Image Deblurring Using Fusion Transformer-Based Generative Adversarial Networks

Jionghui Wang[1(✉)], Zhilin Xiong[2], Xueyu Huang[2,3], Haoyu Shi[2], and Jiale Wu[2]

[1] Minmetals Exploration and Development Co. Ltd., Beijing 100010, China
wangjh@minmetals.com
[2] School of Software Engineering, Jiangxi University of Science and Technology, Nanchang 330013, People's Republic of China
[3] Key Laboratory of Virtual Digital Factory and Cultural Communications, Nanchang 330013, People's Republic of China

Abstract. Using the Transformer for motion deblurring enables a broader receptive field, and by stacking multiple Transformer modules, it captures global correlations in features. However, this increases network complexity and poses convergence challenges. To address this, a Generative Adversarial Network called XT-GAN, which combines multiple-scale Transformers, has been proposed.XT-GAN leverages pyramid features from a convolutional network as a lightweight substitute for multi-scale inputs. Within the output pyramid convolutional features, different-scale features are computed in parallel using multi-head self-attention. These features are combined with a proposed feature enhancement module to represent information at different scales. Finally, the network outputs from various modules are concatenated and restored to the original image size.In experiments conducted on the synthetic dataset GoPro, XT-GAN outperformed ordinary networks such as DeblurGAN, DeepDeblur, and SRN. It achieved a reduction in computational complexity of at least 70% while achieving PSNR and SSIM values of 29.13dB and 0.923, respectively. XT-GAN also demonstrated good robustness in the real dataset RealBlur-J, with PSNR and SSIM values of 28.40 and 0.852. It effectively handles motion blur in real-world scenarios, suppresses image artifacts, and restores natural and clear details.

Keywords: image deblurring · Transformer · multi-head attention · GAN · multi-scale fusion

1 Introduction

In the process of image capture, when the photographed object is in relative motion with the image capture device or is affected by factors such as defects in the capture device or the environment, it often leads to motion blur in the captured photos. This

This work is supported by the National Key Research and Development Program of China 2020YFC1909602.

C. Wu et al. (Eds.): MONAMI 2023, LNICST 559, pp. 136–153, 2024.
https://doi.org/10.1007/978-3-031-55471-1_11

blurring obscures the features of the photographed object, affecting identification, and the resulting low-quality images do not meet the requirements of advanced visual tasks such as object recognition and semantic segmentation. Therefore, conducting research on image deblurring is not only of practical significance but also holds important research value.

Traditional image processing techniques address various degrees of blur by estimating the point spread function (PSF). However, motion blur in the real world is more complex, and solving the PSF for each pixel becomes an ill-posed problem [1]. In traditional methods, errors in calculating the blur kernel directly impact the deblurring results, leading to poor restoration effects or even image distortion.

Deep learning methods have shown better adaptability to various blurry scenarios and possess stronger generalization capabilities, enabling them to handle more complex real-world blurs. Sun et al. [2] utilized a convolutional neural network (CNN) combined with a Markov random field to jointly estimate the motion blur kernel and perform restoration for a single image. However, this method still has limitations when dealing with computationally complex spatial blur kernels. To address this issue, Nah et al. [1] proposed the application of multi-scale information to provide a wide receptive field and avoid direct estimation of the blur kernel, achieving better results. Building upon the "coarse-to-fine" structure design of Nah et al. [1], Tao et al. [3] further introduced weight sharing of the feature extraction module across sub-networks at different scales, reducing the feature extraction burden in different parts of the model.

The stacking of multi-scale sub-networks unavoidably introduces greater model complexity and longer computation time. Cho et al. [4] proposed an asymmetric feature fusion network that uses a single encoder for inputting down-sampled or subsampled images and a single decoder for outputting deblurred images. This approach effectively fuses multi-scale features. Kupyn et al. [5] applied Generative Adversarial Networks (GANs) to image deblurring tasks and introduced the Feature Pyramid Network (FPN) into the deblurring task in their DeblurGANv2 [6]. The fusion of multi-scale pyramid features [7] achieved deblurring results similar to those obtained with multi-scale inputs but at a lower computational complexity. Zhang et al. [8] proposed using a GAN to learn blurring and guide another GAN to learn deblurring, aiming to enable the network to better recover images by learning the blurring process of the images.

Handling complex spatial motion blur requires a larger receptive field [3], and multi-scale fusion achieves the goal of information flow and sharing between different scale features. Existing networks [3, 9] increase model size and inference time by stacking multi-scale convolutions, changing the size of convolution kernels, or increasing network depth to expand the receptive field. However, interactions between features at the same scale are still limited by the distance of the convolution kernel. Transformers, with their global interaction and the ability to capture long-range dependencies, have been proposed in literature [10]. A cyclic network structure completely based on self-attention mechanisms has been introduced, utilizing multiple stacked self-attention modules to establish connections between every feature on the feature map, unrestricted by the size of convolution kernels. While Transformers can capture long-range dependencies for individual features, convolutions still possess the advantages of parameter sharing and

translational invariance. Moreover, the computational complexity of pure attention networks grows quadratically with the length of the input sequence, which is not acceptable for computing-intensive tasks with high-resolution images.

Based on the analysis mentioned earlier, we propose a Fusion Transformer-based Generative Adversarial Network for Image Deblur (XT-GAN). The main contributions and work of XT-GAN are as follows:

1. XT-GAN utilizes pyramid features as a lightweight alternative to multi-scale inputs, resulting in lower complexity and faster processing of high-resolution images.
2. For the multi-scale features from the Feature Pyramid Network (FPN) [7], XT-GAN employs a Transformer-based cross-layer parallel computing structure. This approach extracts rich context while considering the locality of convolutions, thereby improving the deblurring performance of the model.
3. XT-GAN introduces Attention Enhancement Modules to enhance the information interaction between the self-transformer (ST) and low-level/high-level features. This special fusion structure enhances the flow of information within the network, improves the utilization of semantic information from different hierarchical features, and further enhances the network performance of XT-GAN.

2 Related Work

The XT-GAN network structure proposed in this paper is illustrated in Fig. 1.

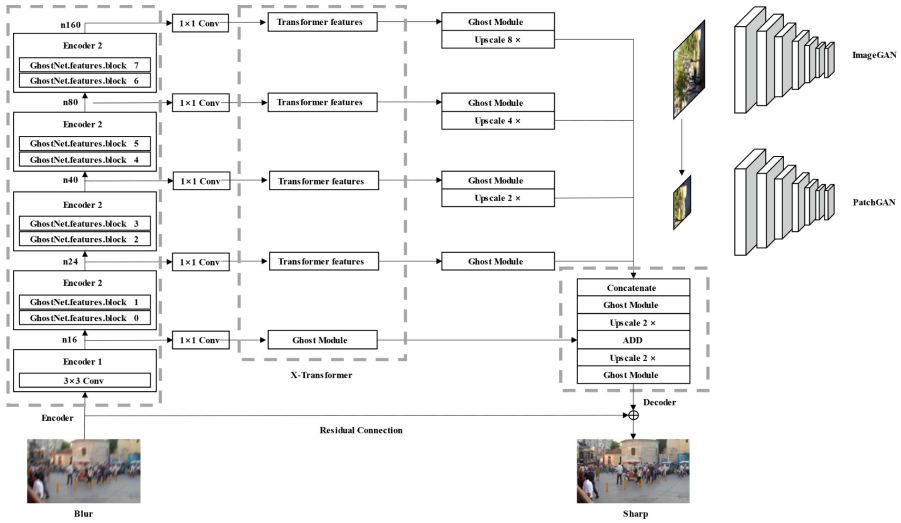

Fig. 1. Depicts the network architecture of XT-GAN.

The generator consists of an encoder feature extraction backbone called the Backbone, a cross-layer feature fusion module called X-Transformer (XT), and a decoder for image scale restoration. The discriminator network consists of two components: ImageGAN and PatchGAN, which are based on VGG19 [11] and a dual-scale discriminator network [12], respectively.

2.1 Design of the Generator

The generator of XT-GAN follows an Encoder-Decoder structure, which is commonly used for image super-resolution restoration and has been proven effective for image deblurring tasks as well [3].

In the Encoder part, the Backbone network performs spatial compression and transformation on the input image. The number of channels and the level of abstraction of the features increase progressively, while the network learns and separates the blurry features of the image. In recent years, various Backbone networks [13] have been proposed, each with different sensitivities to blurry image features and structural advantages [6].

GhostNet [14] achieves multiple groups of feature maps with similar information by stacking inexpensive linear transformations called Ghost-Modules. It effectively utilizes redundant feature information and has shown excellent performance in image processing tasks such as image classification and object detection. Furthermore, research [16] has demonstrated that using GhostNet [14] as the backbone network, along with the corresponding Ghost-Module modules, for image deblurring tasks yields superior deblurring performance and lower model complexity compared to lightweight networks like MobileNetV3 [13].

XT-GAN chooses GhostNet [14] as the backbone network for its generator. Following the approach described in [16], the first 8 feature extraction modules (Blocks) of the network are selected as the backbone output for the deblurring task. The output channel dimension of the last layer is set to 160.

The use of multi-scale image inputs in XT-GAN satisfies the network's requirement for different receptive fields and introduces correlation at different positions for the features. Instead of directly incorporating features from images of different scales during the feature extraction process [4], XT-GAN utilizes the structure of Feature Pyramid Network (FPN) [7] as a lightweight alternative for the multi-scale approach. The outputs of different levels of feature extraction modules from the Backbone are selected as the input feature maps of the XT modules. The convolutional layers of the Backbone network serve as connections between the features, ensuring the continuity of intercorrelation and semantic information among the extracted multi-scale features.

The input to the network is a 3-channel image with a size of 256×256. It undergoes initial processing through an enhancement module consisting of a 3×3 convolutional layer with a stride of 2, normalization, and activation function, which increases the number of channels to 16 and reduces the size to 128×128. This processed image serves as the input to the Backbone network. The outputs of different levels of the Encoder's Blocks are combined pairwise in a shallow-to-deep order to obtain multi-scale features. The output channel dimensions of the multi-scale features are 24, 40, 80, and 160, while the corresponding sizes of the feature maps are 64×64, 32×32, 16×16, and 8×8, respectively. This process can be represented by the following equation:

$$ED_0 = \text{relu}(\text{norm}(\text{conv3}(I_{\text{blur}}))) \tag{1}$$

$$ED_n = ED_n(ED_{n-1}) \tag{2}$$

I_{blur} represents the input blurry image. ED_0 is enhance represents the feature map output of the enhancement module, which is also the output of the first layer of the

Encoder. conv3 represents the 3 × 3 convolution operation. norm represents the normalization operation. relu represents the ReLU activation function. ED_n represents the scale feature output of the nth layer of the Encoder. Its input is the output of the previous layer of the Encoder ED_{n-1}.

The process of fusing the multi-scale feature maps as input to the XT module can be represented by the following equations:

$$F_n = XT(ED_1, ED_2, ED_3, ED_4,) \tag{3}$$

XT represents the XT feature fusion module, and F_n represents the output of the nth XT module.

In the Decoder, the high-dimensional, high-semantic feature maps F_1, F_2, F_3, F_4 from XT are first upsampled to the same size and concatenated along the channel dimension. Then, they are added element-wise with the low-semantic feature map ED_0, and mapped to the size of the original input image through a 1 × 1 convolution. The output is activated using Tanh. We employ residual connections in the network to propagate the input to the end, aiming to capture the relevant information from the original image. This process can be represented as follows:

$$D_{\text{out}} = \text{cat}(F_1, F_2, F_3, F_4) + ED_0 \tag{4}$$

$$I_{\text{sharp}} = \tanh(\text{resize}(D_{\text{out}})) + I_{\text{blur}} \tag{5}$$

D_{out} represents the output of the Decoder, I_{sharp} denotes the generated clear image, tanh represents the activation function used, resize indicates the upsampling operation for image scale restoration, and cat denotes the channel-based concatenation operation.

2.2 Design of the Discriminator

The discriminator in XT-GAN utilizes a dual-scale structure [12], consisting of a global information capture discriminator and a local information extraction discriminator.

Research [17] has shown that using a Markovian discriminator with patch size of 70 × 70 can better promote the generation of visually superior clear images compared to using the complete image size. However, Kupyn et al.[6] demonstrated that using a dual-scale discriminator, called DoubleGAN, can complement the scene background and object motion trajectory in the image, thereby enabling the network to handle complex and diverse blurry scenes in the real world.

During the training process, ImageGAN takes in the complete image size of 256 × 256 for discrimination. PatchGAN, on the other hand, divides the image into patches of size 70 × 70 and performs discrimination on each patch separately, taking the average of the corresponding scores. Therefore, the calculation formula for the overall confidence of the target image is as follows:

$$S_{\text{out}} = \frac{1}{2}(S_{\text{I}} + \text{E}(S_{\text{P}})) \tag{6}$$

where S_{out} represents the average overall confidence of the DoubleGAN discriminator, S_{I} represents the confidence score outputted by ImageGAN, S_{P} represents the confidence scores for each patch and their summation, and E represents taking the average of the aggregated results.

2.3 XT Moudle

Pan et al. [18] pointed out that convolutional layers can provide good local information in the early stages, but in later stages, the network requires globally aware features provided by self-attention mechanisms. In XT-GAN, the Encoder's multi-scale features are fused across layers using Transformer-based XT modules, as shown in Fig. 2(a). The XT feature fusion module, illustrated in Fig. 2(b), consists of Shifted Window-based Multi-Head Self-Attention (W-MSA), and feature enhancement modules, namely Dot Transformer (DT) and Pool Transformer (PT), which collectively extract cross-scale correlation information. The output of XT, depicted in Fig. 2(c), comprises a set of pyramid features with richer semantics and more distinctive characteristics, reducing the overlap and redundancy of different scale feature information.

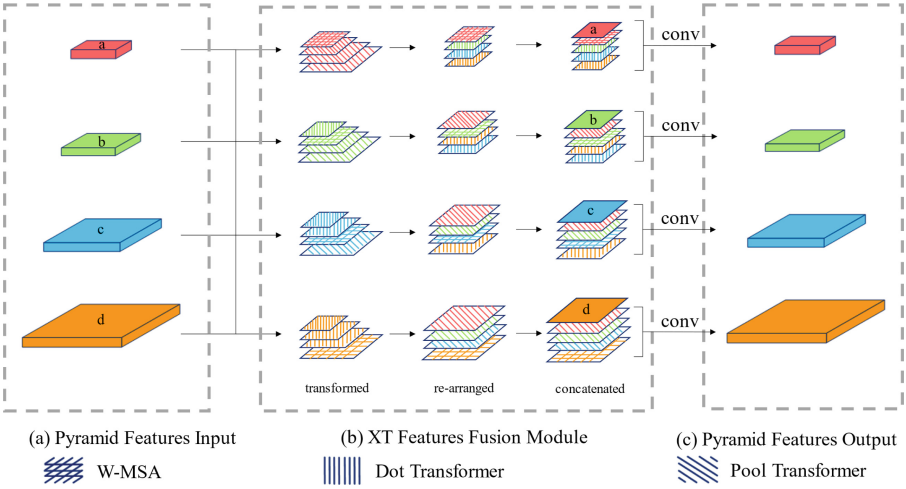

Fig. 2. XT module structure diagram.

Multi-scale structures have been proven effective in handling complex image blurring [19]. However, directly incorporating untreated multi-scale features can easily disrupt the correlation between features. In literature [20–22], stacking Transformer modules or adding convolutional layers to process the feature maps after feature fusion ensures that the information is preserved during semantic fusion. However, this approach increases the complexity of the network and introduces a large number of parameters, which can impact convergence. In XT-GAN, the Encoder extracts pure convolutional features at different stages from the GhostNet [14] network, which eliminates the need for additional convolutional processing when inputting locally spatially correlated scale features. XT utilizes parallel W-MSA and attention enhancement modules to complement global information, implicitly performing feature fusion on pyramid features. The following sections will describe the specific structure and function of XT.

ST Moudle. The Transformer was initially used in natural language processing (NLP) and relies on self-attention (SA) with a Key-Query-Value (KQV) mechanism to capture

long-range dependencies effectively. In this mechanism, K represents the key information, Q represents the query information, and V represents the corresponding values. KQV matrices are obtained through matrix transformations of the original feature map, providing different descriptions of the original features. The correlation matrix between K and Q is computed through matrix multiplication and activated by Softmax. It is then multiplied with the corresponding V. This process can be mathematically represented as follows:

$$Q, K, V = MW_{\mathrm{q}}, MW_{\mathrm{k}}, MW_{\mathrm{v}} \tag{7}$$

$$\text{Attention} = \text{Softmax}(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{d_{\mathrm{k}}}})\mathrm{V} \tag{8}$$

In the equations, W_{q}, W_{k}, W_{v} represent different weight matrices, M represents the original feature map, Attention represents the output of the self-attention (SA) mechanism, Softmax denotes an activation function, K^{T} denotes the transpose of K, and d_{k} represents the dimension of K.

To enhance the representational capacity of self-attention (SA), multi-headed self-attention (MSA) decomposes the input feature dimension into multiple subspaces, each of which corresponds to the number of attention heads. The output is the concatenation of the value vectors from each head. Each head can learn different attention patterns, thereby improving the performance of the model.

The computation of MSA involves calculating the cross-correlation between each feature value in a feature map and every other position's feature value. This requires computing the correlations of all features, resulting in a quadratic complexity growth with respect to the size of the input feature map. Consequently, it becomes impractical to handle high-resolution blurry images. To address this, literature [23] introduces Shifted Window based Self-Attention (W-MSA), which achieves good results in image restoration or denoising tasks with linear complexity. The computational complexities of MSA and W-MSA are as follows:

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C \tag{9}$$

$$\Omega(\text{W - MSA}) = 4hwC^2 + 2M^2hwC \tag{10}$$

h and w represent the height and width of the input feature map, respectively, while C represents the dimension of the feature map. M represents the window size, which is typically set to 8 by default. The computational complexity of W-MSA depends on the number of windows into which the feature map is divided. The output is the linear transformation of the Multi-head Self-Attention (MSA) within each window's subspace. The calculation process is as follows:

$$\text{Attention}_{\text{W - MSA}} = \text{resize}(\text{Attention}_n) \tag{11}$$

$\text{Attention}_{\text{W - MSA}}$ refers to the operation of applying Multi-head Self-Attention (MSA) within each window's subspace. resize represents the merging or concatenation

of the outputs of MSA performed on each window. Attention$_n$ represents the summation or aggregation of the MSA outputs from all the divided windows. Attention$_n$ represents performing MSA calculation within the n divided windows.

To address the issue of limited communication between windows in different positions, the windows are shifted and another round of W-MSA calculation is performed. A mask is used to prevent the propagation of irrelevant features, allowing only the exchange of cross-correlation information between adjacent windows. Although performing W-MSA twice increases the overall computational complexity, it transforms the problem of global computation on the feature map into localized attention calculations within associated windows. This significantly reduces the computational resource requirements for restoring high-resolution images. Additionally, W-MSA provides local information similar to convolutional kernels, while the global context is controlled by the shifted windows. The combination of local information and global context contributes to significant improvements in the deblurring task.

Feature Enhancement Module. The FPN (Feature Pyramid Network) structure, as described in [7], provides an upward path based on upsampling, enabling lower-level information to access richer semantic feature maps from higher levels. Lower-level information may have less semantic information and may not directly serve as effective features. However, it contains global statistical information that can complement the high-level network in capturing fine details.

Inspired by Zhang et al. [24], the feature enhancement module in the XT-GAN network utilizes DT to map higher-level feature maps to the current scale feature map and PT to render lower-level features onto the current scale feature map. Unlike traditional attention mechanisms that enhance the original feature map by weighted summation of attention feature maps, this approach generates new feature maps. To avoid information confusion among different semantic feature maps, the new feature maps are concatenated with the original feature map. The expectation is that the network can use information from other scales to determine the relevance of different-scale features and the semantic representation range of features at the current scale. The combination of these feature maps strengthens the multidimensional expressive power of the current scale, emphasizing its semantic information and characteristics. It implicitly models the semantic information between upper and lower layers and achieves better multi-scale information fusion compared to the FPN structure [7].

DT achieves feature enhancement by treating higher-level features as keys (K) and the current scale as queries (Q) and values (V). It calculates the dot product between K and Q to obtain the correlation matrix of their feature vectors. This correlation matrix is then multiplied element-wise with V to obtain the correlation mapping of K on V. This process can be represented by the following equations:

$$Q_{\mathrm{dt}}, K_{\mathrm{dt}}, V_{\mathrm{dt}} = M_{\mathrm{low}} W_{\mathrm{dt-q}}, M_{\mathrm{high}} W_{\mathrm{dt-k}}, M_{\mathrm{low}} W_{\mathrm{dt-v}} \tag{12}$$

$$\mathrm{Out}_{\mathrm{dt}} = \mathrm{mat}\left(\frac{K_{\mathrm{dt}} Q_{\mathrm{dt}}^{T}}{hw}, \mathrm{V}_{\mathrm{dt}}\right) \tag{13}$$

Q_{dt}, K_{dt}, V_{dt}, $W_{\mathrm{dt-q}}$, $W_{\mathrm{dt-k}}$, $W_{\mathrm{dt-v}}$ represents the matrices for K, Q, V, and their respective weight matrices in the DT operation. M_{low} represents the lower-level feature,

M_{high} represents the higher-level feature, Out_{dt} represents the output result of DT, Q_{dt}^{T} represents the transpose of Q_{dt}, and mat represents matrix multiplication.

PT achieves feature enhancement by using the lower-level feature as V, the global average pooling result of V as Q, and the current scale feature as K. It calculates the element-wise product of K and Q, and then adds this result to the downsampled V, resulting in the rendered feature map of the lower-level feature at the current scale. This process can be represented by the following equations:

$$Q_{\text{pt}}, K_{\text{pt}}, V_{\text{pt}} = \text{avg}(M_{\text{low}})W_{\text{pt - q}}, M_{\text{high}}W_{\text{pt - k}}, \text{down}(M_{\text{low}}) \tag{14}$$

$$\text{Out}_{\text{pt}} = \text{add}(Q_{\text{pt}}K_{\text{pt}}, V_{\text{pt}}) \tag{15}$$

Q_{pt}, K_{pt}, V_{pt} represents the matrices for K, Q, and V in the PT operation. $W_{\text{pt - q}}$, $W_{\text{pt - k}}$ represents the weight matrix. avg represents the global average pooling operation. down represents the downsampling operation. Out_{pt} represents the output result of PT.

2.4 Data Normalization

In image restoration tasks, which often require high-resolution outputs, the computational demand is significant. Therefore, network training is typically performed using small batches (Mini-Batch). Instance Normalization (IN) and Layer Normalization (LN) are normalization techniques that have advantages when training on Mini-Batch data. They help overcome the drawbacks of using Batch Normalization (BN), which can result in significant variations in feature variance across different batches.

In XT-GAN, the W-MSA operation in the XT module utilizes LN for training. This choice is influenced by the findings mentioned in reference [25], which suggests that LN is beneficial for training Transformers. LN is used as a normalization method to stabilize the training process and improve the performance of the W-MSA operation.

Reference [26] introduces an improved version of Instance Normalization called the Half Instance Normalization Block (HIB). HIB normalizes only half of the channels, which helps to preserve more scale information from the original features while reducing complexity. In XT-GAN, HIB is combined with other modules of the network to perform data normalization.

2.5 Loss Function

XT-GAN uses a composite generator loss function, which consists of two components: the content loss of the generator and the adversarial loss from the multi-scale discriminator. It can be represented as follows:

$$L_{\text{G}} = L_{\text{content}} + \frac{1}{2}\lambda(L_{\text{adv - full}} + L_{\text{adv - patch}}) \tag{16}$$

The content loss L_{content} is used to measure the semantic differences between the generated restoration image and the target image, capturing the discrepancy in semantic content. The adversarial loss $L_{\text{adv - full}}$ and $L_{\text{adv - patch}}$ are employed to train the model in

generating more realistic and natural restoration images. The output result L_G represents the average value of the content loss and the adversarial loss multiplied by λ and added to $L_{content}$. The value of λ, as recommended in reference [6], is typically set to 0.01, determining the weight coefficient for balancing the importance of the content loss and the adversarial loss in the overall generator loss.

XT-GAN's discriminator loss is computed based on the results of the multi-scale discriminator. It can be represented as follows:

$$L_D = \frac{1}{2}(L_{D\text{ - full}} + L_{D\text{ - patch}}) \tag{17}$$

where $L_{D\text{ - full}}$ represents the global discriminator loss value and $L_{D\text{ - patch}}$ represents the local discriminator loss value. The output result L_D is the average value of both losses, representing the overall discriminator loss from the multi-scale discriminator.

Loss Function of Generator. In image restoration, commonly used loss functions include pixel-space losses such as Mean Squared Error (MSE) or Mean Absolute Error (MAE). Reference [6] suggests that MSE loss can better reflect the similarity between pixels, allowing for better correction of color errors and texture distortions. However, using only MAE or MSE can struggle to restore high-frequency image details and may introduce abnormal artifacts [27].To address the information loss caused by using pixel-space losses, reference [28] proposes defining and optimizing a perceptual distance loss based on high-level features as part of the content loss function. Compared to traditional pixel-space losses, the perceptual distance loss based on high-level features better reflects human perception of semantic similarity in images. It helps the model learn more accurate and semantically informed restoration images, leading to better visual results.The content loss function used in this paper is represented as follows:

$$L_{content} = \alpha L_{perc} + \beta L_{mse} \tag{18}$$

where L_{perc} represents the perceptual loss and L_{mse} represents the mean squared error (MSE). The values of α and β, as recommended by Kupyn et al.[6], are typically set to 0.006 and 0.5.

Loss Function Of Discriminator. GANs are widely used in image restoration and image super-resolution tasks, aiming to improve both the discriminator's ability to distinguish real samples and the generator's ability to generate realistic fake samples. However, training GANs often faces challenges such as training difficulties and loss convergence issues, resulting in a situation where the discriminator's capability surpasses that of the generator, leading to severe gradient vanishing for the generator. The Wasserstein GAN (WGAN) loss function is commonly used as the discriminator loss in GANs. The Wasserstein GAN with Gradient Penalty (WGAN-GP) loss function is an upgraded version that introduces a gradient penalty term, which computes the Wasserstein distance between real and fake samples, encouraging smooth gradients between the adversarial networks. This helps alleviate gradient vanishing and exploding issues, eliminates the need for manual parameter tuning during training, and accelerates network convergence. Reference [6] proposes the RaGAN-LS loss function, which further enhances stability, computational efficiency, and deblurring performance compared to using WGAN-GP.

In XT-GAN, the RaGAN-LS loss function is utilized by calculating the losses of two discriminator networks. The adversarial loss L_{adv} improves the generator's ability to generate fake samples, while the discriminator loss L_{D} enhances the discriminator's ability to detect fake samples. The discriminator loss L_{D} and the adversarial loss L_{adv} can be represented as follows:

$$\begin{aligned} L_{\text{D}} &= \text{E}[(D(I_{\text{real}}) - E(D(I_{\text{fake - p}})) - 1)^2] \\ &+\text{E}[(D(I_{\text{fake}}) - E(I_{\text{real - p}}) + 1)^2] \end{aligned} \tag{19}$$

$$\begin{aligned} L_{\text{adv}} &= \text{E}[(D(I_{\text{real}}) - E(D(I_{\text{fake - p}})) + 1)^2] \\ &+\text{E}[(D(I_{\text{fake}}) - E(I_{\text{real - p}}) - 1)^2] \end{aligned} \tag{20}$$

where D represents the discriminator network, I_{real} is the label image of the input image, I_{fake} is the fake sample generated by the generator from the input image. These samples are stored in the ImagePool [29] for future use. With a probability of 50%, the discriminator has a choice between directly receiving the new input label I_{real} or the generated I_{fake}, or randomly selecting an image stored in the ImagePool [29].

3 Experimental Evaluation

The experiment was conducted on a server with a 15 vCPU Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60 GHz, 80 GB of memory, and a PTX A5000 GPU with 24 GB of VRAM. The operating system used was Ubuntu 18.04, and the CUDA version was 11.7. The deep learning framework used was PyTorch version 1.11.0. During training, the input image size was set to 256 × 256, and the number of heads (Head-Num) for W-MSA was set to 4. The batch size was set to 1, and Adam optimizer was used for training with an initial learning rate of 10^{-4}.The training process involved freezing the backbone network for the first 3 epochs and then proceeding with normal training. A linear decay learning rate schedule was applied starting from the 50th epoch, decaying the learning rate to 10^{-4} by the 2000th epoch. Data augmentation techniques used included random cropping, random motion blur, median blur, image compression, random sharpening, and random grayscale.The total training duration was 2000 epochs, which took approximately 160 h considering the hardware configuration. This duration was deemed sufficient for the model to fully converge.

3.1 Datasets

In this experiment, the synthetic GoPro dataset [1] and the real-world dataset RealBlur-J [30] were chosen to evaluate the robustness, generalization, and effectiveness of the proposed model. To reduce overfitting to specific datasets, incremental data created by Kupyn et al. [6] was used for the GoPro dataset [1]. Multiple datasets, including GoPro [6], were selected following the same data split [6], with 4400 pairs used for training and 2200 pairs for validation. This ensured that the model performed well when handling blurry images from different sources and could address a wider range of image restoration problems.For testing on real-world images, the model trained on the GoPro dataset [1]

was tested on the test set of RealBlur-J [30], which contains 980 pairs of images. This was done to verify the robustness and generalization ability of the proposed model when dealing with real-world blurry images.

3.2 Comparative Experiment

In this paper, we will compare the proposed Cross-Layer Fusion Transform Generative Adversarial Network with recent and outstanding deblurring network models on the GoPro [1] and RealBlur-J [30] test sets. We will use commonly used image restoration performance metrics, such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), as objective measures to evaluate the deblurring effectiveness of the networks. We will also consider the floating-point operations (FLOPs) and model parameters (Params) as indicators to assess the model complexity and size. Additionally, the runtime required for the model to restore a single image will serve as an objective measure of the model's computational speed. The input image size for testing is set to the standard 720×1280 dimensions in the GoPro dataset [1]. The results of the testing are presented in Table 1.

Table 1. Performance on the GoPro test dataset

Model	PSNR (dB)	SSIM	Runtimes (s)	Parms (MB)	FLOPs (G)
Sun et al. [2]	24.64	0.842	1200	54	–
Xu et al. [31]	25.10	0.890	13.41	37.1	–
DeblurGAN [5]	28.70	0.927	0.85	12.9	678.29
DeblurGAN-v2 [6]	29.55	0.934	0.35	23.8	411.34
Ghost-Deblur [16]	28.75	0.919	0.037	6.08	20.51
SRN [3]	30.10	0.932	1.6	6.8	1434.82
DeepDeblur [1]	29.23	0.916	4.33	11.7	1760.04
XT-GAN(ours)	29.13	0.923	0.068	8.5	120.4

Although the proposed model in this paper exhibits a lower PSNR compared to conventional models like SRN [3], it achieves a significant reduction of 70% in computational complexity. Moreover, it outperforms DeepDeblur [1] with an improved SSIM value by 0.007. When compared to the lightweight model Ghost-Deblur [16], which utilizes the same backbone network, the proposed XT structure enhances the collection of contextual information and improves the deblurring performance of the network. The PSNR and SSIM values are improved by 0.38 dB and 0.004, reaching 29.13 dB and 0.923, respectively.

To evaluate the generalization ability of the proposed network model, we test its effectiveness on real blurry scenes. After training on the GoPro [1] dataset, we perform testing and comparisons on the RealBlur-J [30] test set. The results are shown in Table 2.

Table 2. Performance on the RealBlur-J test dataset(trained on GoPro)

Model	PSNR (dB)	SSIM
Sun et al. [2]	–	–
Xu et al. [29]	27.14	0.8303
DeblurGAN [5]	29.97	0.834
DeblurGAN-v2 [6]	28.70	0.866
Ghost-Deblur [16]	28.25	0.846
SRN [3]	28.56	0.867
DeepDeblur [1]	27.87	0.827
XT-GAN(ours)	28.40	0.854

The XT-GAN model demonstrates good robustness in dealing with real-world blurry scenarios, surpassing the PSNR values of DeepDeblur [1], DeblurGAN [5], and Ghost-Deblur [16] by 0.53 dB, 0.43 dB, and 0.15 dB, respectively. It achieves a PSNR value of 28.40 dB and an SSIM value of 0.854, indicating stable and promising results. This indicates that the XT-GAN model is effective in handling real-world blur and can produce improved image restoration outcomes.

3.3 Subjective Deblurring Effect

Subjective Deblurring Effect of XT-GAN on Real Dataset, as shown in Fig. 3 and Fig. 4.

(a)Blur (b)DeblurGAN (c)GhostDelur (d)DeepDeblur (e)XT-GAN (f)Sharp

Fig. 3. Performance on the GoPro test dataset

Among the compared models, DeepDeblur [1] achieves the highest objective deblurring metrics, as shown in Fig. 3(d). Its restored image exhibits thicker and more distinct edges, but it tends to remove or enhance certain details in the blurry regions, resulting in overall color or structural distortions. On the other hand, the image restored by XT-GAN in Fig. 3(e) appears more natural with even coloring. In Fig. 4, XT-GAN effectively suppresses blurry artifacts and restores the desired contours.

Fig. 4. Performance on the RealBlur-J test dataset

3.4 Ablation Experiment

To validate the effectiveness of the proposed XT module in different positions within the generator network, the combined effectiveness of the three module variations within the XT module, and the integration effect of the XT module's multi-head self-attention with the network, ablation experiments were conducted on the incremental GoPro dataset [1]. The input size for the images was set to 720 × 1280. Apart from the attention hyperparameter experiments, the Head-Num for W-MSA was uniformly set to 4.

Structural Validity. XT-v1 represents the XT structure using MSA. XT-v2 represents the optimized structure using W-MSA [23]. ADD-XT indicates the fusion of the XT structure with the output features of the FPN upsampling structure in an additive manner. FPN-XT indicates the computation of XT features within the FPN upsampling fusion structure. The effectiveness of the model structures is shown in Table 3.

Table 3. Effectiveness of Network Structures: Ablation Tests

Model	PSNR (dB)	SSIM
XT-v2	29.13	0.923
XT-v1	28.89	0.921
ADD-XT	28.59	0.920
FPN-XT	28.69	0.922
Ghost-Deblur [16]	28.75	0.919

From Table 3, it can be observed that XT can effectively serve as an independent structure for cross-scale feature fusion and provide richer deblurring information compared to using FPN upsampling fusion [16]. However, when XT is combined with the FPN structure features proposed in [16] either through addition (ADD-XT) or sequential combination (FPN-XT), redundant modules with similar functionality are introduced, leading to a decrease in deblurring performance with a PSNR reduction of 0.16dB and 0.06dB, respectively. XT-v2, on the other hand, achieves the best deblurring performance and further enhances the results obtained with XT-v1.

Module Validity. The effectiveness of the individual components of the additional Transformer introduced in this paper was evaluated through ablation experiments, and the results are presented in Table 4.

Table 4. Effectiveness of XT Structures: Ablation Tests

FPN	ST	DT	PT	PSNR	SSIM
√	–	–	–	28.75	0.919
–	√	√	√	29.13	0.923
–	√	√	–	29.06	0.924
–	√	–	√	28.83	0.921
–	–	√	√	27.79	0.921
–	√	–	–	29.02	0.921
–	–	√	–	28.89	0.922
–	–	–	√	28.61	0.917

According to the data in Table 4, it can be observed that using ST, DT, and PT individually resulted in changes in PSNR values of 0.27 dB, 0.14 dB, and −0.14 dB, respectively. The inclusion of ST alone provided a significant amount of informative content and achieved similar performance. Although the individual use of DT and PT did not yield satisfactory results, they contributed additional scale information to ST. The combination of ST, DT, and PT achieved the best overall performance. Compared to using interpolation-based upsampling fusion [16], the XT structure yielded an improvement of 0.38 dB in PSNR and 0.004 in SSIM, indicating enhanced deblurring performance.

Hyperparameters Setting. The difference between MSA (Multi-Head Self-Attention) and SA (Self-Attention) lies in the number of Head-Num, which determines the total number of feature maps output by MSA. Therefore, it is important to choose a reasonable number of Head-Num in MSA. Table 5 presents the effectiveness of different Head-Num values for the network.

It can be observed that the network achieves the best deblurring performance when the Head-Num value is set to 16. It achieves the highest PSNR and SSIM evaluation values of 29.20 dB and 0.925, respectively. However, considering the performance on

Table 5. Effectiveness of different Head-Num values in XT's multi-head attention

Model	PSNR (dB)	SSIM
1	29.02	0.923
2	29.08	0.924
4	29.13	0.923
8	29.01	0.922
16	29.20	0.925
32	29.18	0.924

the RealBlur-J [31] test set, where Head-Num = 4 performs the best, we selected it as the optimal value for the hyperparameter setting.

4 Conclusion

This paper proposes a multi-scale fusion generative adversarial network based on the Transformer architecture, which includes an Encoder-Decoder structure. The Encoder extracts the pyramid features from the GhostNet convolutional network as multi-scale feature inputs. The XT structure combines global and local information and the Decoder restores the image to the original scale. This structure has linear complexity, with a floating-point computation of 120.4G when processing high-resolution images of size 720×1280, and a runtime of only 0.068 s, outperforming conventional networks such as SRN [3] and DeblurGAN-v2 [6]. The effectiveness and robustness of the algorithm are verified on the GoPro and RealBlur-J datasets. The next steps will focus on further optimizing the structure of the XT module and developing backbone networks with higher computational efficiency and better performance tailored to different blurry scenarios.

References

1. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3883–3891 (2017)
2. Sun, J., Cao, W., Xu, Z., et al.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 769–777 (2015)
3. Tao, X., Gao, H., Shen, X., et al.: Scale-recurrent network for deep image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8174–8182 (2018)
4. Cho, S.J., Ji, S.W., Hong, J.P., et al.: Rethinking coarse-to-fine approach in single image deblurring. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4641–4650 (2021)
5. Kupyn, O., Budzan, V., Mykhailych, M., et al.: Deblurgan: blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8183–8192 (2018)

6. Kupyn, O., MaPTyniuk, T., Wu, J., et al.: Deblurgan-v2: deblurring (orders-of-magnitude) faster and better. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8878–8887 (2019)
7. Lin, T.Y., Dollár, P., Girshick, R., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
8. Zhang, K., Luo, W., Zhong, Y., et al.: Deblurring by realistic blurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2737–2746 (2020)
9. Zhang, H., Patel, V.M.: Density-aware single image de-raining using a multi-stream dense network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 695–704 (2018)
10. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
12. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion[J]. ACM Trans. Graph. (ToG) **36**(4), 1–14 (2017)
13. Howard, A., Sandler, M., Chu, G., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)
14. Han, K., Wang, Y., Tian, Q., et al.: Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1580–1589 (2020)
15. Xia, X., Xu, C., Nan, B.: Inception-v3 for flower classification. In: 2017 2nd International Conference on Image, Vision and Computing (ICIVC), pp. 783–787. IEEE (2017)
16. Liu, Y., Haridevan, A., Schofield, H., et al.: Application of ghost-DeblurGAN to fiducial marker detection. In:2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6827–6832. IEEE (2022)
17. Isola, P., Zhu, J.Y., Zhou, T., et al.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
18. Pan, X., Ge, C., Lu, R., et al.: On the integration of self-attention and convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 815–825 (2022)
19. Liu, S., Wang, H., Wang, J., et al.: Blur-kernel bound estimation from pyramid statistics. IEEE Trans. Circuits Syst. Video Technol. **26**(5), 1012–1016 (2015)
20. 李现国,李滨.基于 Transformer 和多尺度 CNN 的图像去模糊[J/OL].计算机工程1–10 (2023).https://doi.org/10.19678/j.issn.1000-3428.0065513
21. 杨浩,周冬明,赵倩.结合梯度指导和局部增强 Transformer 的图像去模糊网络[J/OL].小型微型计算机系统 1–10 (2023).https://doi.org/10.20009/j.cnki.21-1106/TP.2022-0344
22. 刘婉春,景明利,王子昭等.基于 Transformer 和双残差网络的图像去模糊算法研究[J].信息技术与信息化 **274**(01), 217–220 (2023)
23. Liang, J., Cao, J., Sun, G., et al.: Swinir: image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1833–1844 (2021)
24. Zhang, D., Zhang, H., Tang, J., et al.: Feature pyramid transformer. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, PaPT XXVIII 16, pp. 323–339. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-58604-1_20

25. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is woPTh 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
26. Chen, L., Lu, X., Zhang, J., et al.: Hinet: half instance normalization network for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 182–192 (2021)
27. Ledig, C., Theis, L., Huszár, F., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690 (2017)
28. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, PaPT II, vol. 14, pp. 69–711. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-46475-6_43
29. Shrivastava, A., Pfister, T., Tuzel, O., et al.: Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2107–2116 (2017)
30. Rim, J., Lee, H., Won, J., et al.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, PaPT XXV, vol. 16, pp. 184–201. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-58595-2_12
31. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1107–1114 (2013)