JORDER [12]	25.4771	0.7968	24.0653	0.7718	20.9817	0.7390
DID-MDN [1]	28.3183	0.8726	21.7603	0.6785	20.6418	0.6887
PReNet [2]	29.1923	0.8508	24.7702	0.7941	21.7710	0.7491
UMRL [10]	30.9506	0.9005	24.0687	0.7801	22.4755	0.7835
MSGMFFNet	31.4849	0.9027	26.0239	0.8166	22.7604	0.7888

Fig. 9 Visual quality comparisons on real-world rainy dataset. (a) Input. (b) JORDER [12]. (c) DID-MDN [1]. (d) PReNet [2]. (e) UMRL [10]. (f) MSGMFFNet.

single image rain removal methods using three different detection models. Figure 10 shows some restored images after rain removal and object detection. Due to over or under de-raining, restored images produced by Refs. [1, 2, 10, 12] contain artifacts which lead to lower mAP even than for the original rainy images. This shows that preserving texture detail during single image rain removal is of great importance. Because the images in this dataset were captured by relatively low resolution surveillance cameras, the dataset itself is very

challenging. Compared with other methods, our method achieves better results, but it still has a long way to go to provide the features that detection algorithms pay attention to, rather than human vision.

4.7 Ablation study

We now consider an ablation study to show the utility of the different components in our proposed MSGMFFNet. We focus on the following components: (1) the attention map learning module, f_{atten} , (2) the

Fig. 10 Task-driven evaluation dataset examples. (a) Rainy. (b) JORDER [12]. (c) DID-MDN [1]. (d) PReNet [2]. (e) UMRL [10]. (f) MSGMFFNet.

multiple inputs, f_{multi} , and (3) the hybrid multiscale loss function, L_H . We started with an architecture composed of an attention map learning module, a multiple input feature extraction module, and a reconstruction module, denoted $f_{\text{atten}} + f_{\text{single}}$. We only used one branch in the multiple input feature extraction module as we only fed the original rainy image with an attention map into the following network. We then trained the network f_{multi} without guidance of the attention maps; it consisted of a multiple input feature extraction module, a multiple feature fusion module, and a reconstruction module. Next, we added an attention map learning module to the network f_{multi} , denoted $f_{\text{atten}} + f_{\text{multi}}$. Finally, we added a hybrid multiscale loss function L_H to $f_{\text{atten}} + f_{\text{multi}}$ to get the final model $f_{\text{atten}} + f_{\text{multi}} + L_H$. Four different architectures were evaluated on Test1. Note that when the L_H loss was not used, we used the L_1 loss with the same weight as L_H loss to compute the difference of the final restored image and the ground truth. Table 3 shows the results of the quantitative evaluation. Performance improves by 1.30 dB when using multiple input features. The guidance of the attention map improves the performance by 1.20 dB, and the hybrid multiscale loss function improves the restored results by 0.62 dB. Figure 11 presents some de-rained images for the four different networks. We can clearly see that the final network provides better rain removal. The quantitative and qualitative evaluations justify that the components designed in this paper contribute to the performance of the final network.

We further present an ablation study on the

proposed loss function to validate the parameter choices. The hybrid multiscale loss function we proposed has 4 parameters as described in Eq. (6): $N_2, \theta_1, \theta_2, \lambda_H$. Firstly, we discuss the effectiveness of the hybrid loss. To do so, we only change λ_H in Eq. (10). To facilitate observation of the effects of different losses, we calculate the losses only at the final output scale. We replace L_H with just L_1 , single edge loss, denoted L_{edge} , and joint L_1 and edge losses in the ratio 1:1, denoted $L_1 + L_{\text{edge}}$. We also removed this item from the objective loss function. As can be seen in Table 4, $L_1 + L_{\text{edge}}$ achieves the best performance, demonstrating the effectiveness of the hybrid loss.

Next, we evaluated the de-raining results of MSGMFFNet when using different scales to calculate losses. During the reconstruction process, our MSGMFFNet has three scales: 1/4 scale, 1/2 scale, and full scale of the input size denoted as $\text{Scale}_{1/4}$,

Table 3 Ablation results for components of MSGMFFNet, on Test1

f_{atten}	f_{multi}	f_{single}	L_A	L_H	L_1	$L_{\text{SSIM}+L_P}$	SSIM	PSNR (dB)
✓	×	✓	✓	×	✓	✓	0.8873	29.5562
×	✓	×	×	×	✓	✓	0.8854	29.6506
✓	✓	×	✓	×	✓	✓	0.8966	30.8601
✓	✓	×	✓	✓	×	✓	0.9027	31.4849

Table 4 Average PSNR/SSIM of de-raining results of MSGMFFNet using different losses, on Test1

L_A	L_{SSIM}	L_P	L_1	L_{edge}	SSIM	PSNR (dB)
✓	✓	✓	×	×	0.8915	30.2140
✓	✓	✓	✓	×	0.8966	30.8601
✓	✓	✓	×	✓	0.8986	31.1110
✓	✓	✓	✓	✓	0.9020	31.1980

Fig. 11 Visual quality comparison for different architectures, on Test1. (a) Input. (b) $f_{\text{atten}} + f_{\text{single}}$. (c) f_{multi} . (d) $f_{\text{atten}} + f_{\text{multi}}$. (e) $f_{\text{atten}} + f_{\text{multi}} + L_H$. (f) Ground truth.

Scale_{1/2}, and Scale₁. We evaluated models trained using Scale₁ lone, mixed Scale_{1/2}, and Scale₁ in the ratio 1:1, and mixed Scale_{1/4}, Scale_{1/2}, and Scale₁ in the ratio 1:1:1 to compute the hybrid loss. We also set the sum of weights in each case to 1 to ensure that this item had the same order of magnitude in all three experiments. Table 5 shows that the PSNR values of results generated by MSGMFFNet using multiscale outputs to compute the hybrid loss are much higher than without multiscale outputs; the values of SSIM are similar. We note that using all 3 scales does no improve the performance, so we simply use $N_2 = 2$ to compute the hybrid loss.

We also investigated choice of θ_1 and θ_2 in Eq. (6) to trade-off the effects of Scale_{1/2} and Scale₁. We conducted three experiments in which the ratio $\theta_1 : \theta_2$ was 1:1, 1:4, and 4:1 respectively. We set $\theta_1 + \theta_2 = 1$ to ensure that the value of this loss item had the same order of magnitude in all three experiments. Table 6 shows the results. As the weight of θ_2 increases, we obtain better results. However, when $\theta_2 = 1$ and $\theta_1 = 0$, it reduces to only using Scale₁ to compute the hybrid loss, and the performance decreases as shown in Table 5. Hence, we simply fix θ_2 to 0.8 and θ_1 to 0.2 in this paper.

Finally, we considered selection of λ_H in Eq. (10). λ_H is the weight determining the importance of the hybrid multiscale loss in the total loss function. During training, we wish to this loss to roughly have the same order of magnitude as the SSIM loss and the perceptual loss. We conducted three experiments with λ_H set to 4, 8, and 12, respectively. Results are presented in Table 7. With increasing λ_H , MSGMFFNet achieves better results. However, when

the weight of the proposed hybrid multiscale loss equals the weight of the SSIM loss, the performance decreases. Thus, we set $\lambda_H = 8$ in this paper.

5 Conclusions

In this paper, a hybrid multiscale loss guided multiple feature fusion de-raining network (MSGMFFNet) was proposed, to detect rainy regions automatically and use features from multiple inputs to provide a better de-rained image. Multiple inputs are obtained from the original rainy image by using gamma correction and contrast enhancement to enhance global visibility and provide more detailed information. To blend features of multiple inputs automatically and improve the final de-rained images, weight maps are determined. In addition, a hybrid multiscale loss combining L_1 loss and edge loss was designed to focus on content and texture information to further boost the rain removal performance. In practice, we build a convolutional Scharr filter layer to obtain the edge map flexibly. Results of extensive experiments demonstrate that MSGMFFNet outperforms several leading methods both on synthetic and real-world rainy datasets.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under No. 2017YFB1003000, the National Natural Science Foundation of China under No. 61872047 and No. 61720106007, the Beijing Nova Program under No. Z201100006820124, the Beijing Natural Science Foundation (L191004), and the 111 Project (B18008).

References

- [1] Zhang, H.; Patel, V. M. Density-aware single image de-raining using a multi-stream dense network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 695–704, 2018.
- [2] Ren, D. W.; Zuo, W. M.; Hu, Q. H.; Zhu, P. F.; Meng, D. Y. Progressive image deraining networks: A better and simpler baseline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3937–3946, 2019.
- [3] Chen, D. Y.; Chen, C. C.; Kang, L. W. Visual depth guided color image rain streaks removal using sparse

Table 5 Average PSNR/SSIM of deraining results of MSGMFFNet using different scales to compute the hybrid loss, on Test1

Scale ₁	Scale ₁ +Scale _{1/2}	Scale ₁ +Scale _{1/2} +Scale _{1/4}
31.1980/0.9020	31.3489/0.9012	31.3175/0.9010

Table 6 Average PSNR/SSIM of deraining results of MSGMFFNet using different values of θ_1 and θ_2 , on Test1

$\theta_1 = 0.8, \theta_2 = 0.2$	$\theta_1 = 0.5, \theta_2 = 0.5$	$\theta_1 = 0.2, \theta_2 = 0.8$
30.9720/0.8994	31.3489/0.9012	31.4849/0.9027

Table 7 Average PSNR/SSIM of deraining results of MSGMFFNet with different values of λ_H , on Test1

$\lambda_H = 4$	$\lambda_H = 8$	$\lambda_H = 12$
30.9986/0.8993	31.4849/0.9027	30.7227/0.9009

- coding. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 24, No. 8, 1430–1455, 2014.
- [4] Chen, Y. L.; Hsu, C. T. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: *Proceedings of the IEEE International Conference on Computer Vision, 1968–1975*, 2013.
 - [5] Fu, X. Y.; Huang, J. B.; Zeng, D. L.; Huang, Y.; Ding, X. H.; Paisley, J. Removing rain from single images via a deep detail network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1715–1723, 2017.
 - [6] Fu, X. Y.; Huang, J. B.; Ding, X. H.; Liao, Y. H.; Paisley, J. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing* Vol. 26, No. 6, 2944–2956, 2017.
 - [7] Kang, L. W.; Lin, C. W.; Fu, Y. H. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing* Vol. 21, No. 4, 1742–1755, 2012.
 - [8] Li, X.; Wu, J. L.; Lin, Z. C.; Liu, H.; Zha, H. B. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11211*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 262–277, 2018.
 - [9] Li, Y.; Tan, R. T.; Guo, X. J.; Lu, J. B.; Brown, M. S. Rain streak removal using layer priors. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2736–2744, 2016.
 - [10] Yasarla, R.; Patel, V. M. Uncertainty guided multi-scale residual learning-using a cycle spinning CNN for single image de-raining. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8405–8414, 2019.
 - [11] Wang, T.; Yang, X.; Xu, K.; Chen, S. Z.; 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*, 12270–12279, 2019.
 - [12] Yang, W. H.; Tan, R. T.; Feng, J. S.; Liu, J. Y.; Guo, Z. M.; Yan, S. C. Deep joint rain detection and removal from a single image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1685–1694, 2017.
 - [13] 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*, 3397–3405, 2015.
 - [14] 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* Vol. 30, No. 11, 3943–3956, 2020.
 - [15] Tan, J.; Zhang, Y.; Fu, H. Y.; Ma, H. D.; Gao, N. Multi-scale attentive residual network for single image deraining. In: *Human Centered Computing. Lecture Notes in Computer Science, Vol. 11956*. Milošević, D.; Tang, Y.; Zu, Q. Eds. Springer Cham, 351–362, 2019.
 - [16] 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*, 633–640, 2013.
 - [17] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* Vol. 9, No. 8, 1735–1780, 1997.
 - [18] Zhang, K. H.; Luo, W. H.; Ren, W. Q.; Wang, J. W.; Zhao, F.; Ma, L.; Li, H. Beyond monocular deraining: Stereo image deraining via semantic understanding. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12372*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 71–89, 2020.
 - [19] Ren, W. Q.; Ma, L.; Zhang, J. W.; Pan, J. S.; Cao, X. C.; Liu, W.; Yang, M. Gated fusion network for single image dehazing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3253–3261, 2018.
 - [20] Ren, W. Q.; Pan, J. S.; Zhang, H.; Cao, X. C.; Yang, M. H. Single image dehazing via multi-scale convolutional neural networks with holistic edges. *International Journal of Computer Vision* Vol. 128, No. 1, 240–259, 2020.
 - [21] Wang, F.; Jiang, M. Q.; Qian, C.; Yang, S.; Li, C.; Zhang, H. G.; Wang, X.; Tang, X. Residual attention network for image classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6450–6458, 2017.
 - [22] Kim, J.; Lee, J. K.; Lee, K. M. Deeply-recursive convolutional network for image super-resolution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1637–1645, 2016.
 - [23] Tai, Y.; Yang, J.; Liu, X. M. Image super-resolution via deep recursive residual network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2790–2798, 2017.
 - [24] 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. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1724–1734, 2014.
 - [25] Ancuti, C. O.; Ancuti, C. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing* Vol. 22, No. 8, 3271–3282, 2013.
 - [26] Choi, L. K.; You, J.; Bovik, A. C. Referenceless prediction of perceptual fog density and perceptual image defogging. *IEEE Transactions on Image Processing* Vol. 24, No. 11, 3888–3901, 2015.

- [27] Chen, D. D.; He, M. M.; Fan, Q. N.; Liao, J.; Zhang, L. H.; Hou, D. D.; Yuan, L.; Hua, G. Gated context aggregation network for image dehazing and deraining. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 1375–1383, 2019.
- [28] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.
- [29] Huang, D. A.; Kang, L. W.; Yang, M. C.; Lin, C. W.; Wang, Y. C. F. Context-aware single image rain removal. In: Proceedings of the IEEE International Conference on Multimedia and Expo, 164–169, 2012.
- [30] Johnson, J.; Alahi, A.; Li, F. F. Perceptual losses for real-time style transfer and super-resolution. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9906*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 694–711, 2016.
- [31] Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.
- [32] HuynhThu, Q.; Ghanbary, M. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters* Vol. 44, No. 13, 800–801, 2008.
- [33] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* Vol. 13, No. 4, 600–612, 2004.
- [34] 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 Conference on Computer Vision and Pattern Recognition, 3838–3847, 2019.
- [35] Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [36] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 39, No. 6, 1137–1149, 2017.
- [37] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. C. SSD: Single shot multibox detector. In: Proceedings of the 14th European Conference on Computer Vision, 21–37, 2016.
- [38] Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, 2980–2988, 2017.
- [39] Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.-X.; Sun, S.-Y.; Feng, W.; Liu, Z.-W.; Xu, J. et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Huiyuan Fu received his Ph.D. degree in computer science from Beijing University of Posts and Telecommunications, China, in 2014. He is an associate professor at the School of Computer Science, Beijing University of Posts and Telecommunications. His research area includes visual big data, machine learning and pattern recognition, multimedia systems, etc. He received a Best Student Paper Award at ICME in 2016.

Yu Zhang is a master student in the Department of Computer Science, Beijing University of Posts and Telecommunications. Her research interests include machine learning and pattern recognition.

Huadong Ma received his Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), in 1995, his M.S. degree in computer science from Shenyang Institute of Computing Technology, CAS, in 1990, and his B.S. degree in mathematics from Henan Normal University, China, in 1984. He is a professor at the School of Computer Science, Beijing University of Posts and Telecommunications. His research interests include multimedia networks and systems, the Internet of things, and sensor networks. He has published over 300 papers in these fields.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.