Circuits, Systems, and Signal Processing (2024) 43:1030–1052
https://doi.org/10.1007/s00034-023-02499-9

DeTformer: A Novel Efficient Transformer Framework for Image Deraining

Thatikonda Ragini[1] · Kodali Prakash[1] · Ramalingaswamy Cheruku[2]

Received: 15 April 2023 / Revised: 19 August 2023 / Accepted: 20 August 2023 /
Published online: 10 September 2023

Abstract

Captured rainy images severely degrade outdoor vision systems performance, such as semi-autonomous or autonomous driving systems and video surveillance systems. Consequently, removing heavy and complex rain streaks, i.e. undesirable rainy artifacts from a rainy image, plays a crucial role for many high-level computer vision tasks and has drawn researchers' attention over the past few years. The main drawbacks of convolutional neural networks are: have smaller receptive field, lack of model's ability to capture long-range dependencies and complicated rainy artifacts, non-adaptive to input content and also increase in computational complexity quadratically with input image size. The aforementioned issues limit the performance of deraining model improvement further. Recently, transformer has achieved better performance in terms of both natural language processing (NLP) and high-level computer vision (CV). We cannot adopt transformer directly to image deraining as it has the following limitations: (a) although the transformer possesses powerful long-range computational capability, it lacks the ability to model local features, and (b) to process input image, transformer uses fixed patch size; therefore, pixels at the patch edges cannot use local features of surrounding pixels while removing heavy rain streaks. To address these issues, in single image deraining, we propose a novel and efficient deraining transformer (DeTformer). In DeTformer, we designed a "gated depth-wise convolution feed-forward network" (GDWCFN) to address the first issue and applied depth-wise convolution to improve the modelling capability of local features and suppress unnecessary features and allow only useful information to higher layers. Also, the second issue was addressed, by introducing multi-resolution features in our network, where we applied progressive learning in the transformer, and thus, it allows the edge pixels to utilize local features effectively. Furthermore, to integrate the extracted multi-scale features and provide feature interaction across channel dimensions, we introduced a "multi-head depth-wise convolution transposed attention" (MDWCTA) module. The proposed network was experimented with on various derained datasets and compared with state-of-the-art networks. The experimental results show that DeTformer network

Extended author information available on the last page of the article

achieves superior performance compared to state-of-the-art networks on synthetic and real-world rain datasets.

Keywords Computer vision (CV) · Deraining transformer (DeTformer) · Gated depth-wise convolution feed-forward network (GDWCFN) · Image deraining · Multi-head depth-wise convolution transposed attention (MDWCTA)

1 Introduction

Various image restoration tasks such as dehazing [33], inpainting [1] and image deraining [17] can improve image quality, and this helps to improve the detection accuracy of high-level CV tasks such as object classification and detection [25]. Therefore, image deraining has grabbed a lot of attention of researchers in this low-level CV area. Although many traditional algorithms have been proposed to remove rain streaks from rainy images, it remains nevertheless complex and difficult, as there is no temporal information available in the captured images [39].

Consider a rainy image X, which can be expressed as a sum of rain layer R and background image B, and the physical model can be given as:

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

Therefore, still image deraining is an ill-posed problem, since we know only X, and there are lots of solutions for both unknown B and R. Most of the existing networks have considered and remained focused to remove rain streaks via the optimization problem. Therefore, existing deraining networks falls in any of the two categories such as traditional model-driven prior-based approaches and deep learning-based data-driven approaches.

Earlier researchers developed traditional model-driven prior-based approaches such as sparse coding [24], decomposition [17] and Gaussian mixture models (GMM) [20] to remove rain streaks from rainy images. However, these traditional networks are very sensitive to image variations as they designed their networks using handcrafted features. Due to wide growth and great improvements in deep learning technology, currently researchers have moved to adopt new data-driven approaches like convolutional neural networks (CNN) [42] and transformer [34] for removing rain streaks. Compared to traditional model-driven approaches, deep learning-based data-driven approaches are more robust and achieve excellent results. Currently most of the data-driven prior-based algorithms use CNN as their backbone to remove rain streaks. However, CNN has limited receptive fields and can capture only local spatial information and fail to capture broad contextual information.

To resolve this problem, some of the deraining networks introduced dilated convolution [18, 38] or construct deeper networks [8, 22, 37] to enlarge CNN receptive fields. However, it still results in local information as the operation of convolution is just sliding a window and computing local weighted summation. If multiple convolutional layers were stacked, it just increases the network complexity, which leads to overfitting.

In recent years, transformer [31] was initially used for NLP task and currently has been adopted in high-level CV tasks [25] and achieved impressive performance. CNN can only model local information, while transformer models the entire image and is adaptive to the input content. Tremendous success has been achieved in high-level CV tasks [9]; therefore, currently transformers have been adopted in low-level CV tasks such as dehazing and deraining networks [5, 28, 45]. U-shaped transformer [34] was proposed by Wang et al. by making refinements to Swin Transformer [23]. A nested U-shaped transformer [35] was proposed by increasing the number of transformer layers. However, we cannot adopt these transformers directly to single image deraining task as there are still many issues. (a) Transformer lacks the ability to model local features; (b) to process input image, transformer uses fixed patch size, therefore pixels at patch edges cannot use local features of surrounding pixels; and (c) hierarchical encoder was incorporated in U-shaped transformer; and it was unable to integrate multi-level features.

Therefore, we propose a deraining network based on transformer named DeTformer, in order to explore and exploit the long-range contextual information during the complex single image deraining process. We introduce multi-scale features in image deraining in order to effectively utilize the transformer fully. Therefore, it enables the transformer to use variable patch sizes and also helps to improve patch boundary defects. Several experiments illustrate that our network not only generates clean images, but also helps in improving the efficiency of subsequent high-level CV tasks.

The contributions can be summarized as follows:

1. A novel efficient transformer-based multi-scale structure was proposed for deraining single rainy images. Therefore, our network was able to model long-range inter-pixel contextual information in removing heavy and long rain streaks from rainy images.
2. We incorporated "gated depth-wise convolution feed-forward network" (GDWCFN) in DeTformer and it uses local features to generate better rain-free images.
3. We designed "multi-head depth-wise convolution transposed attention" (MDWCTA) module to integrate extracted multi-scale features effectively and also performs feature interaction along channels rather than spatial dimensions.
4. Experimental results demonstrate that DeTformer network outperforms SOTA networks on synthetic and real-world rain datasets.

2 Related Work

In this section, a brief review of deraining methods is provided and such networks fall under either traditional model-driven or deep learning-based data-driven prior-based approaches. Additionally, we provide previous works carried out by using multi-scale approaches and transformers.

2.1 Traditional Model-Driven Prior-based Approaches

Traditional model-driven prior-based approaches solve the image deraining process using prior knowledge. In [17], they decompose rain images into low- and high-frequency components, and adopted dictionary learning to remove high-frequency rain components. Li et al. [20] proposed Gaussian mixture models for single image deraining. Discriminative sparse coding [24] used learning the dictionary of rain streaks and background layers during the image deraining. Low-rank model-based traditional method was proposed by Chen et al. [7] to remove rain streaks and they assumed that rain streaks in a local patch have low rank. Filter-based sparsity and low-rank representation model was proposed by Zhang et al. [43] to remove rain streaks. In [32], a single image deraining model was proposed, which employed proximal gradient descent technique and applied convolutional dictionary learning mechanism for rain representation. Although all the tradition-based prior approaches have tried to achieve better results, they fail to remove rain streaks completely and cost time.

2.2 Deep Learning-based Data-Driven Approaches

Due to the wide success of deep learning in deraining [34, 42, 44], CNN-based approaches have replaced traditional model-driven prior-based approaches for removing rain streaks. Therefore, researchers have designed many CNN-based network structures and proposed various loss functions to improve the performance of deraining networks.

Wang et al. [38] proposed a deep learning architecture to remove rain streaks from heavy rains. They created a model which contains two components for representation of rain streak accumulation and for representation of various shapes and directions of overlapping rain streaks. In [42], a density-aware deraining network was proposed, which identifies the rain streak densities and processes these streak densities effectively. "Generative adversarial network (GAN)" [44] was designed to remove rain streaks and generate derained images directly. Fu et al. [10] introduced deep CNN referred to as Derain-Net to remove rain streaks. Wang et al. [40] adopted image enhancement technique for deraining process and incorporated GAN to generate high-quality rain patterns. To remove heavy rain streaks effectively, recursive networks [18, 26, 27] were adopted in deraining single images where the rain streaks were removed progressively and recursively.

Semi-supervised transfer learning technique [35] was adopted for single image deraining problem. This method uses semi-supervised and adds real rainy images without ground truth images into the network during training. Recursive operations were introduced on top of a progressive ResNet in order to exploit deep features across multiple stages and thus formed progressive recurrent network (PReNet) [27]. Yasarla et al. [41] proposed an over- and under-complete CNN which pays special attention while learning local structures by employing receptive field of filters. In [19], a rain-to-rain autoencoder was proposed and rain embedding was introduced in the encoder to improve deraining performance. They also proposed layered LSTM for recursive recurrent deraining and feature refinement was performed at multiple scales by a fine-grained encoder. Fu et al. [13] proposed rain streak removal via graph CNN to model long-range contextual information. Existing deraining networks embed low-quality features into the network directly, so Chen et al. [4] replaced low-quality features by high-quality features. They adopted closed-loop feedback control system to obtain latent high-quality features.

CNN-based deraining networks have achieved unprecedented success when compared to traditional model-driven prior-based networks. However, all the CNN-based deraining networks constructed by stacking multiple CNN layers and to model local information they use their limited receptive field.

2.3 Vision Transformers

Spectacular success has been achieved when the transformer was adopted in NLP field. Recently transformers [9] have been employed for image classification and achieved better results than SOTA CNNs. To learn long-range inter-pixel dependencies between the sequences, attention [31] mechanism was applied and the images were split into patch sequences by transformer. As transformer possesses long-range modelling capability and adaptability to input content, they were adopted in various high-level CV tasks such as object classification, detection, tracking, segmentation and pose estimation. For image restoration, networks which adopted the transformer are Restormer [46], U-former [34], Swin-IR [23], U2-former [16] and Transweather [30]. However, these networks perform poor on real rain images which are affected by high-density rainfall. In addition, to process high-resolution images, it requires huge computational complexity and also generate large number of parameters in transformer-based image deraining networks.

2.4 Multi-Scale Pyramidal Architecture

Using multi-scale learning, feature extraction would be improved to a certain extent since images of multiple scales can be extracted with different features. Lightweight pyramidal network [12] was developed using Gaussian–Laplacian image pyramid decomposition and performs image deraining at each pyramid-scale space. Jiang K et al. [15] constructed a pyramidal structure to improve the networks capability to encode rain streaks. Deep CNN-based recurrent neural networks [18] were constructed to remove heavy rain streaks. They adopted dilated CNN to acquire large receptive field since contextual information plays a vital role during the image deraining process. To remove heavy rain, they incorporated squeeze-and-excitation network and decomposed rain removal into multiple stages and assigned them with different alpha values.

In [26], a combination of multi-scale feature fusion and progressive structure was introduced in their network to separate heavy rain streaks. To extract contextual information from the shallow layers, they adopted U-net and at the last stage they incorporated image original resolution network to generate accurate derained images. A multi-stage architecture [47] was proposed which can progressively learn various image restoration functions for the degraded inputs. A supervised attention module was introduced to reweight local features by using per-pixel adaptive design. A "deep feature interactive aggregation network" [3] was proposed to improve long-range pixel dependencies among the captured features and to build channel correlations among the features for image deraining.

Therefore, by introducing transformer, multi-scale information was added so it can exploit the advantages of the network global connectivity and also learn feature map representation in rain streaks.

3 Proposed Method

Initially, efficient transformer architecture was described and then followed by a brief description of individual components used in our network. To reduce computational complexity of a single-scale network [23], we made key changes to multi-scale hierarchical module and multi-head SA layer. The overall pipeline of DeTformer architecture is shown in Fig. 1. A detailed description of core components of transformer block (TB) is as follows:

(a) "Multi-head depth-wise convolution transposed attention" (MDWCTA) module and

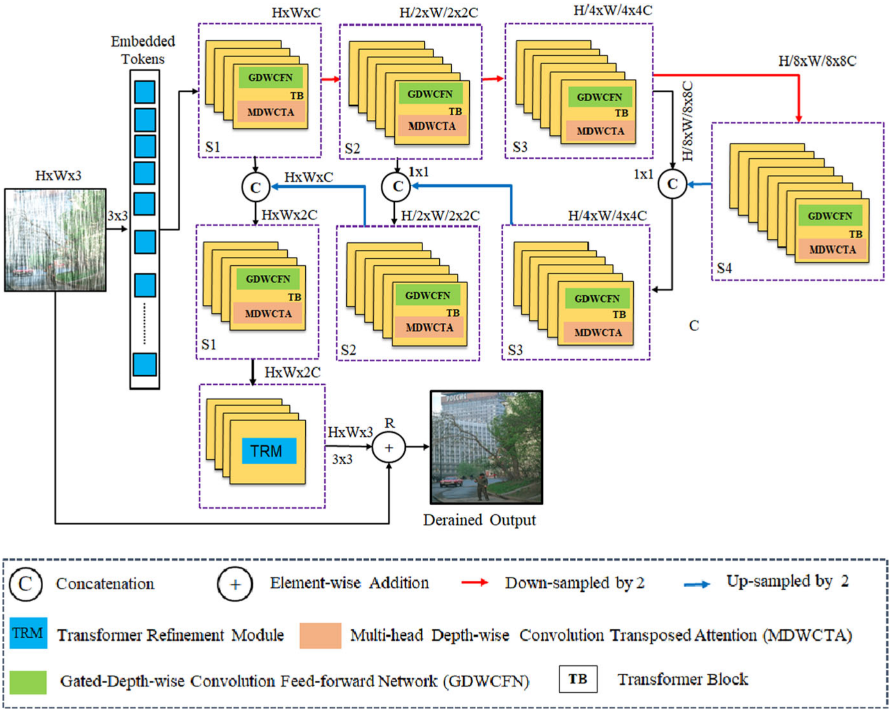

Fig. 1 Architecture of DeTformer

(b) “Gated depth-wise convolution feed-forward network” (GDWCFN). At the end, progressive training scheme and loss function details were provided.

First, the degraded rainy image ∈ $\Re^{\mathrm{H} \times \mathrm{W} \times 3}$ which is fed to a 3 × 3 convolution layer to obtain low-level features $\Re^{\mathrm{H} \times \mathrm{W} \times \mathrm{C}}$ (HW represents the spatial dimension and C represents the number of channels) and then flatten the extracted features into “token”. Next these tokens, i.e. shallow features, pass via four-stage symmetrical encoder–decoder and are then transformed into deep features $\Re^{\mathrm{H} \times \mathrm{W} \times 2\mathrm{C}}$. Each stage of encoder–decoder contains a series of transformer blocks (TB), and to maintain efficiency of our network, we gradually increase the number of transformer blocks from top to bottom levels. Therefore, our encoder network only expands the channel capacity and hierarchically reduces spatial dimensions for the input image. A 4 × 4 convolution with stride 2 was performed during the down-sampling operation; therefore, number of channels was doubled and the feature map became half. The decoder network takes the low-resolution latent features $\Re^{\mathrm{H}/8 \times \mathrm{W}/8 \times 8\mathrm{C}}$ and recovers progressively the high-resolution features. A 2 × 2 transposed convolution with stride 2 was performed during the up-sampling operation, so the number of channels reduces to half and the feature map becomes doubled.

We apply pixel-shuffled and pixel-unshuffled operations [28] for feature up-sampling and down-sampling. To make recovery process easier, we incorporated skip connections to concatenate encoder features with decoder features. After concatenation operation, we apply 1 × 1 convolution to make the number of channels become half at all stages, except at the top level. At stage 1, the low-level image features of encoder transformer block were aggregated with high-level features of decoder transformer block. Therefore, it helps to preserve the textural details and fine structures in output derained images. Now the deep features were enriched further in the refinement stage as it operates with high-spatial-resolution features. Finally, the refined feature map was fed to a 3 × 3 convolution layer to generate the residual feature map $R \in \Re^{\mathrm{H} \times \mathrm{W} \times 3}$ to which original rainy image X is added to reconstruct the derained image: $D = X + R$.

3.1 Transformer Block (TB)

Each transformer block (TB) consists of dual layer normalization layers [2], one “multi-head depth-wise convolution transposed attention” (MDWCTA) and one “gated depth-wise convolution feed-forward network” (GDWCFN) modules as shown in Fig. 2. Layer normalization (*LN*) was applied prior to MDWCTA and GDWCFN modules, and both modules perform element-wise addition using residual skip connections. It can be formulated as follows:

$$Feat_1 = MDWCTA(LN(Feat_0)) + Feat_0 \tag{2}$$

$$Feat_2 = GDWCFN(LN(Feat_1)) + Feat_1 \tag{3}$$

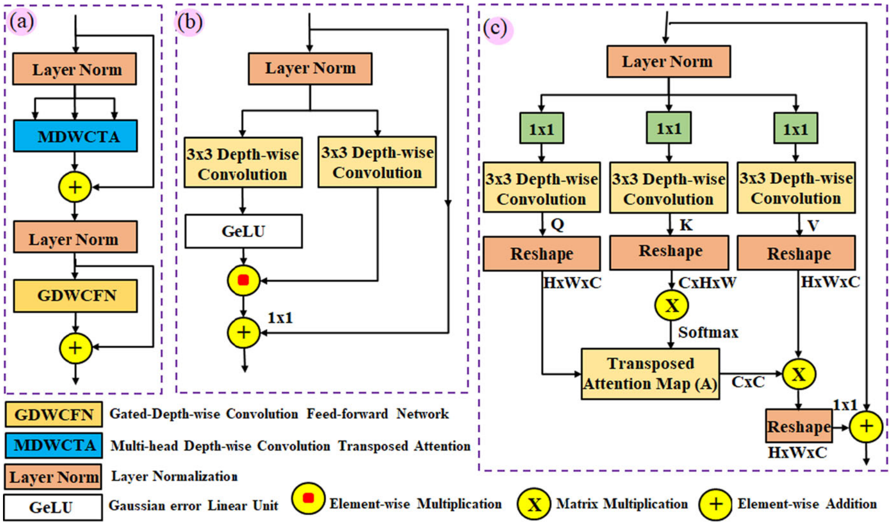
(a)
Layer Norm
MDWCTA
Layer Norm
GDWCFN
(b)
Layer Norm
3x3 Depth-wise Convolution
3x3 Depth-wise Convolution
GeLU
1x1
(c)
Layer Norm
1x1
1x1
1x1
3x3 Depth-wise Convolution
3x3 Depth-wise Convolution
3x3 Depth-wise Convolution
Q
K
V
Reshape
Reshape
Reshape
HxWxC
CxHxW
HxWxC
X
Softmax
Transposed Attention Map (A)
CxC
X
1x1
Reshape
HxWxC
GDWCFN Gated-Depth-wise Convolution Feed-forward Network
MDWCTA Multi-head Depth-wise Convolution Transposed Attention
Layer Norm Layer Normalization
GeLU Gaussian error Linear Unit
Element-wise Multiplication
X Matrix Multiplication
+ Element-wise Addition

Fig. 2 **a** Architecture of transformer block. **b** Gated depth-wise convolution feed-forward network. **c** Multi-head depth-wise convolution transposed attention

where *LN* refers to layer normalization, SA denotes self-attention, $Feat_0$,$Feat_1$ and $Feat_2$ denote input feature map of TB, output feature map of MDWCTA and GDWCFN modules, respectively.

The original transformer [9, 11] increases the computational complexity of the model as it globally calculates self-attention. We adopted "multi-head depth-wise convolution transposed attention" (MDWCTA) [24] in TB, in order to process high-resolution images while removing heavy rain streaks in single image deraining. Earlier works [21, 36] adopted transformer and proved that they are deficient in processing local contextual information. Therefore, we replaced feed-forward network (FFN) [9, 23] in TBs with the proposed "multi-head depth-wise convolution transposed attention" (MDWCTA) module. So, we compensate the transformers lack of capturing the local feature information with convolutional layers.

3.2 Multi-head Depth-wise Convolution Transposed Attention (MDWCTA)

The transformer computational burden increases mainly comes from the self-attention (SA) layer. In the original transformer [9, 11], the memory and time complexity of key–query dot product interaction increases quadratically with the spatial resolution of input, i.e. O(W^2H^2). Therefore, it is not quite feasible to apply SA on image deraining tasks as it often involves high-resolution images.

We proposed MDWCTA module which has a linear complex structure to resolve this issue as shown in Fig. 2c. In this module, they apply SA along the channel dimensions instead of spatial dimensions, i.e. cross-covariance is computed across the channels and generates attention map encoding the global context by default. One key change we made in this module was to introduce a 3 × 3 depth-wise convolution to highlight the local contextual information prior to the feature covariance computing in order to produce global attention map.

To reduce computational complexity burden in our network, and to perform self-attention, a "non-overlap window-based" technique was applied. On the input feature map F ∈ (HxWxC), the layer normalized tensor generates $(\mathrm{HW}/\mathrm{M}^2) \times \mathrm{C}$ local feature maps, as M × M local window slice was applied. Here $(\mathrm{HW}/\mathrm{M}^2)$ is the total divided windows. The obtained local features map was enriched as 1 × 1 convolution was applied to aggregate pixel-wise cross-channel contextual information. Then 3 × 3 depth-wise convolutions were applied to encode the channel-wise spatial contextual information, which yields normalized feature map, and the matrices for query (Q), key (K) and value (V) are given by:

$$Q = FX_Q P_{\mathrm{Q}}, \quad K = FX_K P_K, \quad V = FX_V P_V \tag{4}$$

where P and X perform 1 × 1 point-wise convolution and 3 × 3 depth-wise convolution. In the proposed network, we do not use bias in convolutional layers. A transposed attention map $A \in \Re^{\mathrm{C} \times \mathrm{C}}$ was generated instead of larger regular attention map $A \in \Re^{\mathrm{HW} \times \mathrm{HW}}$ by reshaping the query and key pair projections. The overall process of MDWCTA is formulated as follows:

$$Feat_1 = P \cdot \left(Attention\left(\hat{Q}, \hat{K}, \hat{V}\right)\right) + Feat_0 \tag{5}$$

$$Attention\left(\hat{Q}, \hat{K}, \hat{V}\right) = \hat{V} \cdot Softmax\left(\hat{Q}.\frac{\hat{K}}{\alpha}\right) \tag{6}$$

where $Feat_0$ and $Feat_1$ are input and output feature maps, and α is a learning scalable parameter which is used to control the magnitude of $\hat{Q} \cdot \hat{K}$ before applying softmax. In our module, the number of channels was divided into heads and it learns separate attention maps parallel which is similar to conventional multi-head SA [9].

3.3 Gated Depth-Wise Convolution Feed-forward Network (GDWCFN)

Figure 2b shows the GDWCFN architecture. A regular FFN [9] operates on each pixel separately and identically while transforming the image features. They used two 1 × 1 convolutions initially: one to expand feature channels and other to reduce the channels to get back the original image size. Therefore, we apply a nonlinearity function in hidden layers. To improve representation learning, we made two modifications to a regular FFN. One is that the gated mechanism was incorporated and the other one adopted was depth-wise convolution.

To perform element-wise product of two parallel paths of linear transformation layers, a gating mechanism was formulated, one of which was activated with nonlinear GeLU [14]. As in MDWCTA, we adopted 3 × 3 depth-wise convolutions in GDWCFN to encode information from the spatially neighbouring pixel positions, as it is useful to learn local image structures.

For an input tensor $Y \in \Re^{\mathrm{H} \times \mathrm{W} \times \mathrm{C}}$, GDWCFN was formulated as follows:

$$\hat{Y} = P \cdot Gating(Y) + Y \tag{7}$$

$$Gating(Y) = \mu\Big(W_d^1 W_p^1(LN(Y))\Big)\hat{e} W_d^2 W_p^2(LN(Y)) \tag{8}$$

where μ represents nonlinear GeLU function, $\hat{e}$ denotes element-wise multiplication and *LN* is layer normalization. The proposed network GDWCFN controls the information flow through multi-hierarchical levels and it allows each stage to put focus only on the fine details inverted to other stages. Therefore, this module plays a more vital role compared to MDWCTA module as its focus is to enrich the features with contextual information.

3.4 Progressive Learning

Many existing CNN-based deraining networks usually train networks using fixed image size patches. However, the original transformer model [29] trained on small cropped patches could not achieve optimal performance during image restoration. Therefore, we implemented progressive learning strategy where DeTformer network was trained initially with small image patches in the early epochs and gradually, patch sizes increased in later epochs. As we adopted mixed-size image patches training strategy, we were able to achieve better results during testing even for high-resolution images. Therefore, our network was able to preserve the fine image structures and texture while removing rain streaks as our network was trained using a curriculum learning fashion. We reduced the batch size as the patch size increased while training on large patches since it consumes longer time than usual. We needed to maintain similar time as fixed patch training.

3.5 Loss Function

In order to train deep draining networks, the widely adopted loss functions are mean absolute error (L_1) loss, mean square error (L_2) loss, negative SSIM loss, Charbonnier loss, attention loss, edge loss, adversarial loss and perceptual loss. We adopted Charbonnier loss in our network as it makes the model converge faster and can tolerate small errors. The total loss function is expressed as:

$$L = \sum_{S=1}^{4}\Big[L_{char}(X_S, Y) + \lambda L_{edge}(X_S, Y)\Big] \tag{9}$$

where X_S is the derained image, Y represents the ground truth and L_{char} denotes Charbonnier loss.

$$L_{char} = \sqrt{\|X_s - Y\|^2 + \in^2} \tag{10}$$

In addition, edge loss (L_{edge}) is defined as:

$$L_{edge} = \sqrt{\| \vartriangle (X_s) - \vartriangle (Y) \|^2 + \in^2} \tag{11}$$

where $\vartriangle$ is Gaussian operator which can control the relative importance of the loss terms in Eq. (9), λ (hyperparameter) was set to 0.05 and $\in$ constant was set to 10^{-3}.

4 Experimental Results and Discussion

Here we provide details of our experimental setup, datasets and performance metrics. We evaluated the performance and showed the effectiveness of DeTformer network on benchmark synthetic and real rain datasets.

(a) Experimental setup Our proposed network was implemented on PyTorch 1.7 deep learning framework. AdamW optimizer solution was applied during the network training and trained for 10^5 iterations. Fixed learning strategy was used with 3×10^{-4} learning rate. Batch size was set to 8, and adapted variable patch sizes are set to 128 × 128, 160 × 160 and 192 × 192, respectively. To make the proposed network more robust, various augmentation techniques were applied such as horizontal flip and vertical flip during the network training. In all TBs, window size was fixed to 8. All the experiments were carried on Google Colab pro + which has Tesla V100 GPU. We employed four-level encoder–decoder hierarchy, number of TBs used was (4, 6, 6, 8), number of channels used was (32, 48, 64, 192), number of attention heads used in MDWCTA was (1, 2, 4, 8) and TRM used 4 blocks.

(b) Datasets The effectiveness of our network was evaluated on synthetic paired rain datasets and real rain dataset [12], which includes Rain100L [38], Rain100H [38], Rain800 [44], Rain1200 [42], Rain12 [20] and Rain14000 [11] and renamed Testset as Rain100L, Rain100H, Test100, Test1200 and Test2800. Table 1 shows a brief summary of datasets used in this work.

Table 1 Summary of used datasets

Datasets	Train images	Test images	Test set renamed
Rain800 [44]	700	100	Test100
Rain14000 [11]	11,200	2800	Test2800
Rain1800 [38]	1800	0	NC
Rain100L [38]	0	100	Rain100L
Rain100H [38]	0	100	Rain100H
Rain1200 [42]	0	1200	Test1200
Rain12 [20]	12	0	NC
Total	13,712	4300	

(c) Evaluation Metrics To show the effectiveness and performance of DeTformer network, we evaluated the derained image quality using two evaluation metrics. “Peak signal-to-noise ratio” (PSNR) and “Structural Similarity Index Measurement” (SSIM) were calculated on the derained images. Generally, the larger their values are, the better the deraining effect is.

4.1 Comparison with the State-of-the-Art Networks

We compared the performance of DeTformer network comprehensively with several state-of-the-art (SOTA) deraining networks such as JORDER [38], DID-MDN [42], RESCAN [18], SSTL [29], PReNet [27], DerainNet [10], UMRL [40], MSPNet [15], SAPNet [45], SEMI [35], OUCD [41], ECNet [19], PMSDNet [26], RCDNet [32], DualGCN [13], MPRNet [46], RLNet [4] and DFIANet [3].

The visual quantitative results of DeTformer network on synthetic rain datasets are shown in Table 2. It is clear from the table that our network achieves superior performance over state-of-the-art (SOTA) networks on all synthetic datasets. In particular, on Rain100L and Rain100H datasets, DeTformer network obtains 38.99 and 31.45 dB PSNR which is + 3.79 and + 1.97 dB PSNR higher compared to DFIANet [3] and which clearly shows that our network removes heavy and complex rain streaks more effectively. Table 2 shows that DeTformer network achieves the highest PSNR and SSIM metric values on Rain100L, Rain100H, Test100, Test1200 and Test2800 synthetic datasets. These is due to the fact that our network uses the benefits of transformer as well models the long-range contextual information better.

The visual qualitative results of DeTformer network on synthetic rain datasets are shown in Figs. 3, 4 and 5, respectively. Although the networks (PReNet, ECNet and DFIANet) remove heavy rain streaks, “visible artefacts” and “blurred details” were nevertheless observed in the derained outputs, as shown in Fig. 3.

From the observation of derained images in Fig. 3, this situation occurs in clouds, sky and roof and appears in JORDER [38], RESCAN [18], SEMI [35] and DFIANet [3] networks. As the colour of background is similar to rain streaks, some networks perform excessive deraining and remove the fine details of similar colour as in the second row of Fig. 3. When the test images contain denser objects, it is difficult to remove rain streaks completely and recover finer details simultaneously, as was clear from the telephone booth and black fence in the third and fourth rows in SEMI [35], PReNet [27], ECNet [19], JORDER [38] and DFIANet [3]. OUCD [41] network combines global information in their network and pays attention only to local features and the network fails to remove heavy rain streaks completely. Therefore, compared to all these SOTA networks our network avoids these problems and restores the derained images which are highly similar to ground truth images.

Figures 4 and 5 show that our network exhibits impressive recovery deraining results while removing diverse light and heavy rain from rainy images. From the observed images, our network was able to restore clear image details and appropriate contrast and which are similar to ground truth images. Some more sample deraining results of the proposed network on Rain100H synthetic dataset along with their “mean square error” (MSE), PSNR and SSIM are shown in Fig. 6.

Table 2 Quantitative results of the proposed network on synthetic datasets and made comparison with the SOTA networks

Datasets	Year	Rain100L [12]	Rain100H [12]	Test100 [30]	Test2800 [8.]	Test1200 [34]	Avg. PSNR/
Metrics		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	Avg. SSIM
Networks							
JORDER [38]	2017	31.27/0.92	27.75/0.84	24.72/0.85	31.6/0.91	31.27/0.90	29.32/0.88
DIDMDN [42]	2018	25.23/0.74	17.35/0.52	22.56/0.82	28.13/0.86	29.65/0.90	24.58/0.77
RESCAN [18]	2018	29.80/0.88	26.36/0.78	25.01/0.83	31.29/0.90	30.51/0.88	28.59/0.85
SSTL [29]	2019	25.03/0.84	16.56/0.48	22.35/0.78	24.43/0.78	26.05/0.82	22.88/0.74
PReNet [27]	2019	32.44/0.95	26.77/0.85	24.81/0.85	31.75/0.91	31.36/0.91	29.42/0.89
DerainNet [10]	2017	27.03/0.88	14.92/0.59	22.77/0.81	24.31/0.86	23.38/0.83	22.48/0.79
UMRL [40]	2019	29.18/0.92	26.01/0.83	24.41/0.83	29.97/0.90	30.55/0.91	28.02/0.88
MSPNet [15]	2020	32.44/0.95	28.66/0.86	27.50/0.87	32.82/0.93	32.39/0.91	30.76/0.74
SEMI [35]	2019	22.25/0.84	18.08/0.57	20.72/0.68	24.38/0.73	23.91/0.71	21.87/0.70
OUCD [41]	2021	29.84/0.90	24.38/0.73	23.58/0.80	28.72/0.89	26.09/0.82	26.52/0.83
ECNet [19]	2022	33.42/0.95	27.91/0.86	27.55/0.88	32.42/0.93	30.05/0.90	30.27/0.90
PMSDNet [26]	2022	36.41/0.97	30.38/0.89	30.32/0.90	33.62/0.93	32.96/0.92	32.74/0.92
RCDNet [32]	2020	38.60/0.98	28.83/0.88	24.59/0.82	–	29.81/0.86	30.45/0.88
DualGCN [13]	2021	38.05/0.99	29.06/0.91	28.28/0.89	–	32.98/0.93	25.67/0.93
MPRNet [47]	2021	36.69/0.97	27.65/0.87	27.86/0.85	–	31.73/0.91	30.98/0.90
RLNet[4]	2021	37.38/0.98	28.87/0.90	27.95/0.87	–	32.62/0.91	31.70/0.91
SAPNet [45]	2022	34.77/0.97	29.46/0.89	29.13/0.88	32.18/0.93	32.46/0.91	31.60/0.91
DFIANet [3.]	2022	35.20/0.95	29.48/0.87	28.90/0.88	33.12/0.93	32.92/0.92	31.92/0.91
Proposed		**38.99/0.97**	**31.45/0.90**	**32.07/0.90**	**34.17/0.94**	**33.18/0.92**	**33.97/0.93**

Bold indicates the best results

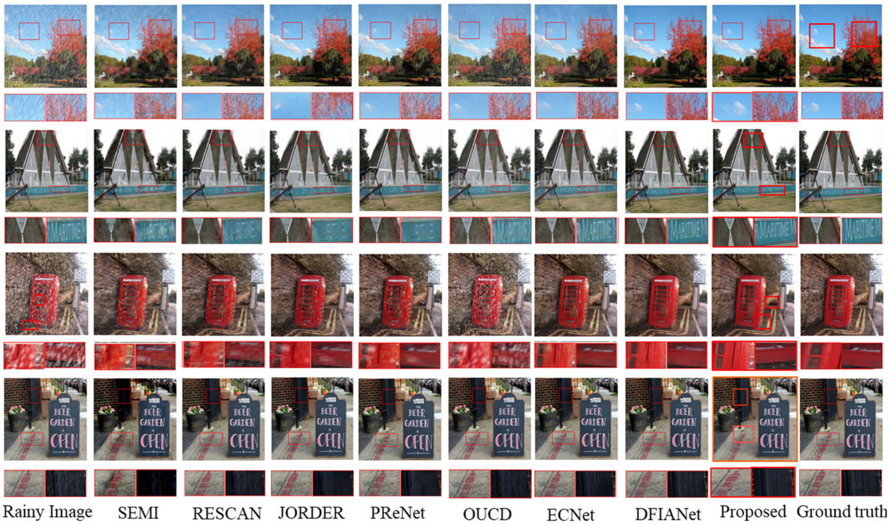

Fig. 3 Qualitative results of the proposed network on synthetic datasets and made comparison with the SOTA networks and DeTformer network generate the best visual results on synthetic datasets

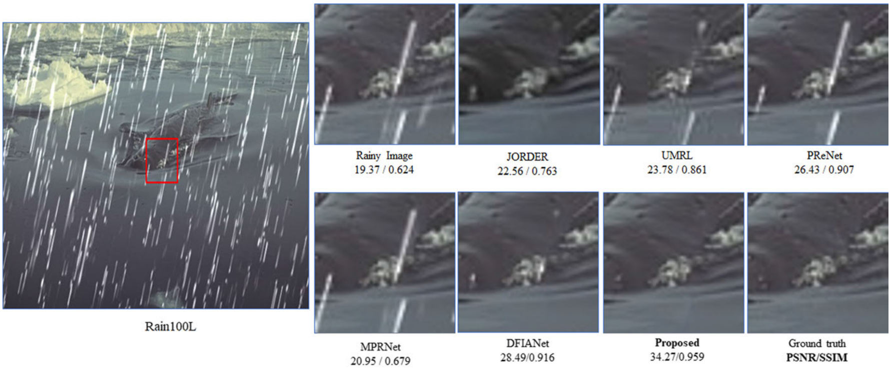

Fig. 4 Visual qualitative results of the proposed network on Rain100L synthetic dataset

To show the robustness and efficiency of DeTformer network, we also made a comparison with SOTA networks on real rain dataset [12]. Figure 7 shows the derained results on real rain dataset of the proposed network and made a comparative analysis with PReNet [27], MPRNet [47], ECNet [19], SAPNet [45] and DFIANet [3] networks. However, many of these networks produce artefacts during the image deraining process, which are not as clear as that of the images restored by our network. Our network removes rain streaks which are more unevenly distributed, and also achieves impressive performance while removing heavy rain streaks and outputs clear and detailed content results. In spite of complex rain scenes present in nature, our network generates excellent results while removing rain streaks under realistic conditions.

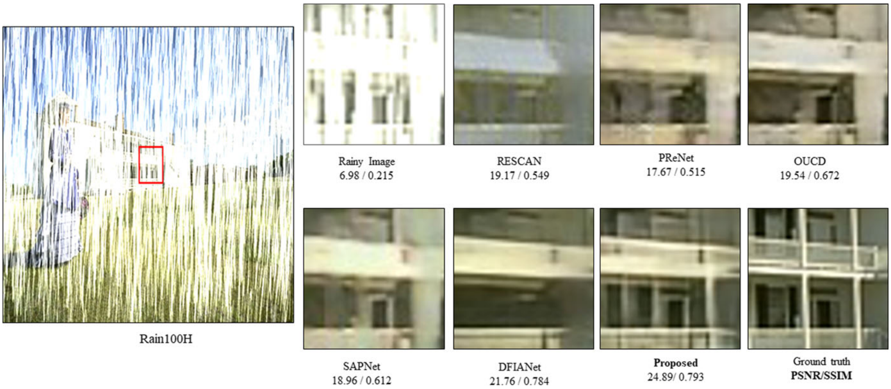

Fig. 5 Visual qualitative results of the proposed network on Rain100H synthetic dataset

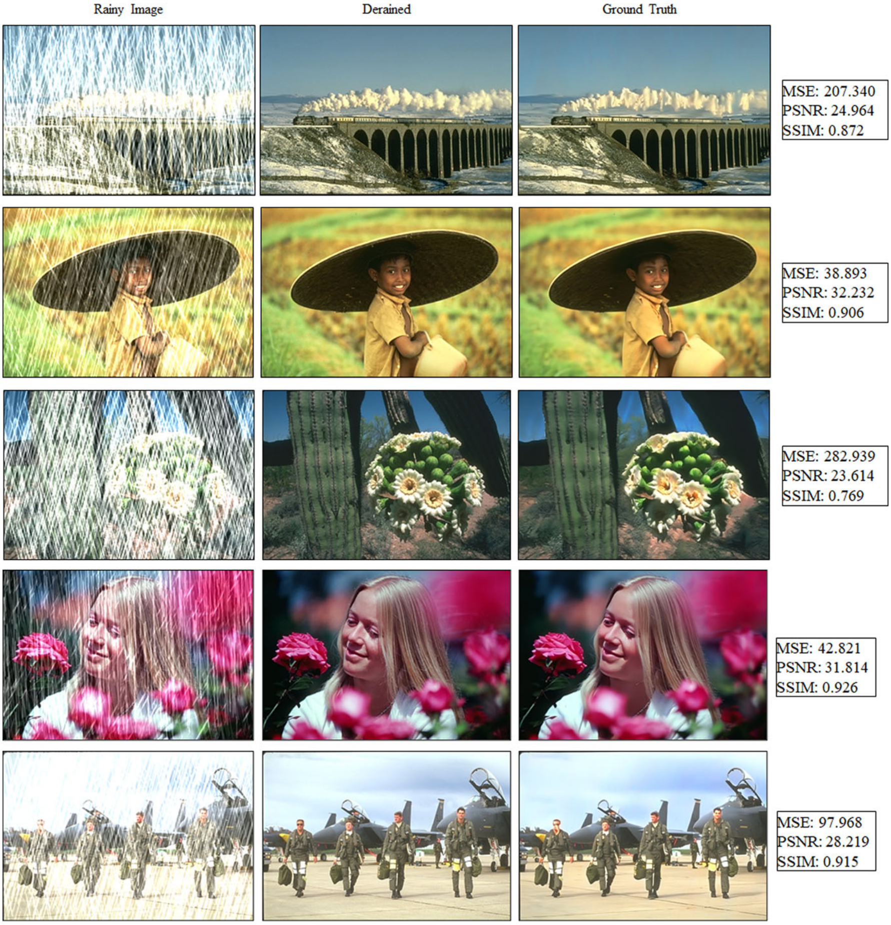

Fig. 6 Visual qualitative deraining results of some sample images of Rain100H

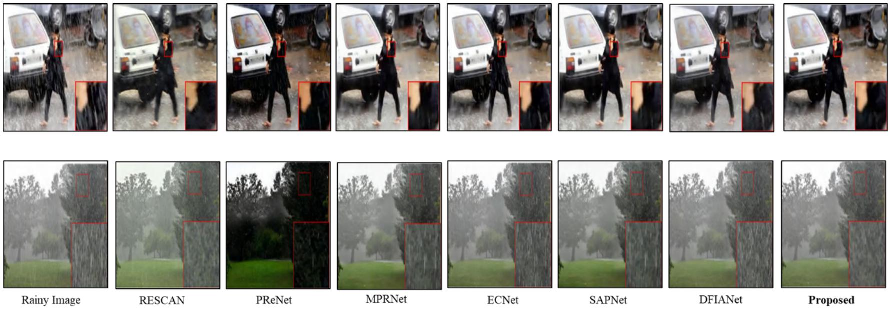

Fig. 7 Visual qualitative derained results on real-world rain dataset of our network and made comparison with SOTA networks

Table 3 Comparison of FLOPS, model parameters and runtime of SOTA networks

Network	FLOPS (G)	Parameters (M)	Runtime (ms)
JORDER [38]	26.76	0.37	254
DDN [11]	7.6	0.059	**12.36**
RESCAN [18]	32	0.15	52.7
DerainNet [10]	89.47	0.58	32
LPNet [12]	**3.57**	**0.007**	60
MSPNet [15]	594.6	21.52	322
PReNet [27]	66.54	0.16	90
RCDNet [32]	116.51	4.29	426.1
SAPNet [45]	92.2	0.283	138
DRDNet [8]	687.43	5.24	443.21
MPRNet [47]	175.8	28.46	1352.43
MPRNet—local [47]	176.9	28.46	1357.21
IPT [5]	34	115	386.8
Restormer [46]	174.5	26.13	287
HiNet [6]	293.79	26.59	412.68
Proposed	87.7	25.31	270

Bold indicates the best results

We also provided a number of parameters required and performed floating-point operations (FLOPS) on a specific Rain100H dataset and made comparison with the SOTA networks in Table 3. It is observed that the number of parameters in our network reduces, as general convolution was replaced by transformer. On a test image 256 × 256, our network runs for just 270 ms (ms) and generates noise-free image. Figure 8 shows the comparison of a number of model parameters, FLOPS generated and runtime (ms) on a 256 × 256 image resolution of various SOTA networks.

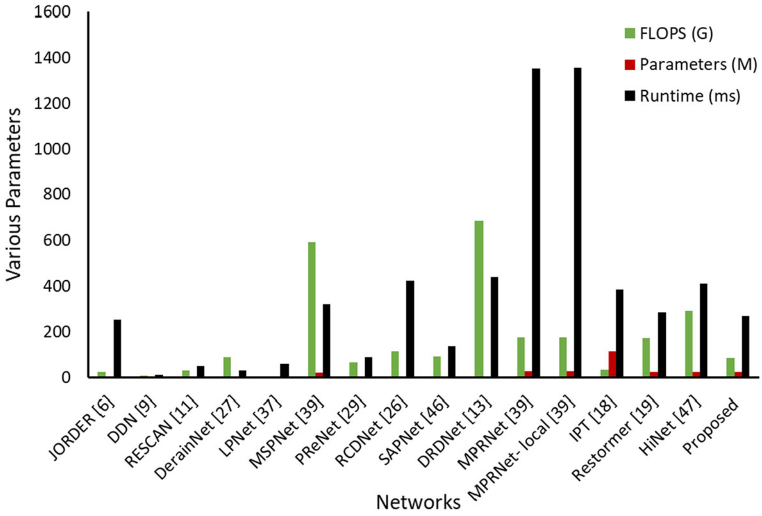

Fig. 8 Comparison results of SOTA networks vs. FLOPS, Parameters and Runtime

4.2 Ablation Studies

A series of ablation studies were conducted to show the impact of various factors on DeTformer network, and we evaluated the ability of our network during the deraining process. All ablation studies use Rain100H during network training and testing.

4.2.1 Effect of Basic Composition

Table 4 shows the ablation study results of the importance of each component separately. Therefore, our network achieves higher-quality performance. As seen from the table, when FFN was replaced with GDWCFN module, PSNR dropped by 0.88 dB. This proves the effectiveness of GDWCFN in enhancing and preserving the local feature information and alleviates the drawback of original transformer in extracting local feature information. If MDWCTA module is removed, PSNR drops by 1.31 dB

Table 4 Effects of basic composition in the proposed network

Component	PSNR	FLOPS (B)	Parameters (M)
Remove MDWCTA	30.14	83.7	24.86
Remove GDWCFN with FFN	29.57	85.3	25.02
Replace TB with CNN	27.86	**81.3**	**23.02**
Eliminate up- and down-sampling layers	28.93	83.4	24.53
Proposed structure	**31.45**	87.7	25.31

Bold indicates the best results

Table 5 Impact of number of scales in the proposed network

No. of scales	PSNR	SSIM
1	30.21	0.87
2	31.13	0.88
3	31.26	0.89
4	**31.45**	**0.9**

Bold indicate the best results

and this proved that the networks performance would be improved by multi-scale feature fusion. PSNR was drastically reduced by 1.52 dB, when all the up-sampling and down-sampling layers were removed and this shows the effectiveness of the designed U-shaped transformer structure. We also provided a number of required parameters required and performed floating-point operations (FLOPS) when a specific component was employed in the proposed network.

We also performed experiments on the number of scales to be employed in the encoder–decoder network structure for removing rain streaks effectively during the deraining process.

4.2.2 Effect of Number of Scales

Table 5 shows the impact of the number of scales to be employed in the proposed network and to show the effectiveness of multi-scale structure. From these observations, it is clear that when $S = 1$, PSNR drops by 0.24 dB, since multi-resolution features can assist the DeTformer network better to remove heavy and complex rain streaks effectively. When $S = 4$, we were able to achieve both higher PSNR and SSIM metric values.

4.2.3 Effect of λ Hyperparameter

From Eq. (9), the total weighted loss function depends on λ hyperparameter which was set to 0.05. In order to obtain better network performance, we performed an ablation study to fix λ parameter. Table 6 shows the influence of λ value on PSNR and SSIM values. So, from these observations, we fixed λ value as 0.05 in weight loss function as it achieves higher PSNR and SSIM.

Table 6 Impact of λ parameter on total loss function

λ	PSNR	SSIM
0	31.16	0.89
0.05	**31.45**	**0.9**
1	31.32	0.9
2	30.97	0.88

Bold indicate the best results

Table 7 Impact of N in the proposed network

N	PSNR	SSIM
1	31.26	0.89
2	**31.45**	**0.90**
3	32.94	0.91
4	33.27	0.91

Bold indicate the best results

Table 8 Effect of loss function for improving deraining performance

Loss function	PSNR	SSIM
L1	31.36	0.89
L2	31.39	0.89
Charbonnier	**31.45**	**0.90**

Bold indicate the best results

4.2.4 Effect of Number of Transformer Blocks in Encoder–Decoder Network

To decide the number of transformer blocks (N) to be employed in the encoder–decoder network, we performed an ablation study. Table 7 shows the impact of N on the proposed network on complexity and computational burden. In order to balance both complex structure and computational complexity, i.e. deraining performance and efficacy, we adopt $N = 2$ in our network.

4.2.5 Effect of Different Loss Functions in Our Network

An ablation study was conducted to show the effectiveness of Charbonnier loss, and make a comparison with other popular loss functions L1 and L2. Table 8 shows the effectiveness of Charbonnier loss, so we adopted this loss function to reconstruct the derained image.

4.2.6 Impact of Progressive Learning

The impact of progressive learning adopted in our network ablation study is shown in Table 9. We achieved better results with progressive learning than with fixed patch learning while still balancing similar training time.

Table 9 Impact of progressive learning on the proposed network

Patch size	PSNR	Train time (Hours)
Progressive (128^2 to 192^2)	**31.45**	23.4
Fixed (128^2)	31.33	**22.3**

Bold indicate the best results

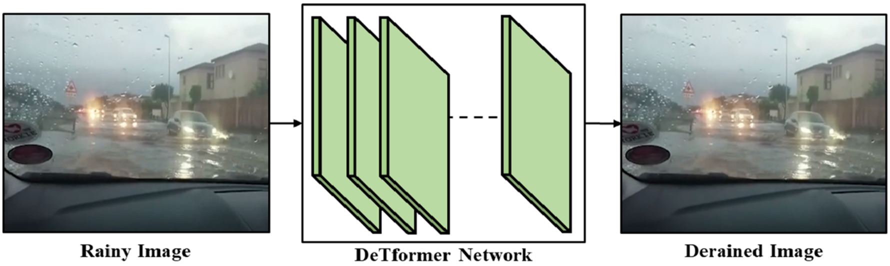

Fig. 9 DeTformer network failure scenario

4.3 Limitation

Although our DeTformer deraining network has achieved superior performance over SOTA networks, it has certain limitations. During the testing stage, we fed our network with a raindrop image and it showed inconsistent behaviour and was unable to remove rain drops as shown in Fig. 9. This is because we did not train DeTformer network with raindrop images.

5 Conclusion

We present a transformer-based deraining network referred to as DeTformer. To process more complex and realistic rain images and restore fine details, we proposed an efficient DeTformer network and also made comparative analysis with SOTA deraining networks. The superior performance of DeTformer network was achieved by a series of improvements. In this work, transformer structure was adopted in deraining single images. We designed “gated depth-wise convolution feed-forward network” (GDWCFN) and applied depth-wise convolution which can improve the capability of modelling local features and suppresses less informative features. We incorporated multi-resolution features in the transformer, where the proposed network can use patches of random scales, and thus, it enables the edge pixels to utilize local features. Furthermore, we designed “multi-head depth-wise convolution transposed attention” (MDWCTA) module which can effectively integrate the multi-scale extracted features and also perform feature interaction across channel dimensions. Extensive experiments on our network demonstrate that it achieves superior performance on both synthetic paired and real rain datasets.

Data Availability Statement All data generated or analysed during this study are included in this published articles Rain100L [38], Rain100H [38], Rain1800 [38], Rain12 [20], Rain1200 [42], Rain14000 [11] and Rain800 [44].

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. M.W. Ahmed, A.A. Abdulla, Quality improvement for exemplar-based image inpainting using a modified searching mechanism. UHD J Sci Technol **4**(1), 1–8 (2020)
2. J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. S. Cao, L. Liu, L. Zhao, Y. Xu, J. Xu, X. Zhang, Deep feature interactive aggregation network for single image deraining. IEEE Access **10**, 103872–103879 (2022)
4. C. Chen, H. Li, Robust representation learning with feedback for single image deraining, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7742–7751 (2021)
5. H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, et al., Pre-trained image processing transformer, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299–12310 (2021)
6. L. Chen, X. Lu, J. Zhang, X. Chu, C. Chen, Hinet: half instance normalization network for image restoration, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 182–192
7. Y.L. Chen, C.T. Hsu, A generalized low-rank appearance model for spatio-temporally correlated rain streaks, in *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 1968–1975
8. S. Deng, M. Wei, J. Wang, Y. Feng, L. Liang, H. Xie, et al., Detail-recovery image deraining via context aggregation networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 14560–14569
9. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. X. Fu, J. Huang, X. Ding, Y. Liao, J. Paisley, Clearing the skies: a deep network architecture for single-image rain removal. IEEE Trans. Image Process. **26**(6), 2944–2956 (2017)
11. X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, J. Paisley, Removing rain from single images via a deep detail network, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 3855–3863
12. X. Fu, B. Liang, Y. Huang, X. Ding, J. Paisley, Lightweight pyramid networks for image deraining. IEEE Trans Neural Netw Learn Syst **31**(6), 1794–1807 (2019)
13. X. Fu, Q. Qi, Z.J. Zha, Y. Zhu, X. Ding, Rain streak removal via dual graph convolutional network, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2 (2021), pp. 1352–1360
14. D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
15. K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, et al., Multi-scale progressive fusion network for single image deraining, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 8346–8355
16. H. Ji, X. Feng, W. Pei, J. Li, G. Lu, U2-former: a nested u-shaped transformer for image restoration. arXiv preprint arXiv:2112.02279 (2021)
17. L.W. Kang, C.W. Lin, Y.H. Fu, Automatic single-image-based rain streaks removal via image decomposition. IEEE Trans. Image Process. **21**(4), 1742–1755 (2011)
18. X. Li, J. Wu, Z. Lin, H. Liu, H. Zha, Recurrent squeeze-and-excitation context aggregation net for single image deraining, in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 254–269
19. Y. Li, Y. Monno, M. Okutomi, Single image deraining network with rain embedding consistency and layered LSTM, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), pp. 4060–4069
20. Y. Li, R.T. Tan, X. Guo, J. Lu, M.S. Brown, Rain streak removal using layer priors, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2736–2744
21. Y. Li, K. Zhang, J. Cao, R. Timofte, L. Van Gool, Localvit: bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021)
22. X. Liu, M. Suganuma, Z. Sun, T. Okatani, Dual residual networks leveraging the potential of paired operations for image restoration, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 7007–7016
23. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, et al., Swin transformer: hierarchical vision transformer using shifted windows, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10012–10022

24. Y. Luo, Y. Xu, H. Ji, Removing rain from a single image via discriminative sparse coding, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 3397–3405
25. C.B. Murthy, M.F. Hashmi, A.G. Keskar, EfficientLiteDet: a real-time pedestrian and vehicle detection algorithm. Mach. Vis. Appl. **33**(3), 47 (2022)
26. T. Ragini, K. Prakash, Progressive multi-scale deraining network, in *2022 IEEE International Symposium on Smart Electronic Systems (iSES)* (IEEE, 2022), pp. 231–235
27. D. Ren, W. Zuo, Q. Hu, P. Zhu, D. Meng, Progressive image deraining networks: a better and simpler baseline, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 3937–3946
28. W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, et al., Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp. 1874–1883
29. F. Tan, Y. Kong, Y. Fan, F. Liu, D. Zhou, L. Chen, et al., SDNet: mutil-branch for single image deraining using swin. arXiv preprint arXiv:2105.15077 (2021)
30. J.M.J. Valanarasu, R. Yasarla, V.M. Patel, Transweather: transformer-based restoration of images degraded by adverse weather conditions, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 2353–2363
31. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., Attention is all you need. in *Advances in Neural Information Processing Systems (NIPS)* (2017), pp. 5998–6008
32. H. Wang, Q. Xie, Q. Zhao, D. Meng, A model-driven deep neural network for single image rain removal, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 3103–3112
33. T. Wang, L. Zhao, P. Huang, X. Zhang, J. Xu, Haze concentration adaptive network for image dehazing. Neurocomputing **439**, 75–85 (2021)
34. Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, H. Li, Uformer: a general u-shaped transformer for image restoration, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 17683–17693
35. W. Wei, D. Meng, Q. Zhao, Z. Xu, Y. Wu, Semi-supervised transfer learning for image rain removal, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 3877–3886
36. H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: introducing convolutions to vision transformers, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 22–31
37. H. Yang, D. Zhou, M. Li, Q. Zhao, A two-stage network with wavelet transformation for single-image deraining. Vis. Comput. **39**, 3887–3903 (2023)
38. W. Yang, R.T. Tan, J. Feng, J. Liu, Z. Guo, S. Yan, Deep joint rain detection and removal from a single image, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1357–1366
39. W. Yang, R.T. Tan, S. Wang, Y. Fang, J. Liu, Single image deraining: From model-based to data-driven and beyond. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 4059–4077 (2020)
40. R. Yasarla, V.M. Patel, 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* (2019), pp. 8405–8414
41. R. Yasarla, J.M.J. Valanarasu, V.M. Patel, Exploring overcomplete representations for single image deraining using cnns. IEEE J Sel Top Signal Process **15**(2), 229–239 (2020)
42. H. Zhang, V.M. Patel, Density-aware single image de-raining using a multi-stream dense network, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 695–704
43. H. Zhang, V.M. Patel, Convolutional sparse and low-rank coding-based rain streak removal, in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2017), pp. 1259–1267
44. H. Zhang, V. Sindagi, V.M. Patel, Image de-raining using a conditional generative adversarial network. IEEE Trans. Circuits Syst. Video Technol. **30**(11), 3943–3956 (2019)
45. S. Zheng, C. Lu, Y. Wu, G. Gupta, SAPNet: segmentation-aware progressive network for perceptual contrastive deraining, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), pp.52–62
46. X. Zhou, Q. Qiu, X. Huang, Deep joint rain detection and removal from a single image. IEEE Trans. Image Process. **28**(2), 865–878 (2019)

47. Y. Zhu, Y. Su, P. Tan, A fast single image haze removal algorithm using color attenuation prior. IEEE Trans. Image Process. **24**(11), 3522–3533 (2018)

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

Authors and Affiliations

Thatikonda Ragini[1] · **Kodali Prakash**[1] · **Ramalingaswamy Cheruku**[2]

✉ Kodali Prakash
kprakash@nitw.ac.in

Thatikonda Ragini
tr712105@student.nitw.ac.in

Ramalingaswamy Cheruku
rmlswamy@nitw.ac.in

[1] Department of Electronics and Communication Engineering, National Institute of Technology Warangal, Hanamkonda, Telangana, India

[2] Department of Computer Science and Engineering, National Institute of Technology Warangal, Hanamkonda, Telangana, India