Int. J. Appl. Math. Comput. Sci., 2023, Vol. 33, No. 4, 651–661
DOI: 10.34768/amcs-2023-0046

DENSEFORMER FOR SINGLE IMAGE DERAINING

TIANMING WANG [a,b], KAIGE WANG [a,b], QING LI [a,b,*]

[a]Intelligent Manufacturing Electronics Research Center
Institute of Microelectronics of the Chinese Academy of Sciences
3 Beitu Cheng West Road, Chaoyang District, Beijing 100029, China
e-mail: liqing@ime.ac.cn

[b]School of Integrated Circuits
University of the Chinese Academy of Sciences
3 Beitu Cheng West Road, Chaoyang District, Beijing 100029, China

Image is one of the most important forms of information expression in multimedia. It is the key factor to determine the visual effect of multimedia software. As an image restoration task, image deraining can effectively restore the original information of the image, which is conducive to the downstream task. In recent years, with the development of deep learning technology, CNN and Transformer structures have shone brightly in computer vision. In this paper, we summarize the key to success of these structures in the past, and on this basis, we introduce the concept of a layer aggregation mechanism to describe how to reuse the information of the previous layer to better extract the features of the current layer. Based on this layer aggregation mechanism, we build the rain removal network called DenseformerNet. Our network strengthens feature promotion and encourages feature reuse, allowing better information and gradient flow. Through a large number of experiments, we prove that our model is efficient and effective, and expect to bring some illumination to the future rain removal network.

Keywords: artificial intelligence, convolutional neural network, image deraining.

1. Introduction

Visual data play a crucial role in fields related to people's livelihoods and industries (Nowak *et al*., 2022; Chen *et al*., 2015; Kian Ara *et al*., 2023; Karlupia *et al*., 2023), such as autonomous driving, road monitoring, and drone aerial photography. Currently, deep learning-based algorithms are used in these fields to achieve object detection. In challenging weather conditions, the visual signals obtained by cameras may become distorted or damaged, especially in situations such as rain, fog, or snow. These conditions affect the visibility of the visual system, resulting in a decrease in the ability of object detection algorithms to detect key objects. In this article, we focus mainly on rain because it is the most common adverse weather condition. Rain is composed of droplets of various sizes and shapes, which hinder the reflection of light from objects in the scene and appear as rain streaks in the image. They can cause image details to be blurred, low contrast, and obstructed content, which may pose safety risks to systems that rely on visual data.

Using a rain removal algorithm and applying advanced visual algorithms to the generated rain-free images is the most naive and intuitive solution. However, if the rain removal algorithm model takes a long time to execute, this will impose an additional computational burden on rainy day object detection, directly limiting the practical application of rain removal algorithms. Therefore, it is crucial to propose an efficient and robust rain removal algorithm.

The process of image raining can be described as

$$I = B + R, \tag{1}$$

where B and R represent background layers and rain layers of the image, respectively, and I represents the rainy image obtained by the camera. Rain removal refers to separating B and R when obtaining I, which is an ill-posed problem. At the same time, the distribution of rain lines is very random, which makes rain removal still challenging.

*Corresponding author

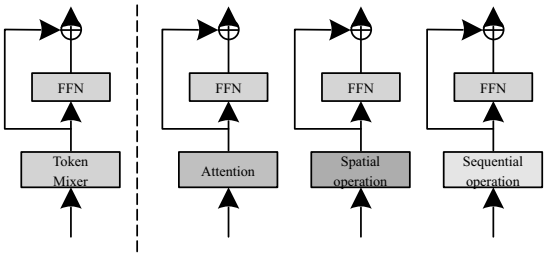

Fig. 1. Metaformer has a basic paradigm, which consists of two modules: a token mixer and a feed-forward network (FFN). Previous work used spatial operation as the token mixer. In this paper, we employ sequential operation in this role.

With the development of deep learning technology, the CNN architecture has been proven to be superior to traditional methods in the field of rain removal. Many modules have been proposed to form various rain removal networks, including the residual structure (He *et al*., 2016), the attention mechanism (Vaswani *et al*., 2017), the multi-stage structure (Zamir *et al*., 2021), the progressive structure (Ren *et al*., 2019), the U-Net architecture (Ronneberger *et al*., 2015), and so on. Despite the continuous improvement of rain removal performance, the structure is becoming more and more complex, and the amount of computations is also increasing. Around the same time, the Transformer structure has performed well in natural language processing. After the introduction of Vit (Dosovitskiy *et al*., 2021), it shines brightly in the visual field, completely changing the pattern of network architecture design. The self-attention module is the core component of the Transformer model. It provides a powerful way to get rid of the local limitation of convolution. Specifically, the attention mechanism calculates global autocorrelation matrix to model the global relationships between pixels. Therefore, it is also considered to be the key to the success of Transformer.

However, the Vit model has some problems when adopted as the network backbone. On the one hand, the calculation of the global attention matrix has quadratic complexity with the size of the input image, which brings a heavy computational burden and affects the operation efficiency. On the other hand, the key to success is still vague. Swin Transformer (Liang *et al*., 2021) and Volo (Yuan *et al*., 2021) introduce local attention, indicating that the neural network can still work well even without global attention. Further, MLP_Mixer (Tolstikhin *et al*., 2021), Poolformer (Yu *et al*., 2022) and Shiftvit (Wang *et al*., 2022) completely abandon the attention mechanism by some simple operations such as MLP, pooling, and shift to build Transformer deformation and have demonstrated encouraging performance. These models have one thing in common, that is, the basic units of them have a basic paradigm, which consists of two modules as shown in Fig. 1. The front module is called the token mixer, which is used to mix information, and the back module is a feed-forward network (FFN). Such a common basic unit is called Metaformer (Yu *et al*., 2022). The above deformation reveals that it seems that as long as the model adopts Metaformer as the general architecture, promising results can be obtained.

In this paper, we propose an efficient rain removal network, DenseformerNet, based on Metaformer. For the rain removal task, the model requires both location information to know where the rain line is, and semantic information to restore it. However, in the design of existing Metaformer, spatial operation has been paid enough attention, but the problem of sequential operation has not been well discussed. Sequential operation is of great significance. Here, the output feature maps of each layer in the network are regarded as a sequence. Rich representations that span levels from low to high are required for computer vision tasks. Even with the depth of features in a convolutional network, a layer in isolation is not enough. For example, the FPN (Belongie, 2017) adopts a multi-layer output scheme. The sequential operation realizes the fusion of sequence information, that is, combining and aggregating the depth features of each layer. In our design, we introduce sequential operations into Metaformer, which is the main difference from other Metaformer paradigm structures. Other operations such as pool, shift, and self-attention, as token mixers, are all operated on a single layer. We introduce a layer aggregation (LA) module as the token mixer, which can obtain the outputs of all previous LA modules as inputs, thus capturing the short- and long-range dependencies of different layer features. Sequential operation models the relationship between feature maps of different layers, and realizes semantic fusion to improve the inference of content and spatial fusion to improve the inference of location.

In addition, since our token mixer only aggregates information from previous layers but not processes, the FFN has to undertake more pressure for model performance. In our design, we introduce a scale-voting convolution (SVC) module to better capture local context. We explored the effects of different FFN designs on the performance of the model in ablation experiments.

In summary, the contributions of this work are as follows:

- We propose a complete rain removal model DenseformerNet based on the Metaformer paradigm, which has been proven to be an efficient structure for building neural network models.

- We propose an LA module to replace attention in Transformers, which can capture short- and

long-term dependencies of different layer features and allow better information and gradient flow.

- We carefully design the FFN and identify key factors of success for Metaformer in rain removal design.
- We conduct comprehensive experiments on various benchmark datasets to demonstrate the superiority of the proposed method. DenseformerNet achieves attractive results both on synthetic and real-world data sets.

The rest of the paper is organized as follows. Section 2 summarizes the relevant work in the past, and Section 3 introduces our methods in detail. Section 4 presents the analysis of the conducted experiments. Section 5 summarizes the main result of this paper.

2. Related work

In this section, we review some single image deraining methods and Metaformer structures.

2.1. Single image deraining. Single image deraining went through an evolutionary process of moving from model-driven to data-driven. Model-driven methods are subdivided into filter-based methods introduced by He *et al*. (2010) and Zheng *et al*. (2013), and prior knowledge-based methods represented by morphological component analysis (Kang *et al*., 2011), sparse coding (Luo *et al*., 2015), dictionary learning (Wang *et al*., 2017), and GMM prior knowledge (Li *et al*., 2016). However, the above methods have common drawbacks, including high computational complexity, long running time, and incomplete rain removal results.

With the proposal and rapid development of convolutional neural networks, data-driven methods have shown amazing results in various computer vision fields. Fan *et al*. (2017b) drew inspiration from ResNet and proposed a deep detail network to remove high-frequency rainfall content, and creatively presented a large-scale synthetic dataset consisting of rain and no-rain image pairs. Fan *et al*. (2017a) learnt the mapping between the rainfall image detail layer and the no-rainfall image detail layer directly from the data. PreNet (Ren *et al*., 2019) uses recursive computation and RCDnet (Wang *et al*., 2020b) proposes a model-driven deep neural network for this task. MPRnet (Zamir *et al*., 2021) proposes a multi-stage architecture to balance spatial details and high-level contextualized information. PHMNet (Yu *et al*., 2023) proposed a new blending and modulating HMM, and further adopts a multi-level refined module to refine the final deraining results. DGUNet (Mou *et al*., 2022) integrated a gradient estimation strategy into the gradient descent step of the proximal gradient descent (PGD) algorithm, driving it to deal with image degradation. Li *et al*. (2022) proposed a rain removal dataset SynRain-13k, and extensively evaluated the performance of rain removal models on it.

After the introduction of the Transformer architecture into the field of computer vision, Transformer models for single image rain removal have been rapidly developed in recent periods. SwinIR (Liang *et al*., 2021), Uformer (Wang *et al*., 2021) and IDT (Xiao *et al*., 2022) use a swin block to build rain removal network. Because Transformer is difficult to apply to large-resolution image restoration tasks, Zamir *et al*. (2022) proposed a more efficient attention module for image restoration called MDTA. Although the Transformer-based methods achieve better results, the reason is not clear.

2.2. Metaformer structure. The Metaformer structure consists of a token mixer and an FFN. Transformer and its variations adopt attention as the token mixer. Vit (Dosovitskiy *et al*., 2021) is a pioneering work that used Transformers for visual tasks and sparked Transformer research. MSViT (Fan *et al*., 2021) obtains multi-scale features by constructing a hierarchical attention layer. Swin Transformer (Liang *et al*., 2021) decomposes the image into local windows so that the amount of calculation is reduced from the quadratic complexity of the image size to linear complexity. Volo uses outlook attention to encode finer level features and contexts. Restomer (Zamir *et al*., 2022) calculates the covariance matrix between feature channels and implicitly models global relationships. In addition to the attention mechanism, MLP_Mixer (Tolstikhin *et al*., 2021) proved that using MLPs directly can also achieve good performance. Recently, some methods (Yu *et al*., 2022; Wang *et al*., 2022) have used zero-parameter operations such as pooling and shift as token mixers, and their performance is even better than Transformers when using the same number of parameters. Our method is based on the paradigm of Metaformer. Specifically, we use layer aggregation as the token mixer to mix features from different stages. We expect that our results can bring some inspiration to the design of the rain removal network.

3. Denseformer for single image deraining

In this section, we first introduce the overall structure of the rain removal network and then describe the core components of the proposed network: the Denseformer layer. Finally, we provide details on the training scheme and the loss function.

3.1. Rain removal network architecture. As shown in Fig. 2, the whole rain removal network architecture includes three parts: input projection, network backbone, and output projection. The whole model process is shown

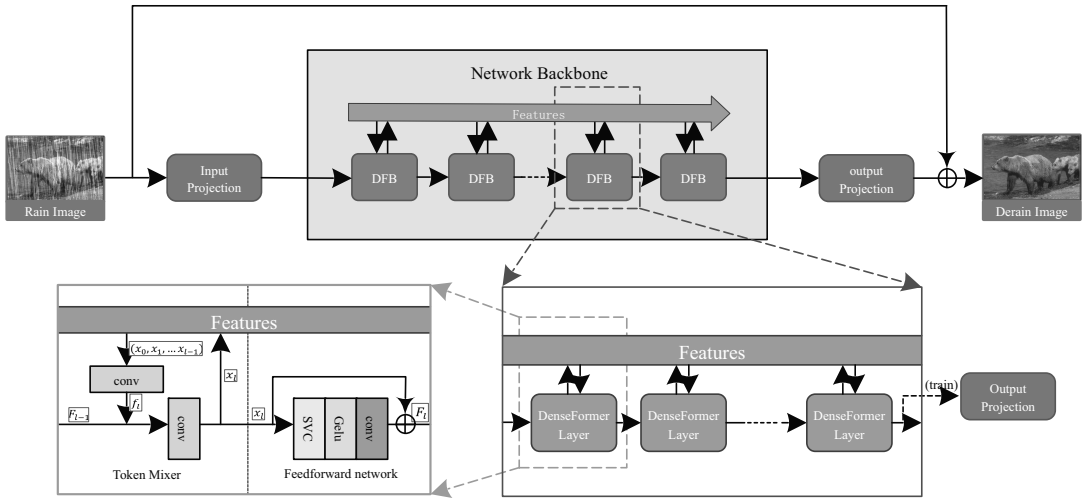

Fig. 2. Architecture of DenseformerNet for image deraining. DenseformerNet uses a single-scale pipeline incorporating efficient Denseformer blocks. The core modules of the network are input projection, network backbone, and output projection. Input projection converts the image to a feature space. The network backbone extracts the depth feature. Output projection restores depth features to images.

Algorithm 1. Model process pseudocode.

Input: O is a rainy image

Parameter: P_1 and P_2 represent input projection and output projection, respectively. $\text{DFL}_{i,j}$ represents the j-th DFL in the i-th DFB. K is the number of DFBs, and N_i is the number of DFLs in the i-th DFB.

Output: B is a derained image

1: Let $x = []$.
2: $f = P_1(O)$.
3: x.append(f)
4: **for** $i = 0$ to K **do**
5: **for** $j = 0$ to N_i **do**
6: $X, f = \text{DFL}_{i,j}(x, f)$
7: x.append(X)
8: **end for**
9: **end for**
10: $B = P_2(f) + O$
11: **return** B

as Algorithm 1. The entire algorithm is explained as follows: In the first line, the feature list is defined. In the second line, the input projection module is used to process the image. In the third line, the output from the previous step is added to the feature list. From the fourth to the ninth line, all the DFBs and their DFLs are traversed to process the features. The sixth and seventh lines indicate that the output of each stage will be added to the feature list. In the tenth line, the output projection module is used to restore the features to an image. Next, we will provide a detailed explanation of the specific functions and processing procedures for each module.

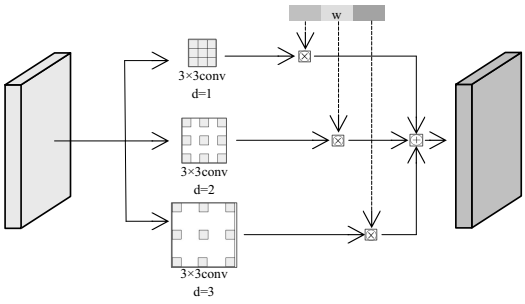

Fig. 3. Construction of the SVC module by combining multiple branches with different dilated convolution layers. Dilated convolution layers assign each branch with an individual receptive field. The weight of each branch in the final output will be controlled by a set of learnable parameters.

Input projection. For the input rainy image $I \in \mathbb{R}^{H \times W \times C_{\text{in}}}$ (H, W, and C_{in} are the image height, width, and input channel number, respectively), we use a convolution layer with both the kernel size and stride p and the number of channels C_{dim} to obtain the high-dimensional feature map $F_0 \in R^{\frac{H}{p} \times \frac{W}{p} \times C_{\text{dim}}}$ as

$$F_0 = conv\,(I)\ . \tag{2}$$

Input projection is very important for early visual

Fig. 4. Qualitative comparison with the state-of-the-arts on two randomly sampled files Rain100H.

processing. Although the resolution becomes $1/p$, it yields more stable optimization and better results. In our implementation, we use $p = 2$.

Network backbone. For the high-dimensional feature F_0 obtained by input projection, we will continue extracting the deep feature $F_i \in R^{\frac{H}{p}\times\frac{W}{p}\times C_{\text{dim}}}$ as

$$F_i = \text{DFB}\left(F_{i-1}\right), \quad i = 1, 2, \ldots, N, \tag{3}$$

where DFB is a Denseformer block and N is the number of DFB blocks in the network. Each DFB contains K_i Denseformer layers (DFL). More specifically, for the i-th DFB, the intermediate features are extracted block by block as

$$F_{i,\,0} = F_{i-1,\,K_{i-1}}, \tag{4}$$

$$F_{i,\,j} = \text{DFL}\left(F_{i,\,j-1}\right), \quad j = 1, 2, \ldots, K_i, \tag{5}$$

where $F_{i,\,j}$ means the output feature map of the j-th DFL in the i-th DFB.

Output projection. We get the feature sequence $(F_1, F_2, \ldots, F_N)$ in the backbone, and the output projection is responsible for restoring it to rain removal images. In reference, only F_N is restored. In order to introduce additional supervision, we restore each deep feature during training. For the i-th depth feature $F_i \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times C_{\text{dim}}}$, we use a set of convolution layers to convert its channel number to $C_{\text{in}} \times p^2$. Finally, PixelShuffle is directly used and combined with the residual structure to transform it into a rain removal image $B_i \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ as

$$B_i = PixelShuffle\left(convs_i\left(F_i\right)\right) + I. \tag{6}$$

3.2. Denseformer layer. The structure of the Denseformer layer is shown in Fig. 2. It consists of two parts: a token mixer and a feed-forward network. In our design, we use layer aggregation as a token mixer to combine past and current features.

The structure of layer aggregation is shown in Fig. 2. Firstly, it accepts the output of all LA in the past as the input, and combines them to aggregated feature $f_l \in R^{\frac{H}{p}\times\frac{W}{p}\times C_{\text{dim}}}$ through one 1×1 convolution layer as

$$f_l = conv_{l1}\left(concat\left(x_0, x_1, \ldots x_{l-1}\right)\right), \tag{7}$$

where $concat\left(x_0, x_1, \ldots x_{l-1}\right) \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times(l\times C_{\text{dim}})}$ represents the concatenation of feature maps generated from layers 0 to $l-1$ on the dimension of feature channel. Then f_l and the output of the previous Denseformer layer F_{l-1} are aggregated to $x_l \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times C_{\text{dim}}}$ as

$$x_l = conv_{l2}\left(concat((F_{l-1},\ f_l)\right). \tag{8}$$

The second part is FFN. Instead of using a linear projection layer, we use convolution layers to expand

Fig. 5. Qualitative comparison with the state-of-the-art on four randomly sampled real-world images.

the receptive field. Extracting features at different scales can improve the performance of visual tasks. In this paper, we designed a scale-voting convolution (SVC) module to learn features at different receptive fields. SVC can adaptively learn which parts are more effective in rainwater removal. The structure of our proposed module is shown in Fig. 3. SVC is a multi-branch convolution block. Its internal structure can be divided into two parts: a multi-branch convolution layer with different dilation rates and a set of learnable voting coefficients. The former part uses dilation convolution, which is responsible for generating feature maps of multiple receptive fields. The latter part learns a set of adaptive weights w to weight the feature maps of different receptive fields for voting. This process can be formulated as

$$X_l = \sum_{i=1}^{M} w_i conv_{d_i}\left(x_l\right) \ , \tag{9}$$

where w_i is a voting coefficient and M is the number of branches. In our implementation, we use $M = 3$ and the corresponding dilation rates are 1, 2, and 3, respectively. We set w_i as a set of learnable parameters after the softmax operation, so that the network can automatically select the appropriate scale in a certain position. We use a 3 × 3 convolution for subsequent processing and use GELU as the activation function. Our FFN is formulated as

$$F_l = conv_{l3}(GELU(X_l)) + \ x_l \ . \tag{10}$$

Compared with the vanilla version, our FFN has the ability to capture local information, which is what we lose in the token mixer.

3.3. Loss function. In the training process, each DFB will output the deep feature maps F. They will be restored to a rain removal image B. We adopt negative SSIM for each B, and use different coefficients controlled by hyper-parameters α. Consequently, the loss function of the network can be formulated as

$$L = -\sum_{i=1}^{N} \alpha_i \text{SSIM}\left(B_i, B_{\text{gt}}\right) \ , \tag{11}$$

where B_{gt} represents ground truth image. When $i = N, \alpha_i = 1$, otherwise $\alpha_i = 0.1$.

3.4. Training scheme. Our network uses PyTorch to complete training with a single NVIDIA 3090 GPU. In our experiment, all networks share the same training settings. The patch size is 112 × 112 and the batch size is 8. We use the Adam optimizer to train for 100 epochs, with an initial learning rate of 10^{-3}. At the 30th, 50th, and 80th epochs, the learning rate is multiplied by an attenuation of 0.2.

4. Experiments

4.1. Experimental settings. Datasets. We trained our model on the synthetic dataset Rain100H and tested it on both Rain100H and Rain100L datasets (Yang *et al*., 2017). The Rain100L dataset contains 100 images for testing. The images in this dataset have sparse rain streaks and relatively few types of rain streaks, corresponding to scenes with light rain. The Rain100H dataset contains 1800 images for training and 100 images for testing. The images in this dataset have dense rain streaks and complex types of rain streaks, which have a significant impact on image quality, corresponding to scenes with heavy rain. Training and testing are based on the official split. Additionally, we used the real-world dataset provided by Wang *et al.* (2020a), which comprises rainy images collected from the Internet. During testing, we randomly selected 100 images from this dataset.

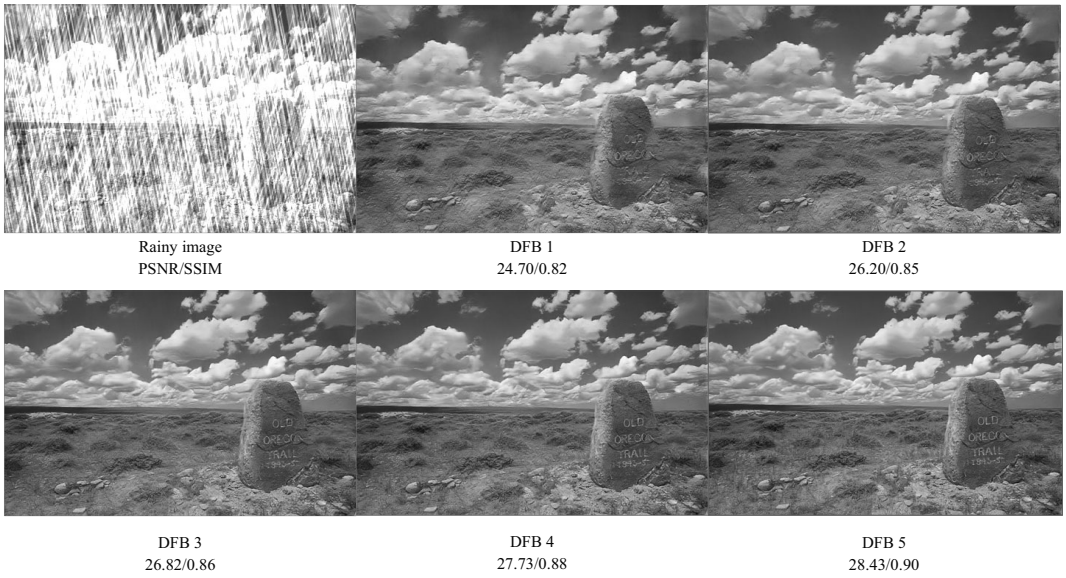

Fig. 6. Image results for different numbers of blocks.

Evaluation methods. We adopt the commonly used PSNR and SSIM indicators to evaluate the rain removal performance on the synthetic dataset. Following previous work (Zamir *et al*., 2021), we evaluate on the Y channel in the YCbCr color space. We also use NIQE (Mittal *et al*., 2012) to measure the performance of our model on the real-world data set. NIQE is a non-reference image quality score. The smaller the value, the higher the image quality. The test time is calculated at an image resolution of 256×256 with one Tesla V100 GPU.

Architecture variants. We designed three DenseformerNet variants: DenseformerNet-T (tiny), DenseformerNet-M (medium), DenseformerNet-L (large). Specifically, we set different channels and the number of DFBs and DFLs. The details are as follows:

- DenseformerNet-T: $C = 36$, depths = $[1, 1, 2, 2, 2]$,
- DenseformerNet-M: $C = 48$, depths = $[3, 3, 3, 3, 4]$,
- DenseformerNet-L: $C = 60$, depths = $[4, 4, 4, 4, 4]$.

4.2. Experimental results. We compare our models with excellent performance and commonly used structures GMM (Li *et al*., 2016), PreNet (Ren *et al*., 2019), RCDNet (Wang *et al*., 2020b), MPRNet (Zamir *et al*., 2021), and the Transformer structure Restomer (Zamir *et al*., 2022). For PreNet, MPRNet, and RCDNet, we retrained them with the same settings as DenseformerNet. For Restomer, we use the unofficial version (`https://github.com/leftthomas/Restormer`). We tested our model on synthetic datasets Rain100H and Rain100L and real-world datasets.

The quantitative results in Tables 1 and 2 indicate that DenseformerNet-L achieves the strongest performance compared with the current state-of-the-art architectures, both on synthetic and real-world datasets. As shown in Table 1, on the rain100H test set, DenseformerNet-L outperforms GMM, PreNet, RCDNet, MPRNet, and Restormer by 18.00 dB, 3.56 dB, 1.17 dB, 1.27 dB, and 1.09 dB in terms of PSNR, respectively. As shown in Table 2, DenseformerNet-L achieves the lowest NIQE score, indicating its superior performance on real-world datasets. In addition, DenseformerNet achieves optimal performance at a comparable running speed. For example, at a processing speed of approximately 40 ms, DenseformerNet-L outperforms MPRNet. These improvements demonstrate the effectiveness of building a rain removal model based on Denseformer. Table 1 also presents the results of different versions of DenseformerNet, which have varying numbers of parameters (from 0.83 M to 10.32 M) and different performance (PSNR increases from 29.76 dB to 33.05 dB). Specifically, the lightest version can achieve a running speed of over 100 fps.

Owing to our proposed network structure and loss function, DenseformerNet achieves the best results for all indicators, while the visualization in Fig. 4 illustrates that our network outperforms the rest of the networks in terms of rain removal. Specifically, DenseformerNet recovers the box area better, removes the rain lines more

Table 1. Quantitative results of different SID methods.

Model	Rain100H	Rain100L	Test time(ms)	Parameters(M)
GMM	15.05/0.425	28.66/0.865	1000+	-
PreNet	29.49/0.903	35.49/0.974	21	0.16
RCDNet	31.88/0.915	37.15/0.977	83	2.98
MPRNet	31.78/0.932	38.12/**0.984**	37	3.64
Restormer	31.96/0.916	38.41/0.982	69	26.10
DenseformerNet-T(w/o SVC)	29.76/0.913	35.69/0.977	**8**	0.83
DenseformerNet-T(w SVC)	30.05/0.915	35.91/0.977	14	1.58
DenseformerNet-M(w/o SVC)	31.01/0.924	37.59/0.981	16	3.07
DenseformerNet-M(w SVC)	31.37/0.937	37.69/0.982	28	5.73
DenseformerNet-L(w/o SVC)	32.64/0.943	38.28/ **0.984**	23	6.13
DenseformerNet-L(w SVC)	**33.05/0.948**	**38.50/ 0.984**	40	10.32

Table 2. NIQE of different models on real-world dataset.

Models	GMM	PreNet	RCDNet	MPRNet	Restormer	DenseformerNet-L
NIQE(↓)	5.01	4.13	4.18	4.12	4.11	**4.09**

effectively, and preserves the image background more realistically. Figure 5 shows our test results on real-world datasets. We can see that our method can remove the rain lines more cleanly and restore more details.

4.3. Ablation study and analysis. We conducted extensive ablation experiments to explore the impact of each component of our model on the rain removal performance. Unless specified, the experiments in this section adopt common parameters and training strategies on DenseformerNet-M. We demonstrated PSNR and SSIM on Rain100H.

Input projection. Input projection projects the image from the RGB space to a feature space by a patch-embed operation,which is adopted in many visual tasks to divide the image into small patch. These small blocks are still part of the original image, and the pixel values have not been changed. Table 3 shows the results with different patch sizes. When the patch size is 2, the PSNR is higher by 1.23 dB and 0.53 dB compared with patch sizes of 1 and 4, respectively. If the patch size is 1, the model will consume more computing resources, but lead to performance degradation. This shows that input projection is very important for early visual processing, which brings more stable optimization and better results.

Aggregation mode. Table 4 shows the effects of different aggregation methods on rain removal performance. Here $\times$ indicates that there is no aggregation method. We directly used a 1×1 convolution layer in a token mixer. We also made a comparison with the another aggregation method: LSTM (Graves, 2012; Shi *et al*., 2015). LSTM introduces a gate mechanism to control the flow and loss of features. In our implementation, all token mixers in each block are a common LSTM, and all LSTMs share the hidden state. In addition, we conducted experiments on spatial aggregation, including the pool and shift mentioned in the first section. It can be seen that dense connection outperforms the methods of no aggregation, LSTM, pooling, and shift by 1.44 dB, 0.11 dB, 2.78 dB, and 0.33 dB in terms of PSNR, respectively, indicating the importance of sequence aggregation in feature processing, and that a dense connection is the most effective method for temporal aggregation.

FFN. Previous work (Wang *et al*., 2021) indicated that introducing locality into the FFN is more suitable for image restoration compared with the token mixer, so we designed different FFNs and showed the results in Table 5. Vanilla FFN consists of two 1×1 convolution layers. It can be seen that using a vanilla FFN will lead to serious performance degradation. This is because the whole network will completely lose the ability to capture local features. We also compared SVC with single receptive field methods. The experimental results show that our voting strategy outperforms using a single receptive field methods by at least 0.26 dB, indicating that widening the receptive field at each stage is advantageous for image deraining.

Output of each block. Figure 7 shows the SSIM and PSNR values of the B_i $(i = 1, 2, 3, 4, 5)$. We can see that the output of a later layer may lead to a higher PSNR. This is due to the benefit of dense connection method, which allows the later layers to directly obtain the desired features. Figure 6 visualizes this process, and it can be seen that the texture details are constantly improving.

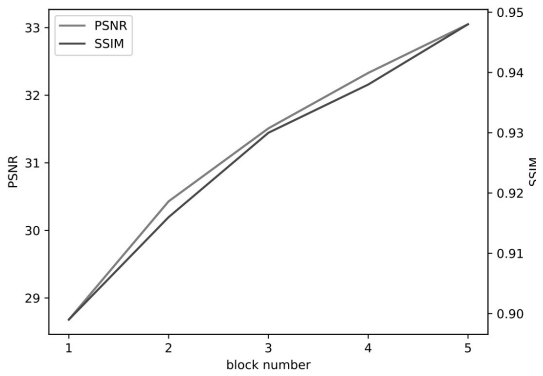

Fig. 7. Results of the SSIM and PSNR for different numbers of blocks.

Table 3. Results with different patch sizes.

Patch size	PSNR	SSIM
1	30.14	0.915
2	31.37	0.937
4	30.84	0.910

Table 4. Effects of different aggregation methods on rain removal performance.

Aggregation mode	PSNR	SSIM
×	29.93	0.907
LSTM	31.26	0.927
Pool	28.59	0.888
Shift	31.04	0.927
Dense	31.37	0.937

Table 5. Different FFNs and their results.

FFN	Dilation rate	PSNR	SSIM
Vanilla	1	18.76	0.740
Convolution 3×3	1	31.01	0.924
Convolution 3×3	2	30.79	0.923
Convolution 3×3	3	30.39	0.919
SVC	(1,2,3)	31.37	0.937

5. Conclusion

In this paper, we propose an effective rain removal model. The model is built based on the Metaformer structure, so it is very simple. Our model encourages the reuse of features in different stages, which effectively improves the performance. We carefully designed the network details, deeply discussed the impact of different methods on the performance, and designed a loss function to further enhance the rain removal effect of the model. Experiments show that our model has a good effect on synthetic datasets. We hope that our model can bring some new thoughts to the design of rain removal networks.

References

Belongie, T.L.D.G.H.H. (2017). Feature pyramid networks for object detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA*, pp. 2117–2125.

Chen, C., Seff, A., Kornhauser, A. and Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving, *Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile*, pp. 2722–2730.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale, *International Conference on Learning Representations*, pp. 1–21, (online).

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J. and Feichtenhofer, C. (2021). Multiscale vision transformers, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6824–6835, (online).

Fu, X., Huang, J., Ding, X., Liao, Y. and Paisley, J. (2017a). Clearing the skies: A deep network architecture for single-image rain removal, *IEEE Transactions on Image Processing* **26**(6): 2944–2956.

Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X. and Paisley, J. (2017b). Removing rain from single images via a deep detail network, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA*, pp. 3855–3863.

Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, Berlin, pp. 37–45.

He, K., Sun, J. and Tang, X. (2010). Guided image filtering, *European Conference on Computer Vision*, *Heraklion, Greece*, pp. 1–14.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA*, pp. 770–778.

Kang, L.-W., Lin, C.-W. and Fu, Y.-H. (2011). Automatic single-image-based rain streaks removal via image decomposition, *IEEE Transactions on Image Processing* **21**(4): 1742–1755.

Karlupia, N., Mahajan, P., Abrol, P. and Lehana, P.K. (2023). A genetic algorithm based optimized convolutional neural network for face recognition, *International Journal of Applied Mathematics and Computer Science* **33**(1): 21–31, DOI: 10.34768/amcs-2023-0002.

Kian Ara, R., Matiolanski, A., Grega, M., Dziech, A. and Baran, R. (2023). Efficient face detection based crowd density estimation using convolutional neural networks and an improved sliding window strategy, *International Journal of Applied Mathematics and Computer Science* **33**(1): 7–20, DOI: 10.34768/amcs-2023-0001.

Li, W., Zhang, Q., Zhang, J., Huang, Z., Tian, X. and Tao, D. (2022). Toward real-world single image deraining: A new benchmark and beyond, *arXiv:* 2206.05514.

Li, Y., Tan, R.T., Guo, X., Lu, J. and Brown, M.S. (2016). Rain streak removal using layer priors, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA*, pp. 2736–2744.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L. and Timofte, R. (2021). SwinIR: Image restoration using swin transformer, *Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, Canada*, pp. 1833–1844.

Luo, Y., Xu, Y. and Ji, H. (2015). Removing rain from a single image via discriminative sparse coding, *Proceedings of the IEEE International Conference on Computer Vision, Boston, USA*, pp. 3397–3405.

Mittal, A., Moorthy, A.K. and Bovik, A.C. (2012). No-reference image quality assessment in the spatial domain, *IEEE Transactions on Image Processing* **21**(12): 4695–4708.

Mou, C., Wang, Q. and Zhang, J. (2022). Deep generalized unfolding networks for image restoration, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA*, pp. 17399–17410.

Nowak, T., Nowicki, M.R. and Skrzypczyński, P. (2022). Vision-based positioning of electric buses for assisted docking to charging stations, *International Journal of Applied Mathematics and Computer Science* **32**(4): 583–599, DOI: 10.34768/amcs-2022-0041.

Ren, D., Zuo, W., Hu, Q., Zhu, P. and Meng, D. (2019). Progressive image deraining networks: A better and simpler baseline, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA*, pp. 3937–3946.

Ronneberger, O., Fischer, P. and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation, *International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany,* pp. 234–241.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K. and Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting, *Proceedings of the 2015 Neural Information Processing Systems (NIPS) Conference, Montreal, Canada*, pp. 802–810.

Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M. and Dosovitskiy, A. (2021). MLP-Mixer: An ALL-MLP, *Proceedings of the 2021 Neural Information Processing Systems (NIPS) Conference, NIPS2021*, pp. 24261–24272, (online).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017). Attention is all you need, *Proceedings of the 2017 Neural Information Processing Systems (NIPS) Conference, Long Beach, USA*, pp. 5998–6008.

Wang, C., Wu, Y., Su, Z. and Chen, J. (2020a). Joint self-attention and scale-aggregation for self-calibrated deraining network, *Proceedings of the 28th ACM International Conference on Multimedia, Seattle, USA*, pp. 2517–2525.

Wang, G., Zhao, Y., Tang, C., Luo, C. and Zeng, W. (2022). When shift operation meets vision transformer: An extremely simple alternative to attention mechanism, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2423–2430, (online).

Wang, H., Xie, Q., Zhao, Q. and Meng, D. (2020b). A model-driven deep neural network for single image rain removal, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, USA*, pp. 3103–3112.

Wang, Y., Liu, S., Chen, C. and Zeng, B. (2017). A hierarchical approach for rain or snow removing in a single color image, *IEEE Transactions on Image Processing* **26**(8): 3936–3950.

Wang, Z., Cun, X., Bao, J. and Liu, J. (2021). Uformer: A general U-shaped transformer for image restoration, *arXiv:* 2106.03106.

Xiao, J., Fu, X., Liu, A., Wu, F. and Zha, Z.-J. (2022). Image de-raining transformer, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(11): 12978–12995.

Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z. and Yan, S. (2017). Deep joint rain detection and removal from a single image, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA*, pp. 1357–1366.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J. and Yan, S. (2022). Metaformer is actually what you need for vision, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA*, pp. 10819–10829.

Yu, X., Zhang, G., Tan, F., Li, F. and Xie, W. (2023). Progressive hybrid-modulated network for single image deraining, *Mathematics* **11**(3): 691.

Yuan, L., Hou, Q., Jiang, Z., Feng, J. and Yan, S. (2021). Volo: Vision outlooker for visual recognition, *arXiv:* 2106.13112.

Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S. and Yang, M.-H. (2022). Restormer: Efficient transformer for high-resolution image restoration, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA*, pp. 5728–5739.

Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.-H. and Shao, L. (2021). Multi-stage progressive image restoration, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA*, pp. 14821–14831, (online).

Zheng, X., Liao, Y., Guo, W., Fu, X. and Ding, X. (2013). Single-image-based rain and snow removal using multi-guided filter, *International Conference on Neural Information Processing, Daegu, Korea*, pp. 258–265.

Tianming Wang received a BSc degree at Shandong University, China, in 2018. He is currently a PhD student at the Intelligent Manufacturing Center, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and artificial intelligence.

Kaige Wang received a BSc degree in engineering from the Harbin Institute of Technology, China, in 2020. He is currently a graduate student at the Intelligent Manufacturing Center, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and robotics.

Qing Li received his doctoral degree in engineering from Xidian University, China, in 2006. Currently, he is a professor in the Intelligent Manufacturing Center, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and the Internet of vehicles.

Received: 4 December 2022
Revised: 7 March 2023
Re-revised: 13 April 2023
Accepted: 19 April 2023