IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2021, accepted August 24, 2021, date of publication August 27, 2021, date of current version September 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3108516

ETDNet: An Efficient Transformer Deraining Model

QIN QIN[1], JINGKE YAN[1], QIN WANG[3], XIN WANG[1,2,4], MINYAO LI[2], AND YUQING WANG[1]

[1]School of Marine Engineering, Guilin University of Electronic Technology, Guangxi, Beihai 536000, China
[2]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guangxi, Guilin 541004, China
[3]Basic Teaching Department, Guilin University of Electronic Technology, Guangxi, Beihai 536000, China
[4]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China

Corresponding authors: Jingke Yan (20172301005@mails.guet.edu.cn), Xin Wang (304379506@qq.com), and Qin Wang (283252764@qq.com)

This work was supported in part by the General Project of Guangxi Natural Science Foundation under Grant 2019GXNSFAA245053, in part by Guangxi Science and Technology Major Project under Grant AA19254016, and in part by Guangxi Graduate Student Innovation Project under Grant YCSW2021174.

ABSTRACT Rainy days usually degrade the visual effect of images and videos. At present, most deraining models for single images adopt gradual optimization or elimination to remove rain streaks, but actually with relatively low efficiency in real tasks. An efficient one-stage deraining model, Efficient Transformer Derain Network (ETDNet), is proposed to remove rain streaks in single images efficiently. A new Transformer architecture is designed to provide rich multiple scales and context information, making the model extract features in a coarse-to-fine way. Multiple expansion filters with different expansion rates are embedded to predict the suitable kernel for each pixel of the rainy image in a multi-scale way. A multi-scale Loss Function is introduced to restore features with high-fidelity and detail textures. Experiments on Rain100L, Rain100H, and SPA datasets show that the proposed ETDNet reaches the highest PSNR and SSIM values at the fastest speed compared with other models.

INDEX TERMS Rain removal, ETDNet, transformer, multi-scale, loss function.

I. INTRODUCTION

Rainy days will decrease visibility, and the dense rainwater will also cause diffuse reflection, making it hard to restore images with details on rainy days. Therefore, it is difficult to detect objects in some fields such as automatic driving and traffic monitoring, which are based on object detection algorithms (object tracking [1]–[3], pedestrian re-recognition [4]–[6], semantic segmentation [7], [8]).

Restoring images blurred by rain has always been a hot topic in computer vision tasks. Removing rain streaks and restoring all images at a fast speed has become the primary concern for researchers. The latest researches in this field focus on exploring the physical properties and physical layers of rain streaks, which use deep learning to remove rain steaks by optimizing and predicting the relationship between rain streaks and their physical layers. Representative methods include RCDNet [9], JORDER [10], SNet [11], MPRNet [12], etc. They first predict the unknown rain layers and background images to be recovered and then use various network structures to normalize and separate them.

However, those methods often involve many iterations and continuous optimization, which cannot cover all rainfall conditions and require substantial computational overhead. The detail texture restored by those methods will be too smooth as the rain streaks and the background are essentially overlapping in feature space, resulting in losing details in the non-rainfall area of the restored background image. Therefore, those methods are not applicable for removing rain steaks from images with different rainfalls.

To address the above problems, a novel deraining model with small computational overhead to remove rain streaks from single images with different rainfall is suggested, which is featured with the following aspects:

1) A new efficient Transformer module (Efficient Transformer) is designed. Firstly, the self-attention module is introduced to dynamically calculate the weight based on the similarity and affinity between each pair of tokens and construct the multi-scale eigenvector model. Then, Relative Position Encoding and Global Position

The associate editor coordinating the review of this manuscript and approving it for publication was Genoveffa Tortora.

Encoding are designed to make the encoding position more flexible. This module enables the model to extract features in a coarse-to-fine way.

2) Multiple expansion filters with different expansion rates are embedded to predict the suitable kernel for each pixel of the rainy image in a multi-scale way to remove rain streaks effectively.
3) A multi-scale Loss Function is introduced to encourage networks to solve the geometric, color, and texture loss between the actual rainless image and the restored background image with different resolutions. In addition, the prediction process does not require additional computational overhead.
4) The proposed model is compared with other models on synthetic datasets to prove its high performance in removing rain streaks. As shown in Figure 1, ETDNet achieves a better deraining effect on the SPA dataset than the advanced MPRNet.

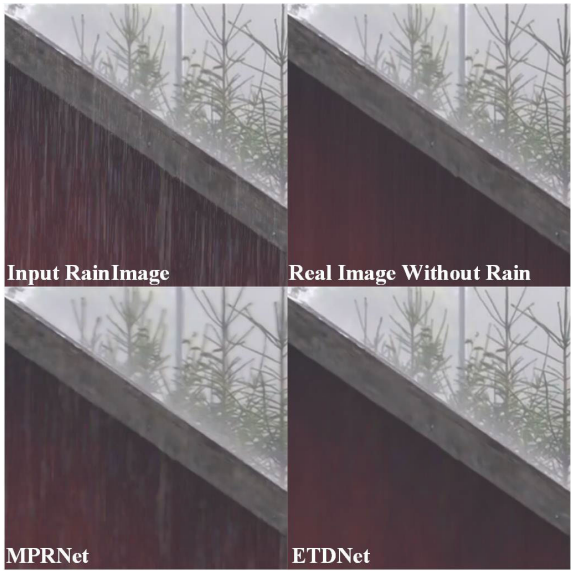


FIGURE 1. The rain-removing effect of the ETDNet and the advanced MPRNet on SPA dataset. The comparisons are conducted on the same PC and both models are retrained for comparison.

II. RELEVANT WORK

This part reviews the latest deraining models, which are roughly divided into video deraining methods and single-image deraining methods based on the difference of the input data.

A. VIDEO DERAINING METHODS

Video deraining methods mainly use the spatial and temporal frequency between adjacent frames of rain streaks in videos to extract their inherent features to restore the background. Garg *et al.* [13] first comprehensively analyzed the impact of rain on the visual effect of the imaging system. They captured the running state of rain based on the spatial and temporal frequency between the rainy image's keyframes and explained the photometry of rain based on Motion Blur Theory. Then they considered the influence of rain streaks and color attributes in videos and extracted background images from the rainy video. Ren *et al.* [14] proposed a model to decompose rain streaks in videos by matrix. They divided the rain streaks into sparse ones and dense ones, extracted the background fluctuation information and optical flow information of the rain streaks in videos, and then marked the moving objects and the sparse rain streaks as multi-label Markov Random Field (MRF), and the dense rain streaks as Gaussian distribution. Finally, they removed the sparse and dense rain streaks by the low-rank matrix of the background. Models based on matrix decomposition also include sparse matrix coding [15], generalized low-rank matrix [16], etc. Liu *et al.* [17] proposed a dynamic routing residual loop network. First, this network extracted the spatial features of the rainy image through the residual network. Then, the context feature information along the time axis of spatial feature is embedded into the network in a ''dynamic routing'' manner. Finally, they used the selection gate of context feature information to select the fusion feature based on the spatial and temporal relationship as the final fusion feature to reconstruct the background image. Liu *et al.* [18] constructed a combinative cyclic deraining network composed of a background reconstruction network based on spatial and temporal correlation, a background reconstruction network based on textures with spatial relationships, and a classification network based on rain streaks. The sub-networks mentioned above can achieve a good effect by removing rain streaks and retaining background details.

Although these models have been well applied to remove rain streaks in videos, they are less efficient in removing rain streaks in single images as there is no time information between two adjacent images. Therefore, removing rain streaks in rainy images becomes a new challenge for researchers.

B. SINGLE-IMAGE DERAINING METHODS

Deep learning has been widely studied in the image denoising field. Most participants in the NTIRE 2020 Competition [19] achieved excellent results by applying deep learning. Tian *et al.* [20] proved that deep learning is helpful for image denoising through a large number of experiments. Ignatov *et al.* [21] developed an image denoising model based on end-to-end deep learning, which can be applied to smartphones to process noisy images quickly and restore background images with high-fidelity. The single image deraining task can be regarded as a sub-task of the image denoising task. Yang *et al.* [22] verified that the data-driven deep learning model is superior to the model-driven sparse coding model [23] and GMM [24] in a single image deraining field. Regarding data-driven deraining models, Wang *et al.* [9] proposed a simple Rain Convolution Dictionary (RCD) model, removing rain streaks iteratively by adopting the near-end gradient technology. The model contained two subnetworks (M-net and B-net) in every iteration. The two subnetworks

used the inherent convolution dictionary to encode the rain shape and update the rainy image, rain kernel convolution, and background layer. Yang *et al.* [10] proposed a model to detect and remove rain streaks in single images through the deep contextual network. They introduced a new rainfall model and a multi-task deep learning network by using prior knowledge. Firstly, the rainfall model simulated the rain in various environments by integrating the binary rainy image and considering the accumulation of rain streaks and the overlapping rain streaks with various shapes and directions. Then a contextual extension network was incorporated into the multi-task deep learning network to learn the binary rain streaks, the rain streak layers, and the clean background. At last, the above steps needed to be repeated to loop the network to remove the rain streaks further. Wang *et al.* [11] proposed to classify rain streaks as the transmission medium, and the rainy image was modeled together with the transmission medium and the fog. SNet (a unit in ShuffleNet [25]) was used to capture the features of the transmission medium of rain streaks. VNet network was used to predict rain fog and rain streaks in a multi-scale way. The encoder of ANet was used to predict atmospheric light and eventually the predicted transmission medium and atmospheric light were used to restore the background. The SNet, VNet, and ANet were trained jointly. He *et al.* [26] considered the density and size of rain streaks and designed a multi-scale network. In their network, a multi-scale rain streaks predictor was used to predict the rain streaks. Then the convolutional sparse coding matrix was used to remove the noise in the rain streaks. Finally, the rain streaks were subtracted from the rainy image to get the rainless background. Jiang *et al.* [27] proposed a multi-scale progressive fusion network, which captured rain streaks by fusing features in a multi-scale collaborative way and circular calculations and used the captured rain streaks to restore the background. Wang *et al.* [28] proposed a deep residual learning model based on FastDerainnet. This model first used a low-pass filter and high-pass filter to remove rain streaks preliminarily, then used the residual structure on the high-frequency component to learn the residual features of rain streaks, and at last used high-frequency components and low-frequency components to calculate rainless background images. Deng *et al.* [29] proposed the DRDNet model, which regarded rain streaks removing and detail recovery as two independent sub-tasks, and the two sub-tasks could remove rain streaks and recover details collaboratively.

In real applications, the above single-image deraining models require complex optimization and gradual improvement, indicating that those models still need further perfection. In this paper, ETDNet, a new single image deraining model, is proposed to efficiently remove rain streaks and restore the rainless background.

III. METHODOLOGY

This section describes ETDNet in great detail. The ETDNet structure is introduced briefly, and its three main components: Efficient Transformer module, expansion filter, and a multi-scale Loss Function are described exhaustively.

A. NETWORK STRUCTURE

The ingle-image deraining models decompose the rain image I^R into the rainless background image I^{NR} and the rain streaks R. Their common formulas ([30]–[32]) are shown in formula (1).

$$I^R = I^{NR} + R, \tag{1}$$

The rainless background image I^{NR} can be obtained by subtracting the predicted rain streaks R from the rainy image I^R. The deraining models based on formula (1) assume the shapes of rain streaks are similar. However, in reality, rain streaks usually have different shapes and irregular distributions due to the swing of the camera and different shooting distances. Those models predicting rain streaks in advance cannot simulate actual complex situations. The network is utilized to extract multi-scale features and assign different weights to rain streaks to address the above problems according to their sizes and transparency. Generally speaking, the deraining process can be considered an image degradation recovery process, which may cause fog, motion blur, occlusion, etc. Therefore, it is reasonable to extract multi-scale information of images for processing, effectively dealing with various degradation problems. To be specific, the rainless image I^{NR} can be obtained by processing the rain image $I^R \in R^{H\times W}$ with multi-scale pixel-level filtering, which can be expressed in formula (2).

$$I^{NR} = K \otimes I^R, \tag{2}$$

where $I^{NR} \in R^{H\times W}$ represents the output background image. $\otimes$ stands for multi-scale pixel filtering operation. Each pixel is processed by the appropriate kernel. $K \in R^{H\times W\times k^2}$ contains the kernel of all pixels. To effectively extract features with multi-scale pixel filtering, the following challenges need to be considered: estimating the spatial transformation and scale transformation of rain image effectively and estimating the detailed transformation of semantic perception. In this paper, the generation function is defined as G, through which θ_G is parameterized and transmitted into the feedforward network for training. Loss Function $\mathcal{L}_{\text{Multi}}$ is optimized to obtain the weight and deviation of the deep network. The calculation method of θ_G is shown in Equation (3).

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\text{Multi}} \left(G_{\theta_G} \left(I_n^R \right), I_n^{NR} \right), \tag{3}$$

In ETDNet, the Shuffle Attention [33] module and Efficient Transformer (III.B) are used to assign different weights to rain streaks with different sizes according to the interdependence of each convolution layer. The SPP [34] structure and expansion filter (III.C) are used to obtain multiple receptive fields of different sizes and capture more context information. By defining the multi-scale Loss Function (III.D), the Loss Function of the first several layers can guide the

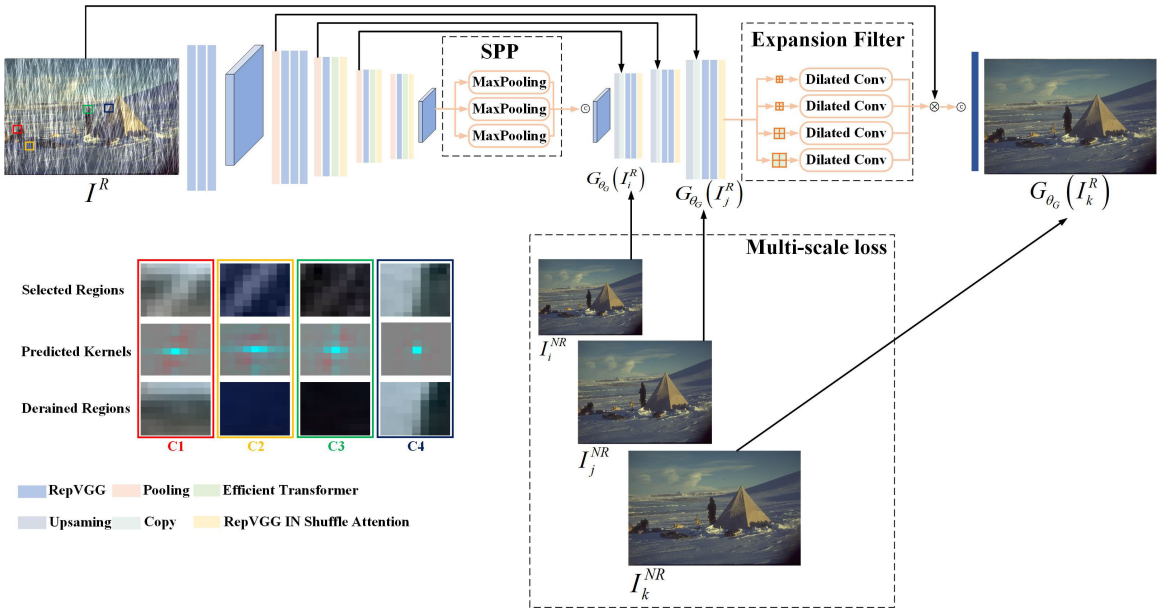


FIGURE 2. The network structure of ETDNet. MaxPooling represents the Global Average Pooling operation. The cores of 3 Global Average Pooling are 5, 9, and 13, respectively, with a step size of 1. Four dilated Convs refer to self-adaptive expansion filtering. They can adapt to different pixels, and their expansion factors are 1, 2, 3, 4, respectively. ˙ represents the fusion of feature maps, ⊗ represents matrix multiplication.

Loss Function of the later layers to optimize the weight and deviation of the network and reduce the loss of detail texture features.

Four representative regions are visualized to illustrate the effectiveness of ETDNet. From C1 to C4 in Figure 2, it can be observed that ETDNet is applicable to remove rain streaks of different shapes and directions. From C1 to C3, it can be noted that this network allocates higher weights to feature maps with fewer rain streaks and lower weights to feature maps with more rain streaks. C4 indicates that ETDNet will not lose the details of the original image and can recover the details better.

Algorithm 1 Learning ETDNet

```
Input: Rainy images I^R
for i = 1 to epoch do
  for j = 1 to batchnum do
    Generate rain map via R = RainMix(R)
    Sample an image pair via (I^R, I^NR) ~ (T^R, T^NR);
    Sample X ~ (I^R, I^NR) and Perform I^R = X + R;
    Derain via Eq. 5 and Eq. 11;
    Calculate Eq. 15 and do back-propagation;
    Update parameters of Conv(·)
  end for
end for
```

B. EFFICIENT TRANSFORMER

The Transformer [35] has attracted researchers working on machine vision for its outstanding performance in natural language tasks. It encodes the dependent items in the input field by the self-attention mechanism to make the extracted features highly expressive. The calculation method of the self-attention mechanism of the early Transformer is as follows:

$$Att(X) = softmax\left(\frac{QK^T}{\sqrt{C}}\right)V, \tag{4}$$

where X, QK^T, V and C represents the input feature map, the key encoding, the value matrix, and the number of channels, respectively. An Efficient Transformer module is proposed in this paper, which can remove rain streaks in a coarse-to-fine way from rainy images with only a small amount of computational overhead. It is divided into Factorized Attention Mechanism, Convolutional Relative Position Encoding, and Convolutional Position Encoding, as shown in Figure 3.

As can be seen from Equation (4), the spatial complexity and time complexity of Factorized Attention Mechanism, Softmax and Attention Map of the early Transformer are $O\left(N^2\right)$ and $O\left(N^2C\right)$, respectively. The early Transformer used $\frac{1}{\sqrt{C}}$ to normalize the output values. Although its normalization is proved to be effective in some tasks (target tracking [1]–[3], pedestrian re-recognition [4]–[6], semantic segmentation [7], [8]), $\frac{1}{\sqrt{C}}$ significantly increases the time complexity. Besides, the application of $\frac{1}{\sqrt{C}}$ will decrease the deraining efficiency as the network of the deraining model is relatively shallow. Therefore, deleting the scaling factor $\frac{1}{\sqrt{C}}$ not only makes the time complexity become $O\left(N^2\right)$, but also brings better deraining performance in experiments. The mathematical expression of Factorized

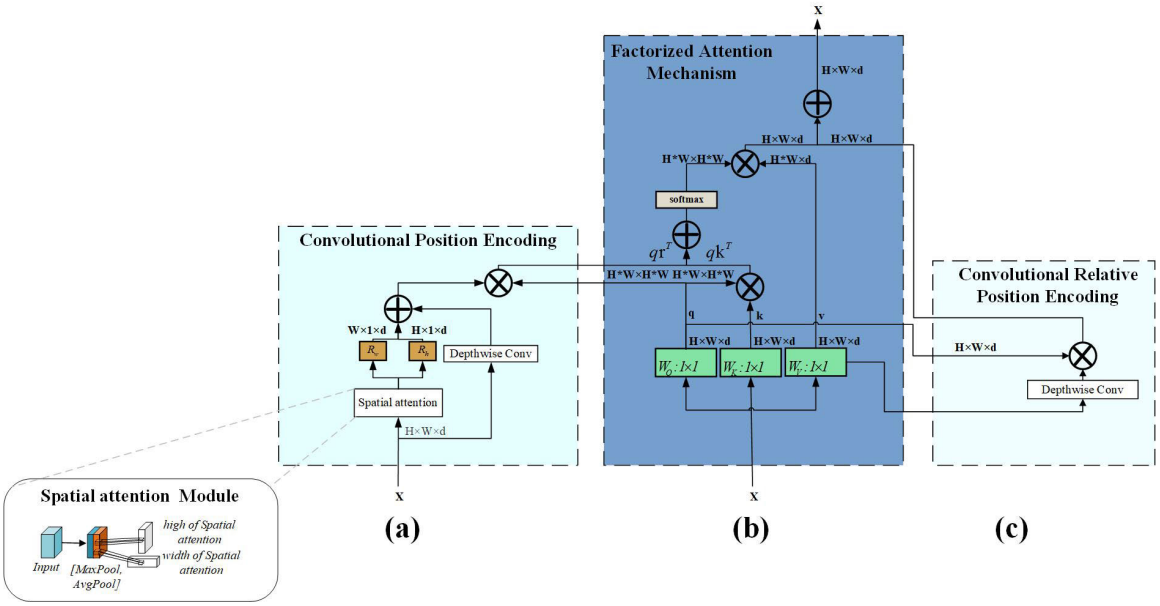


FIGURE 3. Multi-head Attention mechanism layer in Efficient Transformer, 1 × 1 means point-by-point convolution, ⊕ means matrix summation, ⊗ means matrix multiplication.

Attention Mechanism is shown in Formula (5).

$$FactorAtt(X) = softmax\left(QK^T\right)V, \tag{5}$$

Without Position Encoding, the Transformer can only be composed of linear layers and self-attention modules in the deraining algorithms. The Transformer does not know the difference between similar features (for example, big rain streaks can be removed while small rain cannot be removed).

In Convolutional Relative Position Encoding, ViT [36] and Deit [37] insert the absolute position into the input features, limiting the relative position communication between local features. Coat [38] explores the relative position encoding in work. The relative position encoding $P = \left\{\mathbf{p}_i, i = -\frac{M-1}{2}, \ldots, \frac{M-1}{2}\right\}$ in this study can be obtained through the sliding window M in Transformer, and the corresponding feature map $EV \in R^{N\times C}$ can be obtained through the relative position encoding P.

$$\text{RelFactorAtt(X)} = softmax\left(QK^T\right)V + \widehat{EV}, \tag{6}$$

The mathematical expression of encoding matrices $E \in R^{N\times C}$ and EV are as follows:

$$E_{ij} = \mathbb{1}(i,j)\mathbf{q}_i \cdot \mathbf{p}_{j-i}, \quad 1 \le i,j \le N, \tag{7}$$

$$\widehat{EV}_i = \sum_j E_{ij}v_j, \tag{8}$$

where $1(i,j) = 1_{\{|j-i|\le(M-1)/2\}}(i,j)$ represents indicator function. E_{ij} denotes the query of relationship between vector $\mathbf{q}_i$ and value vector $\mathrm{v_j}$ in window M. $(EV)_i$ represents the aggregation of all the value vectors $\mathrm{v_j}$ associated with $\mathbf{q}_i$. $\widehat{EV}$ can be calculated by the two-dimensional Depthwise Convolution. Its mathematical expression is as follows:

$$\widehat{EV} = Q^{\text{img}} \circ \text{DepthwiseConv2D}\left(P^{img}, V^{\text{img}}\right), \tag{9}$$

where $\circ$ represents the matrix product operation. In Convolutional Position Encoding, as the transformer cannot capture long-distance dependencies of features and adapt to the changeable features that are input dynamically, the Relative Position Encoding and the Global Position Encoding are combined to focus on the relative distance between features and the global feature information. This method effectively associates the object information and position information and dynamically adjusts weight to adapt to input feature maps, eventually making the encoding position more meaningful. The mathematical expression of Convolutional Position Encoding is as follows:

$$\hat{\mathrm{x}} = \mathrm{GL(x)} + \text{DepthwiseConv2D (x)}, \tag{10}$$

where GL(x) represents the Global Position Encoding of each pixel's linear operation of the input feature. Inspired by Spatial Attention, a simple but effective Global Position Encoding of each pixel's linear operation is presented. To be specific, this algorithm divides the Global Position Encoding into two modules: one module performs Global Position Encoding on the height axis of the feature map, and the other module performs on the width axis. The Global Attention performing on the width axis and the height axis simulates the Global Position Encoding effectively, which adds the object feature information to the position information, making the dynamic input feature more sensitive to the position information and able to encode the spatial structure of images.

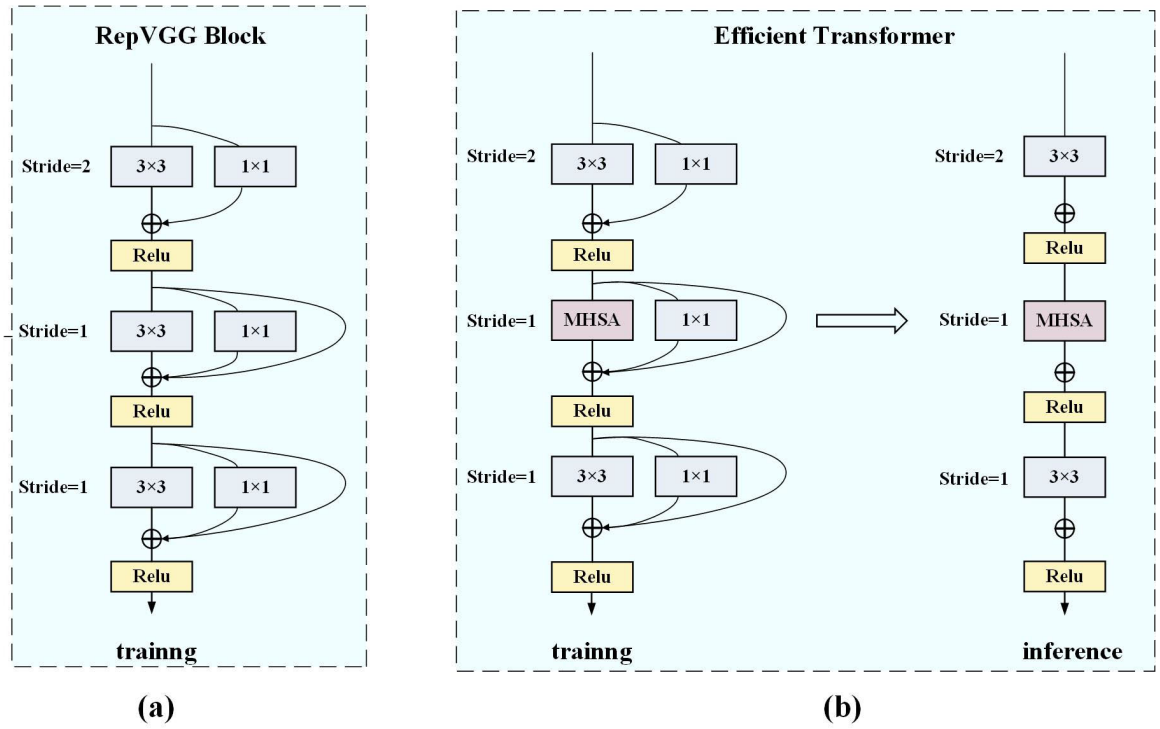


FIGURE 4. (a) is the RepVGG Block, (b) is the Efficient Transformer. The Efficient Transformer can interpret the existing system in Transformer as the bottleneck structure with Multi-head Self-attention (MHSA), but with different concept from the bottleneck structure. It is different from the RepVGG Block in that it uses MHSA instead of 3 × 3 convolution. ⊕ represents the summation of matrixes.

The mathematical expression of GL(x) is as follow:

$$\begin{aligned} &\mathrm{GL(x)} \\ &= \mathrm{x}_h * \sigma(\mathrm{DW}\mathit{Conv}([\mathrm{AvgPool}(\mathrm{x}_h); \mathrm{MaxPool}(\mathrm{x}_h)]) \\ &\quad + \mathrm{x}_w * \sigma(\mathrm{DW}\mathit{Conv}([\mathrm{AvgPool}(\mathrm{x}_w); \mathrm{MaxPool}(\mathrm{x}_w)])), \end{aligned} \tag{11}$$

where $\sigma(\cdot)$ represents the sigmoid operation. In addition, the latest research shows that the RepVGG Block structure [39] is featured with high parallelism and large receptive field of the multi-path architecture during training and the fast speed and memory saving of the single-path architecture during prediction. In ETDNet, the proposed deraining model, the RepVGG Block is extensively used, and an Efficient Transformer module with a bottleneck structure is defined, as shown in Figure 4. The Efficient Transformer adds a parallel 1 × 1 convolution branch and an identity mapping branch to each MHSA. When ETDNet is trained, its Efficient Transformer uses a multi-branch structure to expand the receptive field to speed up the model fitting speed and establishes residual mapping to alleviate the gradient disappearance. When ETDNet conducts prediction, the Efficient Transformer can integrate a 1 × 1 convolution layer and identity mapping into the MHSA layer to reduce the storage units for parameters and speed up the prediction.

C. EXPANSION FILTER

The JORDER model proposed by Yang *et al.* [10] has proved that aggregating context information in a multi-scale way can expand the convolutional receiver field in deraining models, which is very effective for models to learn the features of rain streaks. Therefore, multiple expansion filters are embedded into ETDNet, as shown in Fig. 5. The expansion filters with multiple pixel-wise dilation rates can adapt to rain streaks of different intensities in a multi-scale way. The expansion filters can perceive the position of rain streaks according to the intensity of rain streaks through different expansion factors in the input feature map and the original rain map without loss of resolution. The expansion filters can perceive the position of rain streaks according to the intensity of rainfalls. They allocate relatively low weight to the position with dense rain streaks and relatively high weight to the position with sparse rain streaks and then extract features in a multi-scale way. To not affect the efficiency of the ETDNet model, four filters with expansion filters as 1, 2, 3, 4 respectively are assigned to the network to process rain streaks. Through the expansion filters, four processed feature maps can be obtained. The four feature maps are confused into one feature map, on which a 3 × 3 convolution operation are implemented to remove rain streaks. $\mathbf{f}_{\mathrm{in}}$ is defined as the input feature map and $\mathbf{f}_{\mathrm{out}}$ as the output feature map to provide a detailed description:

$$\mathbf{f}_{\mathrm{out}} = \max\left(0, \sum_{t_p=1}^{4}\left(\mathbf{W}_{\mathrm{in},t_p} * \mathbf{f}_{\mathrm{in},t_p} + \mathbf{b}_{\mathrm{in},t_p}\right).\mathbf{f}_{\mathrm{in}}\right), \tag{12}$$

where * represents convolution operation, variable t_p represents different expansion filters. Variable W_{in} and $\mathbf{b}_{\mathrm{in}}$

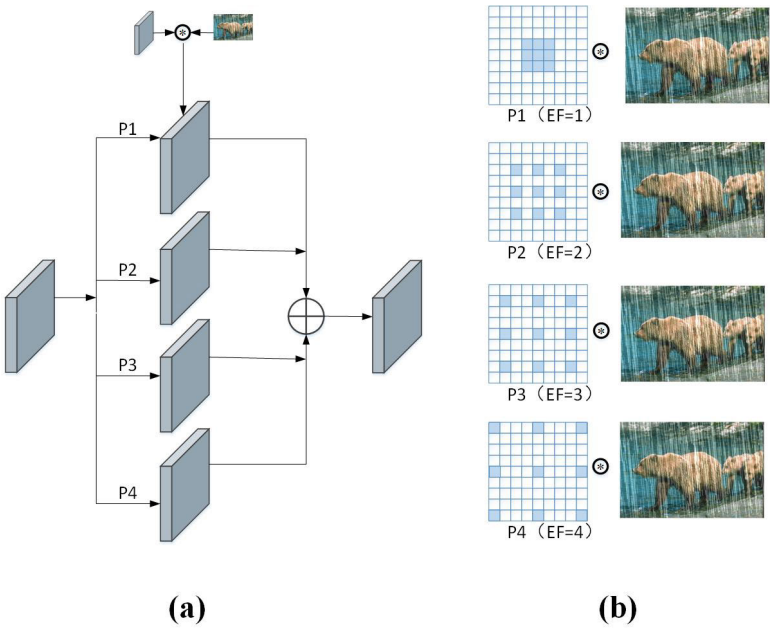


FIGURE 5. (a) is the expansion filter. Multiple expansion filters predict each pixel; (b) denotes the feature map predicted by the expansion filter multiplying the rainy image, making the network suitable for rain streaks with different sizes.

represent the parameters of the expansion filter and the basic parameters of the convolution layer, respectively.

D. MULTI-SCALE LOSS FUNCTION

The study on deraining explores how to remove more rains and restore high-quality images to the largest degree. In ETDNet, the shallow network layer can learn the edge, color, brightness, and other underlying features, the middle network layer can learn the texture features, and the deep network can learn distinctive, identifiable, and relatively complete features. MSE is often used as the Loss Function in deraining algorithms, making the output images too smooth (losing many details or high-frequency parts). Therefore, feature maps of specific layers selected appropriately are input into the Loss Function to recover more details. Multi-scale Feature Loss Function is involved in training ETDNet, which makes the network modulate the rain streak features in the feature space from multiple scales in a coarse-to-fine way to ensure the model restores features with high fidelity and details.

The feature maps with low-resolution in the shallow layer usually contain rich features with high-fidelity, such as edges, colors, and brightness, but it is challenging to acquire their accurate geometric and texture features. MSE Loss Function and the TV Loss Function are used to define the Loss Function of the shallow layers. They can roughly transfer features with high-fidelity to the feedforward network, which leaves the features with high-fidelity to the subsequent network layer to carry out the calculation. The Loss Function of the shallow layer is defined as follows.

$$\mathcal{L}_{\text{Shallow}} = \lambda_{mse} \parallel \hat{y} - y^2 \parallel + \lambda_{tv} \hat{y}^2, \tag{13}$$

λ_{mse} and λ_{tv} represent the weights of MSE Loss Function and TV Loss Function, respectively. The middle network layer further enhances the texture features with perceptual meaning by using the previous information and the pre-trained VGG Loss Function [40], MS-SSIM Loss Function [41], and L_1 Loss Function, which reduces the detail loss and increases the fidelity of the restored background. The Loss Function of the middle layer is defined as follows.

$$\mathcal{L}_{Middle} = \lambda_{per} \parallel \phi(\hat{y}) - \phi(y)^2 \parallel + \lambda_1 \parallel \hat{y} - y \parallel_1 + \lambda_{mssim} \prod_{i}^{M} \parallel f_2(\hat{y}) - f_2(y) \parallel_1, \tag{14}$$

where ϕ stands for using feature maps from conv1. . . , Conv5 in VGG network [42], and M signifies the scaling factor of the feature map, $M = 0.5, 1.0, 2.0, 4.0, 8.0$. λ_{per}, λ_1, and λ_{mssim} represent the weights of VGG Loss Function, L_1 Loss Function, and MS-SSIM Loss Function, respectively. In the deep network layer, L_1 Loss Function and SSIM Loss Function are utilized to reuse features and explore key features to maintain the minimum distance between the recovered results and the deep feature space of the real background, further enhancing the expression ability of details. The Loss Function of the deep layer is defined as follows:

$$\mathcal{L}_{\text{Deep}} = \lambda_1 \parallel \hat{y} - y \parallel_1 + \lambda_{ssim} \parallel f_2(\hat{y}) - f_2(y) \parallel_1, \tag{15}$$

λ_1 and λ_{ssim} represent the weights of the L_1 Loss Function and the SSIM Loss Function, respectively. The overall object Loss Function can be written as:

$$\mathcal{L}_{NR} = \mathcal{L}_{\text{Shallow}} + \mathcal{L}_{Middle} + \mathcal{L}_{\text{Deep}}, \tag{16}$$

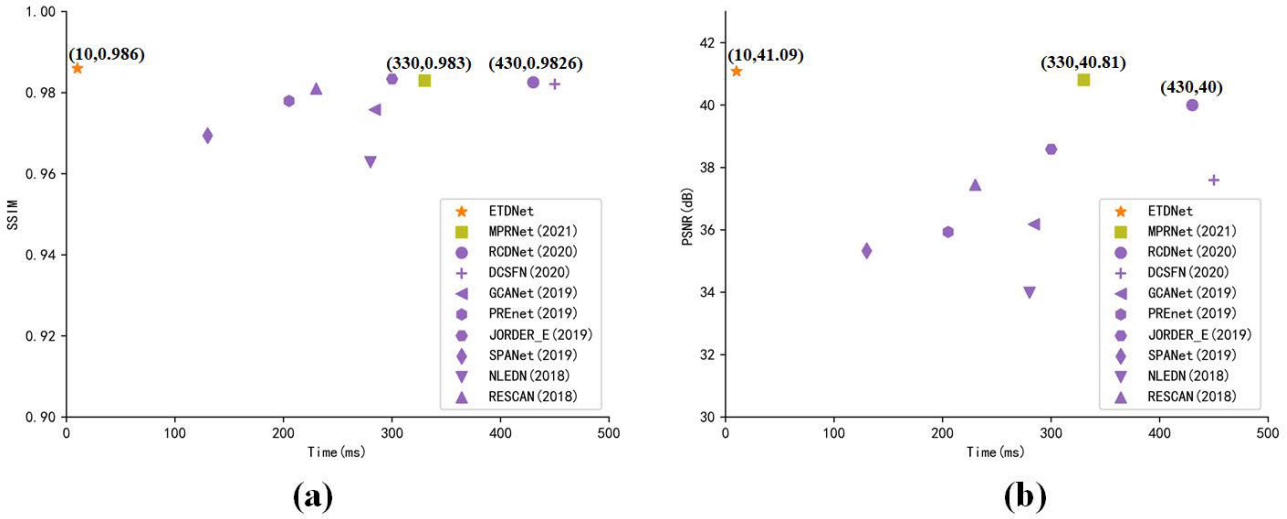


FIGURE 6. Comparison of PSNR, SSIM and speed on Rain100L.

IV. EXPERIMENT AND RESULT ANALYSIS

In this section, the experimental environment is introduced first, and then the ETDNet, the proposed model, is verified on three different datasets to prove its high performance.

A. EXPERIMENT ENVIRONMENT

Datasets: comparative experiments of different techniques are carried out on three popular datasets: Rain100L (synthetic datasets), Rain100H [10], [43] (synthetic datasets), and the SPA [44] (synthetic datasets similar to the actual rainfalls) to explore their influences on the performance of deraining models.

Evaluation indexes: peak signal-to-noise ratio (PSNR) and structural similarity Index Measure (SSIM) are used as evaluation indexes to evaluate the performance of deraining models. Generally, the larger their values are, the better the deraining effect is.

Benchmark models: different deraining models are experimented on Rain100L and Rain100H to verify their performances, including NLEDN (Li *et al.* [45], 2018), RESCAN (Li *et al.* [46], 2018), JORDER (Yang *et al.*, 2019), SPANet (Wang *et al.*, 2019), GCANet (Chen *et al.* [47], 2019), PRENET (Ren *et al.* [48], 2019), RCDNet (Wang *et al.* [9], 2020), DCSFN (Wang [49], 2020), MPRNet (Zamir *et al.* [12], 2021). Then the proposed ETDNet is compared with 6 benchmark models on SPA datasets to verify its efficiency, including RESCAN (Li *et al.*, 2018), SIRR (Wei *et al.* [50], 2019), PRENet (Ren *et al.*, 2019), SPANet (Wang *et al.*, 2019), RCDNet (Wang *et al.*, 2020), MPRNet (Zamir *et al.*, 2021).

All models are trained on Ubuntu 18.04, GPU1060, and Pytorch1.6. Adam is used as the optimizer. The initial value of the learning rate is 0.002.

TABLE 1. Comparison of PSNR and SSIM on Rain100L and Rain100H.

Datasets	Rain100L		Rain100H	
Metrics	PSNR(dB)	SSIM	PSNR(dB)	SSIM
NLEDN(2018)	33.99	0.9629	27.21	0.8650
RESCAN(2018)	37.45	0.981	23.98	0.7612
GCANet(2019)	36.18	0.9758	25.99	0.8200
PRENet(2019)	35.94	0.978	26.10	0.8661
JORDER E(2019)	38.59	0.9834	30.50	0.8967
SPANet(2019)	35.33	0.9694	25.11	0.8332
DCSFN(2020)	37.60	0.9821	27.76	0.8860
RCDNet(2020)	40.00	0.9826	31.28	0.9093
MPRNet(2021)	40.81	0.983	31.31	0.9160
ETDNet	**41.09**	**0.986**	**32.35**	**0.9299**

B. COMPARATIVE EXPERIMENTS ON RAIN100L AND RAIN100H

In Table 1, ETDNet is compared with the other nine benchmark models on Rain100L and Rain100H. The experimental results indicate that on Rain100H, compared with the NLEDN (2018), the PSNR and SSIM of the proposed ETDNet increase by 4.04 DB and 0.0649, respectively. Compared with the latest MPRNet, they increase by 1.04 DB and 0.0139, respectively. Compared with the RCDNet (2020), they increase by 1.07 DB and 0.0206, respectively.

Figure 6 shows the comparison of ETDNet with the other nine benchmark models on Rain100L. Figure 6 illustrates that the deraining speed of ETDNet on Rain100H is 33 times faster than that of MPRNet and 43 times faster than that of the RCDNet. Figure 6(a) and Figure 6(b) indicate that the ETDNet can achieve the highest PSNR and SSIM at the fastest speed, proving its excellent performance.

In Figure 7, the partial predicting results of ETDNet, RCDNet, and MPRNet on Rain100L and Rain100H are visualized. In Case 1 and Case 2, it is obvious that RCDNet and MPRNet destroy the details of the picture when removing rain

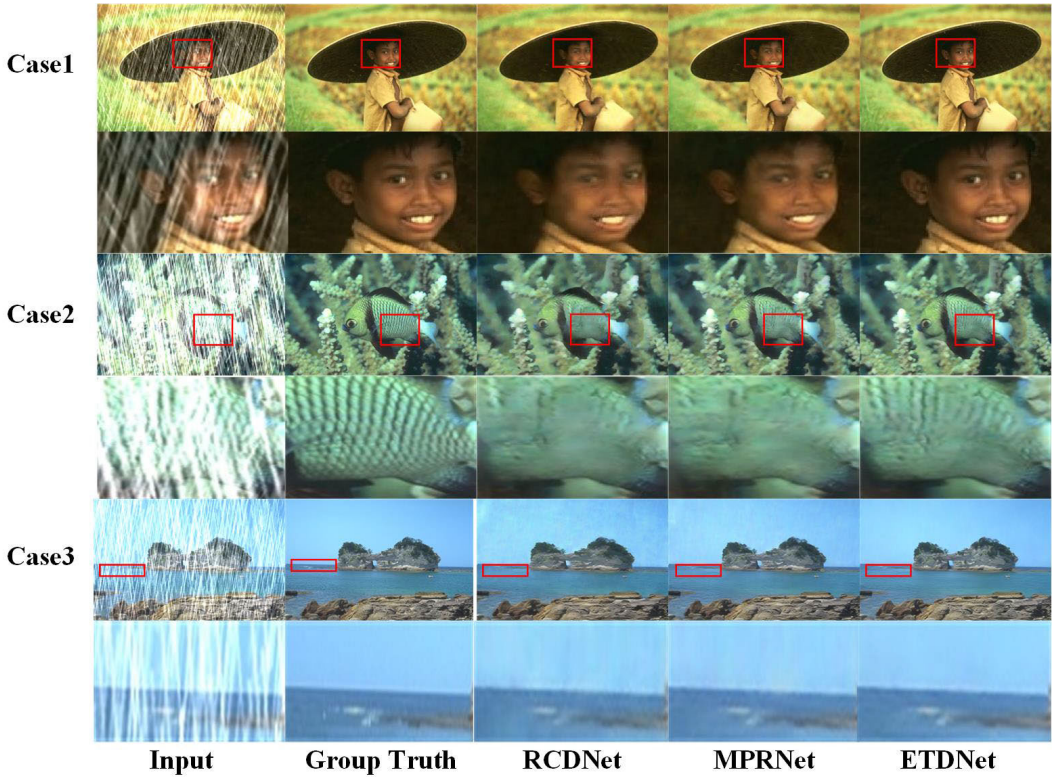


FIGURE 7. Three visualized results of RCDNet, MPRNet, and ETDNet on Rain100L(Case1) and Rain100H (Case 2 and Case 3). The main differences are magnified.

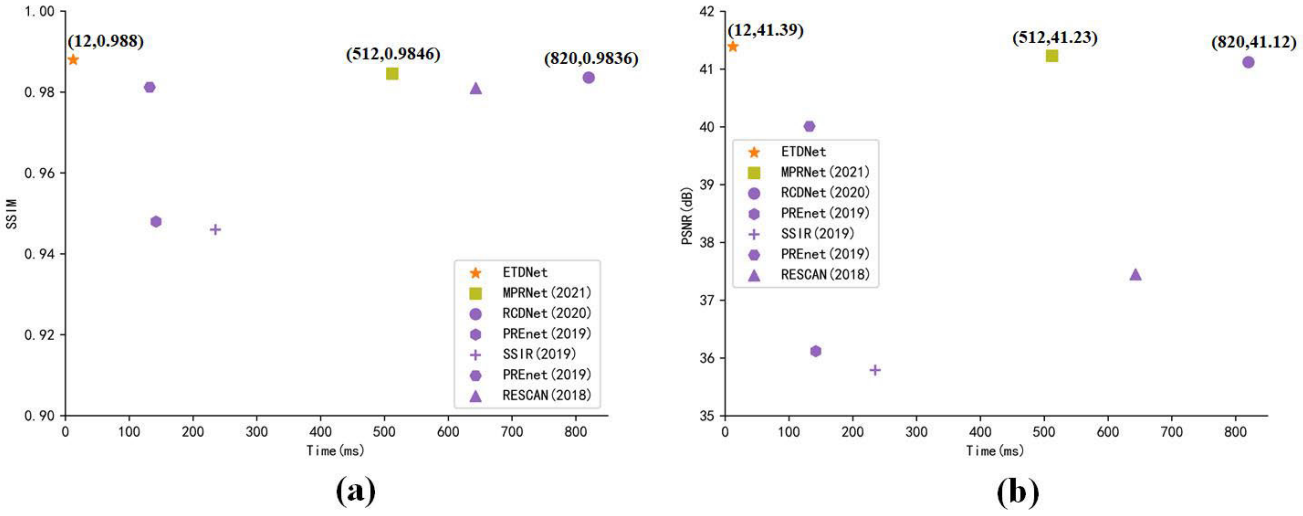


FIGURE 8. Comparison of PSNR, SSIM and speed on SPA.

streaks, such as the face of a boy in Case 1 and the fine lines of fish in Case 2. However, ETDNet removes more rain streaks but retains more detailed features. In Case 3, RCDNet and MPRNet do not remove small rain streaks, while ETDNet removes all rain streaks, indicating that ETDNet is more efficient in removing rain steaks of different sizes.

C. COMPARATIVE EXPERIMENTS ON SPA DATASETS

The synthetic rain images in Rain100L and Rain100H datasets are different from the real rain images, but the synthetic rain images in SPA datasets [44] are much more similar to the ones taken in real life. ETDNet is compared with the other six benchmark models on SPA datasets.

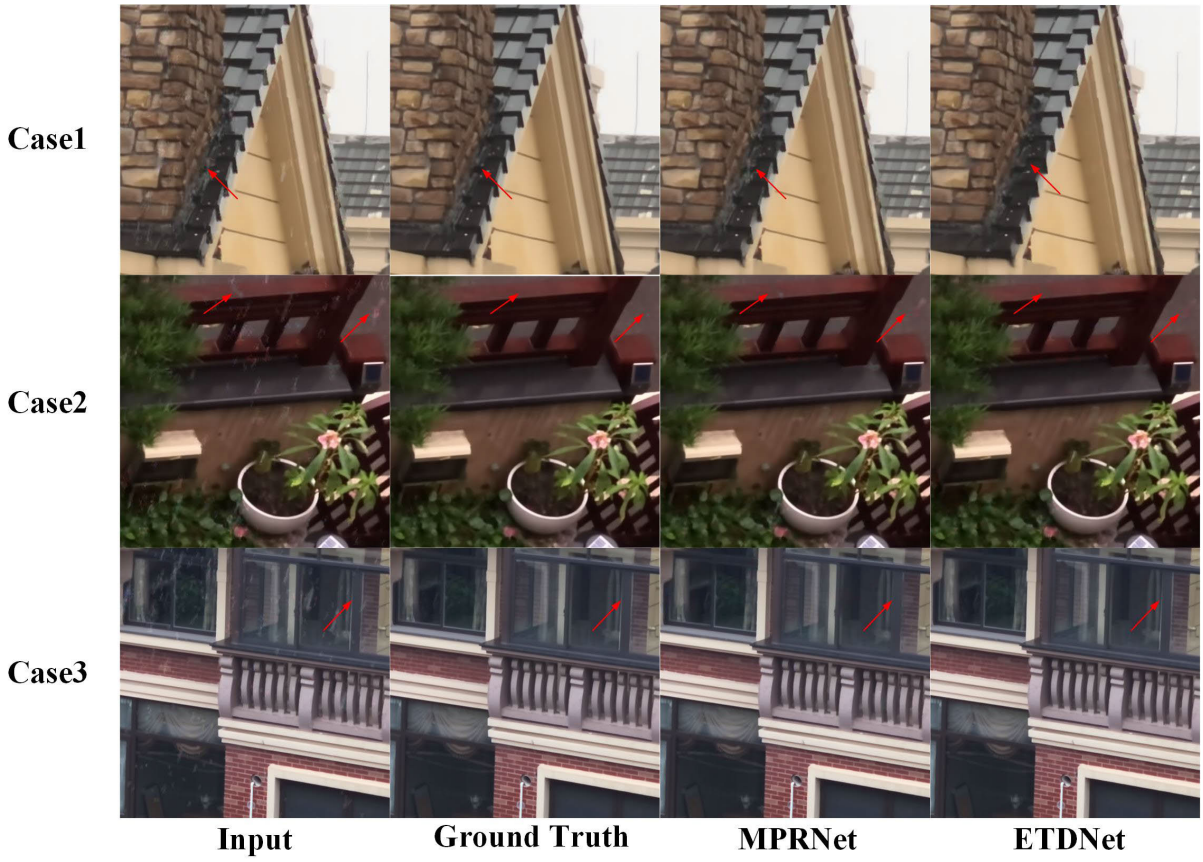


FIGURE 9. The visualized results of MPRNet and ETDNet on SPA. The red arrow shows the main difference between ETDNet and MPRNet.

Figure 8(a) and Figure 8(b) show that the SSIM and PSNR of ETDNet are higher than those of other benchmark models. Besides, the deraining speed of ETDNet is 26 times faster than that of MPRNet on SPA. In Figure9, the deraining effect of MPRNet and the ETDNet on SPA are visualized. In Case 1, MPRNet does not remove small rain streaks, while ETDNet successfully removes small rain streaks. In Case 2, MPRNet does not remove relatively large rain streaks, but ETDNet successfully removes all rain streaks. In Case 3, when removing rain streaks, MPRNet deletes the window frame, but ETDNet successfully retains the details of the window frame. The above all show that the proposed ETDNet is more efficient in removing rain streaks of different shapes and retaining details than other models.

D. ABLATION EXPERIMENT

This section defines four variant models to verify the deraining effects of the expansion filter, the Efficient Transformer, and the multi-scale Loss Function in ETDNet on Rain100H. The first model (ETDNet-v1) consists of RainMix, SPP, Shuffle Attention module, and RepVGG Block; the second model (ETDNet-v2) adds the expansion filter based on ETDNet-v1; the third model (ETDNet-v3) adds the Efficient Transformer module based on ETDNet-v2. The above three models are trained with L1 Loss Function. The fourth method (ETDNet-v4) adds the multi-scale Loss Function based on the ETDNet-v3. As shown in Table 2, the PSNR and SSIM of the four mentioned models improve gradually, indicating that the expansion filter, the Efficient Transformer, and the multi-scale Loss Function can facilitate removing rain streaks. In addition, the parameters of the four models are analyzed. It is observed that ETDNet-v3 with the Efficient Transformer instead of convolution reduces parameters, suggesting the Efficient Transformer proposed in this article is applicable in other real-time model architectures.

TABLE 2. The ablation experiments of four variant models on Rain100H on GPU 1060.

Variant	Parameter(M)	PSNR(dB)	SSIM
ETDNet-v1	32.02	30.22	0.8843
ETDNet-v2	32.03	30.37	0.8982
ETDNet-v3	32.91	31.41	0.9181
ETDNet-v4	**32.97**	**32.35**	**0.9299**

As is shown in Figure 10, the prediction results on Rain100H further prove the advantages of the proposed ETDNet. In Case 2 and Case 4, ETDNet-v2 remove more rain streaks than ETDNet-v1, indicating that the expansion filter helps to remove more rain streaks. In Case 2 and Case 4, ETDNet-v3 removes more small rain streaks than ETDNet-v2, denoting that the Efficient Transformer helps

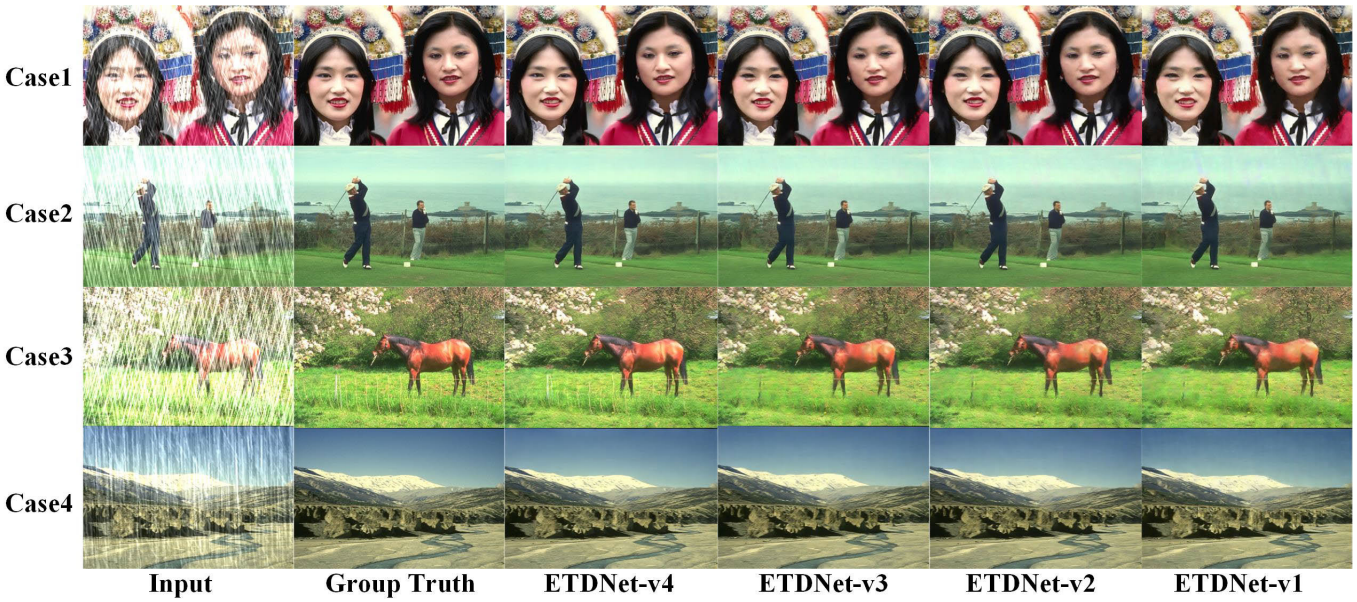


FIGURE 10. The four visualized results of the four variant models of ETDNet on Rain100H.

remove small rain streaks. In Case 1 and Case 3, ETDNet-v4 removes rain streaks and restores more detailed features, such as the face in Case1 and the fence in Case2, signifying that the multi-scale Loss Function helps recover details.

V. CONCLUSION

In this paper, a one-stage deraining model for images based on deep learning is introduced from a new perspective. Compared with other deraining models, the ETDNet model proposed in this paper can remove rain streaks faster and recover more texture details. The high performance of ETDNet is attributed to the following aspects. First, the Efficient Transformer module is introduced, which can adapt to different features dynamically, enabling the model to extract features in a coarse-to-fine way and remove small rain streaks. Then, the expansion filters are incorporated to enlarge the receptive field, finding suitable kernels for different rain streaks to remove rain streaks in a multi-scale way. At last, the multi-scale Loss Function is integrated to solve the loss of precise color with high-frequency and detailed texture in the background. Comparative experiments of the proposed model and other models are conducted on three synthetic datasets to verify the efficiency and high performance of ETDNet in removing rain streaks. Actually, some failure methods have been tried in experiments, such as Frelu, Batch Normalization, and evaluation network for conflict training. However, the PSNR and SSIM of ETDNet have decreased, so it is concluded that these methods may be more suitable for models with deep networks.

This research still needs to be perfected. For example, there is no detailed discussion on whether the ETDNet model is suitable for images with the same parameters to denoise or whether multiple sliding windows of different sizes can further realize the exchange of feature information and improve the performance of the Transformer. In the future, we plan to apply ETDNet into other computer vision tasks, such as image reconstruction, target detection, panoramic segmentation, and use simpler feedforward networks to deal with more complex application scenarios.

ACKNOWLEDGMENT

The authors would like to thank Peng Xie from Southwest Jiaotong University and Jiaqi Xu from the University of Science and Technology of China for their insights and feedback on the first draft of this article.

REFERENCES

[1] M. Danelljan, L. Van Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7183–7192.

[2] P. Dai, R. Weng, W. Choi, C. Zhang, Z. He, and W. Ding, "Learning a proposal classifier for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 2443–2452.

[3] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple object tracking with correlation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 3876–3886.

[4] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 26, 2021, doi: 10.1109/TPAMI.2021.3054775.

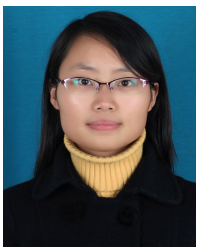
[5] X. Liu, P. Zhang, C. Yu, H. Lu, and X. Yang, "Watching you: Global-guided reciprocal learning for video-based person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13334–13343.

[6] Z. Bai, Z. Wang, J. Wang, D. Hu, and E. Ding, "Unsupervised multi-source domain adaptation for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 12914–12923.

[7] G. Sun, W. Wang, J. Dai, and L. V. Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 347–365.
[8] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 8782–8791.
[9] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3103–3112.
[10] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, Jun. 2020.
[11] Y. Wang, Y. Song, C. Ma, and B. Zeng, "Rethinking image deraining via rain streaks and vapors," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 367–382.
[12] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 14821–14831.
[13] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, pp. 1–15.
[14] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4210–4219.
[15] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1708–1716.
[16] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.
[17] J. Liu, W. Yang, S. Yang, and Z. Guo, "D3R-net: Dynamic routing residue recurrent network for video rain removal," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 699–712, Feb. 2019.
[18] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3233–3242.
[19] A. Abdelhamed, M. Afifi, R. Timofte, and M. S. Brown, "Ntire 2020 challenge on real image denoising: Dataset, methods and results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 496–497.
[20] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Netw.*, vol. 131, pp. 251–275, Nov. 2020.
[21] A. Ignatov, K. Byeoung-Su, R. Timofte, and A. Pouget, "Fast camera image denoising on mobile gpus with deep learning, mobile ai 2021 challenge: Report," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 2515–2524.
[22] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, "Single image deraining: From model-based to data-driven and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 19, 2020, doi: 10.1109/TPAMI.2020.2995190.
[23] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3397–3405.
[24] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2736–2744.
[25] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[26] J. He, L. Yu, G.-S. Xia, and W. Yang, "Single image deraining with continuous rain density estimation," 2020, *arXiv:2006.03190*. [Online]. Available: http://arxiv.org/abs/2006.03190
[27] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8346–8355.
[28] X. Wang, Z. Li, H. Shan, Z. Tian, Y. Ren, and W. Zhou, "Fastderainnet: A deep learning algorithm for single image deraining," *IEEE Access*, vol. 8, pp. 127622–127630, 2020.
[29] S. Deng, M. Wei, J. Wang, Y. Feng, L. Liang, H. Xie, F. L. Wang, and M. Wang, "Detail-recovery image deraining via context aggregation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14560–14569.
[30] Y. Wei, Z. Zhang, Y. Wang, H. Zhang, M. Zhao, M. Xu, and M. Wang, "Semi-deraingan: A new semi-supervised single image deraining," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.
[31] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Semi-supervised image deraining using Gaussian processes," 2020, *arXiv:2009.13075*. [Online]. Available: http://arxiv.org/abs/2009.13075
[32] K. Han and X. Xiang, "Decomposed cyclegan for single image deraining with unpaired data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 1828–1832.
[33] Q.-L. Zhang and Y.-B. Yang, "SA-net: Shuffle attention for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2235–2239.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762
[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*. [Online]. Available: http://arxiv.org/abs/2010.11929
[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2020, *arXiv:2012.12877*. [Online]. Available: http://arxiv.org/abs/2012.12877
[38] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," 2021, *arXiv:2104.06399*. [Online]. Available: http://arxiv.org/abs/2104.06399
[39] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," 2021, *arXiv:2101.03697*. [Online]. Available: http://arxiv.org/abs/2101.03697
[40] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
[41] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," 2015, *arXiv:1511.08861*. [Online]. Available: http://arxiv.org/abs/1511.08861
[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[43] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1357–1366.
[44] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12270–12279.
[45] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, "Non-locally enhanced encoder-decoder network for single image de-raining," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1056–1064.
[46] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 254–269.
[47] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua, "Gated context aggregation network for image dehazing and deraining," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1375–1383.
[48] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3937–3946.
[49] C. Wang, X. Xing, Y. Wu, Z. Su, and J. Chen, "DCSFN: Deep cross-scale fusion network for single image rain removal," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1643–1651.
[50] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3877–3886.

QIN QIN is currently a Senior Engineer, and mainly studies image processing, big data, and wireless sensors.

JINGKE YAN is currently pursuing the Ph.D. degree with the School of Marine Engineering, Guilin University of Electronic Technology. His research interests include image processing, deep learning, knowledge graph, reinforcement learning, and natural language.



QIN WANG is currently a Lecturer with Guilin University of Electronic Technology, focusing on computer science, translation theory and practice, professional literature translation.

XIN WANG is currently an Associate Professor and a Master's Supervisor. He is the main participant of the National Natural Science Foundation of China, and the Principal Investigator of the General Project of Guangxi Natural Science Foundation, Guangxi Science and Technology Major Project, and Guangxi Government Planning Project. He has published more than 40 articles on core journals at home and abroad. He has also applied for more than ten software copyrights. His main research interests include image processing, network information security, the Internet of Things, and data mining.

MINYAO LI is currently pursuing the degree with the School of Computer Science and Information Security, Guilin University of Electronic Technology. His main research interests include machine learning and microcontroller development.

YUQING WANG is currently pursuing the Ph.D. degree with the School of Marine Engineering, Guilin University of Electronic Technology. Her main research interests include machine learning and reinforcement learning.