applied sciences

MDPI

Article

Image Deblurring Based on an Improved CNN-Transformer Combination Network

Xiaolin Chen *, Yuanyuan Wan, Donghe Wang and Yuqing Wang

Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
* Correspondence: cl.hong@163.com

Abstract: Recently, using a CNN has been a common practice to restore blurry images due to its strong ability to learn feature information from large-scale datasets. However, CNNs essentially belong to local operations and have the defect of a limited receptive field, which reduces the naturalness of deblurring results. Moreover, CNN-based deblurring methods usually adopt many downsample operations, which hinder detail recovery. Fortunately, transformers focus on modeling the global features, so they can cooperate with CNNs to enlarge the receptive field and compensate for the details lost as well. In this paper, we propose an improved CNN-transformer combination network for deblurring, which adopts a coarse-to-fine architecture as the backbone. To extract the local features and global features simultaneously, the common methods are two blocks connected in parallel or cascaded. Different from these, we design a local-global feature combination block (LGFCB) with a new connecting structure to better use the extracted features. The LGFCB comprises multi-scale residual blocks (MRB) and a transformer block. In addition, we adopt a channel attention fusion block (CAFB) in the encoder path to integrate features. To improve the ability of feature representation, in the decoder path, we introduce a supervised attention block (SAB) operated on restoration images to refine features. Numerous experiments on GoPro and RealBlur datasets indicated that our model achieves remarkable accuracy and processing speed.

Keywords: image deblurring; coarse-to-fine strategy; transformer; CNN-transformer combination

Citation: Chen, X.; Wan, Y.; Wang, D.; Wang, Y. Image Deblurring Based on an Improved CNN-Transformer Combination Network. *Appl. Sci.* **2023**, *13*, 311. https://doi.org/10.3390/app13010311

Academic Editors: Hyo Jong Lee, Yong Yang and Byung Gyu Kim

Received: 1 November 2022
Revised: 20 December 2022
Accepted: 22 December 2022
Published: 27 December 2022

1. Introduction

Image deblurring aims to restore a high-quality image by removing blurring degradations [1]. Due to the object movements, the limitations of cameras, or intricate light conditions, different degrees of blurring degradations are inevitably introduced during the image-acquisition process. Since these degradations usually lead to unsatisfactory visual perception, they consequently hinder subsequent high-level vision tasks where sharp images can perform better. Moreover, each degraded image may have numerous restoration possibilities as it belongs to an ill-posed problem [2]. Therefore, image de-blurring is a challenging but significant research work.

Due to neural networks' advantage of learning features from large-scale datasets, deep-learning-based methods play a dominant role in image-restoration fields. At first, most deep-learning-based deblurring algorithms commonly focused on utilizing the neural network to estimate the blur kernel and then used the blur kernel to restore images [3–6]. As a result, these kinds of methods heavily depend on the prediction of blur kernels and are not suitable for blurry images caused by various blur kernels. Afterward, with the rapid development of deep learning, it was found that neural networks are capable of much more than blur kernel estimation. Thus, recent deep-learning-based models have started to adopt an end-to-end training strategy, which can directly obtain the complex mappings between sharp and blurry images [7–19].

Among the existing deblurring models, convolution neural networks (CNNs) have been frequently employed to encode semantic information by stacking many convolution

layers and downsample operations. However, the CNN methods still have limited receptive fields and the downsample operations can hinder detail recovery. In addition, transformers explore the potential of self-attention and perform well in establishing long-range pixel-wise or channel-wise relationships. Unfortunately, transformers' ability to obtain local features is not as strong as CNNs. In general, convolution has powerful local modeling capabilities [20] and usually models the relationships between neighborhood pixels. In contrast, a transformer is a global operation that models the relationships between all pixels [20]. Generally speaking, local features and global features can cooperate to extract more detailed image features. However, how to combine the advantages of CNNs and transformers is worth investigating.

Aiming at this goal, this paper proposes a new network. Like most deblurring models, we also utilized the encoder-decoder structure to extract multi-scale features, and then recover images by the coarse-to-fine approach. To fully integrate the multi-scale input images and features from the previous stage, we designed the channel attention fusion block (CAFB) in the encoder path. Specifically, we devised a local-global feature combination block (LGFCB) to extract local and global features. The LGFCB is composed of two parts: three multi-scale residual blocks (MRB) and a transformer block. The MRB is a designed CNN structure, and the transformer block is employed to explore long-range relationships of images. The CNN branch and transformer branch extract features in parallel paths. Then, we introduce cross-path feature fusion so that we can better use the advantages of the CNN and transformer. In addition, we adopt a supervised attention block (SAB) in the decoder path, which utilizes the restoration-image-producing attention map to supervise and obtain more informative features.

In this paper, the main contributions are as follows:

An elaborately designed local-global feature combination block (LGFCB) composed of multi-scale residual blocks (MRB) and transformer blocks. This combination is beneficial for retaining local image details and exploring long-range global features.

A channel attention fusion block (CAFB) fuses features and can improve the capability of feature representation.

An efficient, supervised attention block (SAB) to use the restoration image to obtain more informative features.

We propose an improved CNN-transformer combination network for image de-blurring and demonstrate the effectiveness of our model via extensive experiments on datasets.

2. Related Work

2.1. Deep Learning Deblurring

Recently, deep learning has been the main approach in the image-deblurring field. Sun et al. [6] proposed a CNN-based model to estimate the blur kernels to remove blurriness. However, the idea that every blurry image is only caused by one blur kernel is too idealized. In fact, most blur problems may be influenced by multi-blur kernels. Thus, it is hard to predict the number of real blur kernels even when using CNN methods. Later, Nah et al. proposed DeepDeblur [7], the pioneer of direct adoption of the coarse-to-fine structure to recover image sharpness. However, this method suffered from a high computation time; this is because the design did not share parameters across multi-scales. To address this issue, the encoder-decoder structure with skip connection was introduced to share parameters and capture context, such as MPRNet [2] and MIMO-UNet+ [8]. In addition, Kupyn et al. introduced DeblurGAN [9] and DeblurGAN-v2 [10] successively by employing a generative adversarial approach. Zou et al. designed SDWNet [16] with wavelet transformation and some dilated convolutions to enlarge the receptive fields. In addition, Tsai et al. proposed BANet [17] by adopting the multi-kernel strip pooling attention structure to extract multi-scale features. These models have achieved promising performance; however, they failed to obtain the global feature, which is also important in the image-deblurring field. Only considering local features limits the accuracy of the reconstruction results.

2.2. Vision Transformer

The transformer model was first used in natural language tasks [21,22]. Due to its strong capability to learn long-range dependencies between pixels or channels, the transformer has also been adopted for low-level computer vision tasks, including image enhancement [20], image recognition [23,24], and object detection [25,26]. The vision transformer [24] uses the image like a language sequence by introducing the concept of a patch. That is, the input image is divided into patches one by one, and then uses the transformer structure to obtain their relationships. However, ignoring convolution completely is not a good idea since the transformer only relies on global-level attention but does not capture local fine-grained details. As far as we know, local features and global information are both important for obtaining high-quality image reconstruction. Therefore, how to effectively combine these features is the core goal of our work.

3. Proposed Approach

3.1. Framework

As shown in Figure 1, the model is a structure with three parts: the encoder path, the skip connection, and the decoder path. Like the MIMO-Unet+ [8], we also rescaled the original blur images to a different resolution, input them in different levels of encoder paths, and restored images at a different level of decoder path, respectively. In general, the encoder path is designed to extract local and global features. The decoder path is used for image reconstruction. Meanwhile, the skip connections are bridges to achieve feature integration between encoder and decoder paths. To better describe the model framework, we define I_1, I_2, I_3 as the input blurry images with three resolutions and $\hat{S}_1, \hat{S}_2, \hat{S}_3$ are the corresponding restoration images at each scale, respectively.

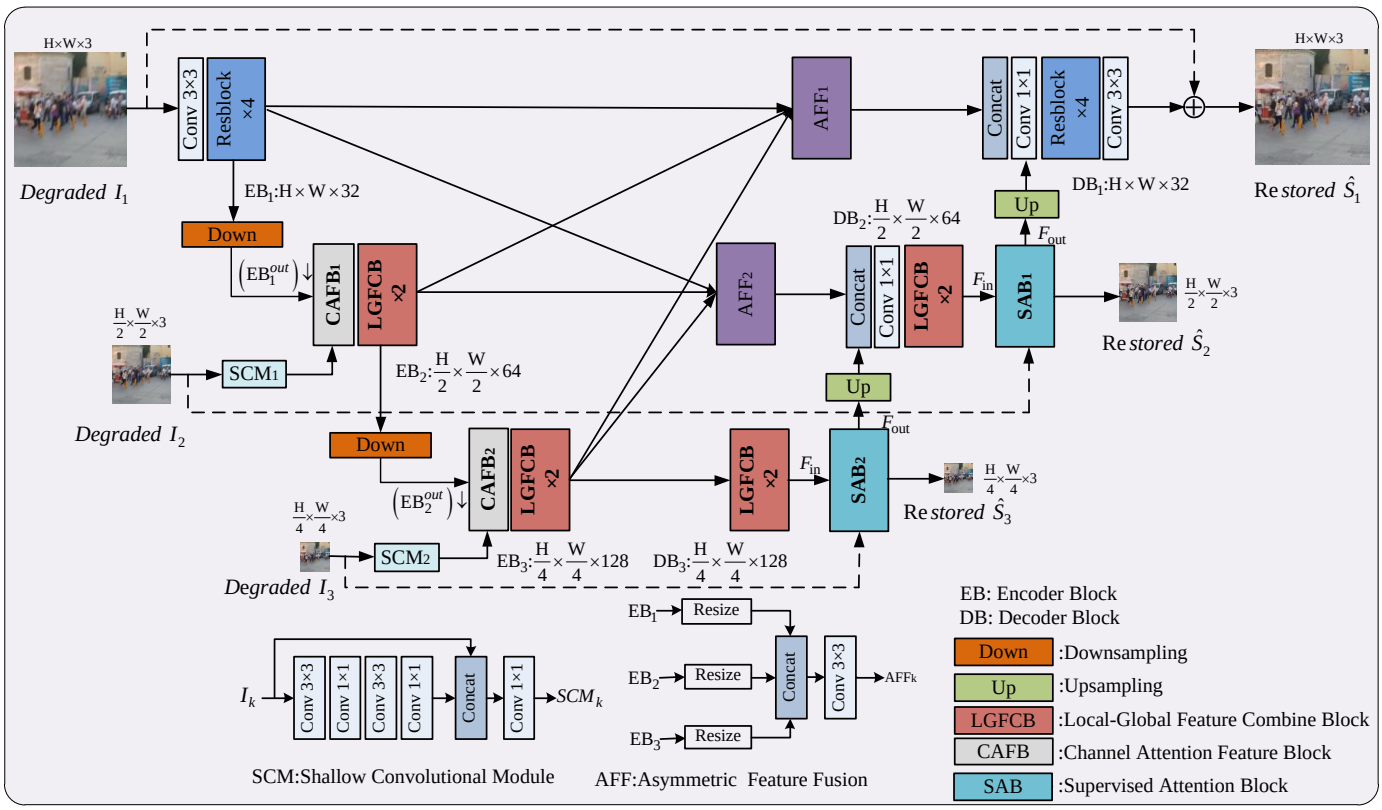


Figure 1. The whole structure of the proposed deblurring network.

(a) Encoder Path

As mentioned above, our proposed model adopts a coarse-to-fine structure, which has multi-scale input and multi-scale output. This strategy has been employed by numer-

ous CNN-based deblurring models and has already demonstrated its effectiveness. Given a blurred image $I_1 \in \mathbb{R}^{H\times W\times 3}$, we can take rescale operations to obtain $I_2 \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$ and $I_3 \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 3}$ successively. First, we input the I_1 to the model and apply a 3×3 convolutional layer and four residual blocks as Encoder Block 1 (EB1) to extract features. After that, the extracted feature needs to pass two more complicated encoder blocks. Each encoder block includes a downsample layer, a shallow convolutional module (SCM), a channel attention fusion block (CAFB), and two local-global feature combination blocks (LGFCB). Next, we can start to introduce the complicated encoder block in detail. First, supposing $k = 2, 3$, the previous encoder output EB_{k-1}^{out} passes a downsample operation which consists of a 3×3 kernel convolutional layer with a stride of two. Meanwhile, we can use SCM_{k-1} to extract shallow features of rescaled blur image I_k. Then, to fuse the SCM_{k-1}^{out} and the downsampled EB_{k-1}^{out}, we can design a channel attention fusion block (CAFB), which can selectively emphasize or suppress the feature from the previous level. After, different from MIMO-Unet+ [8], which employs eight residual blocks as the core part of each encoder level, we can specially design a local-global feature combination block (LGFCB) which can obtain local details and global features simultaneously. It is important to note that the SAM is a module used in MIMO-Unet+ [8]; the detailed structure of SAM is shown in Figure 1.

(b) Skip Connection

In general, skip connections exist in most conventional U-Net structures. However, they usually only transmit the current scale feature from an encoder to decoder paths. To achieve feature communication between different levels, our model adopts asymmetric feature fusion (AFF), as shown in Figure 1. Each AFF integrates all the encoder outputs ($EB_k^{out}, k = 1, 2, 3$) with convolutional layers and allows feature information to be transmitted from different scales. Then, the output of AFF is delivered to the corresponding decoder paths. The detailed process can be formulated as follows:

$$AFF_1^{out} = AFF_1(EB_1^{out}, (EB_2^{out})^{\uparrow}, (EB_3^{out})^{\uparrow}), \quad (1)$$

$$AFF_2^{out} = AFF_2((EB_1^{out})^{\downarrow}, EB_2^{out}, (EB_3^{out})^{\uparrow}), \quad (2)$$

The ($\uparrow$) represents upsample and the ($\downarrow$) is downsample operation. Note that AFF is a module used in MIMO-Unet+ [8].

(c) Decoder Path

The decoder paths focus on feature utilization and reconstructing sharper images. As shown in Figure 1, the EB_3^{out} is transmitted to Decoder Block 3 (DB_3) directly, and like the encoder stages, DB_2 and DB_3 also adopt two LGFCBs. In addition, they employ a supervised attention block (SAB) to refine the feature information before passing to the next stage and outputting the restored image at different scales simultaneously. We use a pixelshuffle as the upsample operation to enlarge the output feature. Compared with transpose convolution, which has the disadvantage of resulting in a checkerboard pattern, pixelshuffle can alleviate the problem and produce a high-quality image. In DB_1, we also adopt four residual blocks and a 3×3 convolutional layer and convert the feature into a final restoration image $\hat{S}_3$, with $\hat{S}_3 \in \mathbb{R}^{H\times W\times 3}$.

3.2. Local Global Feature Combination Block

The LGFCB is designed for extracting local details and global features. To better take advantage of a CNN and transformer, we explore an elaborate structure, as shown in Figure 2. Given the input feature x, we adopt two parallel paths to obtain local and global features simultaneously. Among them, the multi-scale residual block (MRB) is a CNN structure for extracting local features, and the transformer block is for extracting global features. To make further use of the extracted features, we design cross-path fea-

ture fusion to introduce global or local features during the feature-extraction process of each path. Thus, the two paths are not independent anymore. When we obtain global features, we introduce MRB_1^{out} which can attract local attention. Additionally, while obtaining local features, we fuse the transformer block output before MRB_3 and integrate more global features. Finally, we concentrate the MRB_3^{out} and transformer block output together, then apply a 1×1 convolutional layer to reduce the channel number.

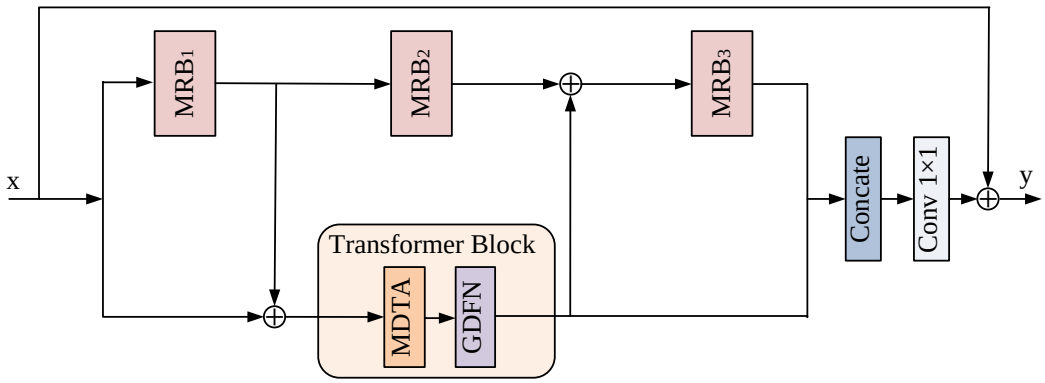
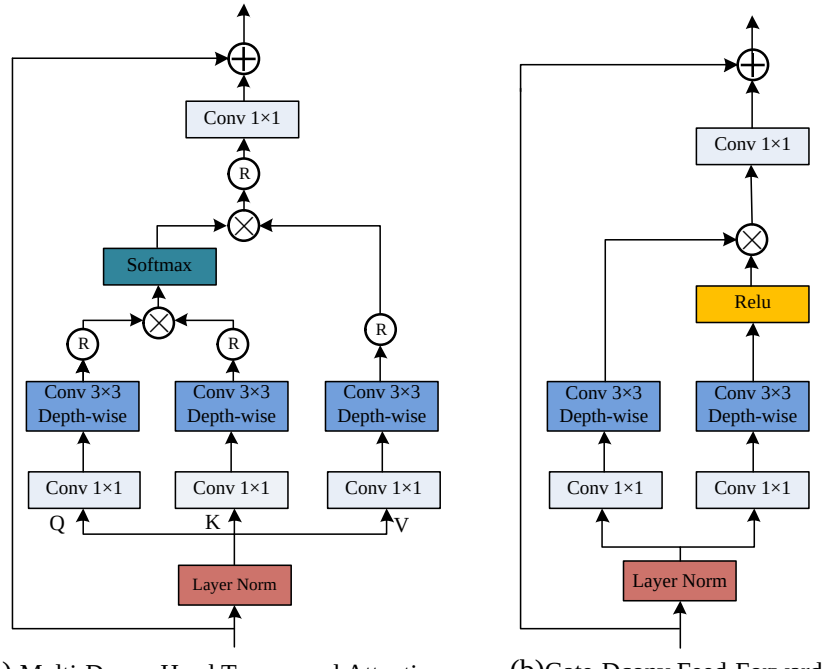


Figure 2. The structure of the proposed Local Global Feature Combination Block.

(a) Transformer Block

The transformer consists of multi-head attention and feed-forward; in addition, it is well-known for establishing long-term dependence of an image. However, to some extent, the high computation and large memory usage restrict its wide application. In normal cases, the computation complexity is mainly attributable to the self-attention (SA) layer, which is the core of the multi-head attention part. To ameliorate this problem, we adopt multi-dconv head transposed attention (MDTA) motioned in [18], but with different numbers of attention heads. Compared with applying SA in the spatial dimension, MDTA employs SA between channels to generate attention feature maps and then encode the global features. Furthermore, we use depth-wise convolutions to obtain more contextual information ahead of obtaining attention maps, as depicted in Figure 3a.



(a) Multi-Dconv Head Transposed Attention (b)Gate-Dconv Feed-Forward

Figure 3. The structure of the transformer block.

The MDTA first crosses a layer normalized and projects query (Q), key (K), and value (V) ($Q, K, V \in \mathbb{R}^{C\times H\times W}$) by using a 1×1 convolutional layer. Subsequently, a 3×3 depth-wise convolutional layer is used for obtaining spatial content at the channel-level. The input feature $X \in R^{C\times H\times W}$ and $\hat{X} \in \mathbb{R}^{C\times H\times W}$ are obtained from a layer-normalized operation.

$$Q = H^{3\times 3}_{\text{dconv}}(H^{1\times 1}_{\text{conv}}(\hat{X})), \tag{3}$$

$$K = H^{3\times 3}_{\text{dconv}}(H^{1\times 1}_{\text{conv}}(\hat{X})), \tag{4}$$

$$V = H^{3\times 3}_{\text{dconv}}(H^{1\times 1}_{\text{conv}}(\hat{X})), \tag{5}$$

Next, we need to capture global attention by obtaining the correlation between Q and K. The common method involves reshaping Q and K into $\hat{Q} \in \mathbb{R}^{C\times HW}$,$\hat{K} \in \mathbb{R}^{HW\times C}$. Applying a dot product interaction generates a transposed-attention map with a size of $\mathbb{R}^{C\times C}$. Then, we can reshape V into $\hat{V} \in \mathbb{R}^{C\times HW}$ and multiply the attention map with $\hat{V}$ to obtain a feature map $X_{weight} \in \mathbb{R}^{C\times HW}$. In the end, we reshaped X_{weight} into $\hat{X}_{weight} \in \mathbb{R}^{C\times H\times W}$ and adopted a 1×1 convolutional layer. Before the softmax operation, we applied $\sqrt{\alpha}$, which is a temperature parameter [18] used to control the magnitude of the dot product of $\hat{Q}$ and $\hat{K}$. The above procedure can be defined as:

$$X_{weight} = Softmax(\hat{Q}\cdot\hat{K}/\sqrt{\alpha})\cdot\hat{V}, \tag{6}$$

$$Y = H^{1\times 1}_{\text{conv}}(\text{Reshape}(X_{weight})), \tag{7}$$

In the meantime, the depth-wise convolutional was also used in the gate-donv feed-forward network (GDFN). As shown in Figure 3b, after layer normalization, we adopted a depth-wise convolution to obtain more information from spatially nearby pixel positions. In addition, we also used the element-wise product of two parallel paths, one of which passes through a RELU activation function. Assuming that x is the input feature, the GDFN is:

$$\hat{x} = H^{3\times 3}_{\text{dconv}}(H^{1\times 1}_{\text{conv}}(x)), \tag{8}$$

$$Y = H^{1\times 1}_{\text{conv}}(\hat{x}\cdot relu(\hat{x})), \tag{9}$$

(b) Multi-Scale Residual block

As is well-known, both global features and local features are of great significance for image deblurring. In general, multi-scale receptive fields are beneficial for extracting finer features. Thus, inspired by the inception in [27,28], we designed a multi-scale residual block to obtain multi-scale features, as shown in Figure 4.

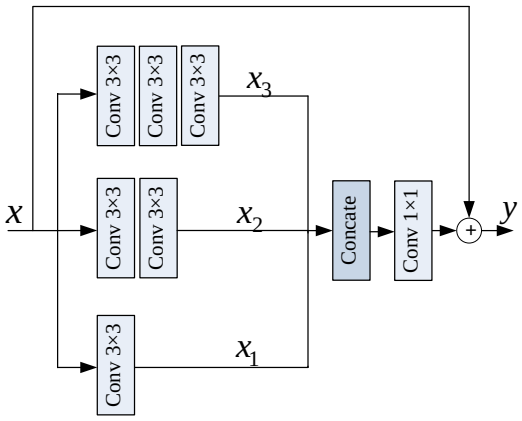


Figure 4. The structure of Multi-Scale Residual Block.

Different from the original inception structure, we replaced one 5×5 convolutional layer with two 3×3 convolutional layers and applied three 3×3 convolutional layers

instead of one 7×7 convolutional layer as well. In this way, we can greatly reduce the model parameters while still having the same receptive fields. Given the input x, we need to extract multi-scale features in three parallel paths, different from [28]. Using the element-wise summation to fuse the feature, we applied the concatenate operation to fuse features of three paths and used a 1×1 convolutional layer to reduce the channel numbers. Finally, we added a residual connection to fully capture the feature and make the training process stable at the same time. The above process can be expressed as:

$$x_1 = H_{\text{conv}}^{3\times3}(x), \tag{10}$$

$$x_2 = H_{\text{conv}}^{3\times3}\left(H_{\text{conv}}^{3\times3}(x)\right), \tag{11}$$

$$x_3 = H_{\text{conv}}^{3\times3}\left(H_{\text{conv}}^{3\times3}\left(H_{\text{conv}}^{3\times3}(x)\right)\right), \tag{12}$$

$$y = H_{\text{conv}}^{1\times1}(H_{\text{cat}}(x_1, x_2, x_3)) + x, \tag{13}$$

3.3. Channel Attention Fusion Block

In the encoder path, $\left(\text{EB}_{\text{k}}^{\text{out}}\right)\downarrow$ is the downsample feature of the output of encoder stage k, and $\text{SCM}_{\text{k}}^{\text{out}}$ is the output from the shallow feature extraction of I_k; they are in the same scale and integrating these features together is an important part of the coarse-to-fine strategy. To fully integrate the features of $\left(\text{EB}_{\text{k}}^{\text{out}}\right)\downarrow$ and $\text{SCM}_{\text{k}}^{\text{out}}$, we introduced the channel attention fusion scheme to obtain better feature representation capabilities.

As depicted in Figure 5, to better use the $\left(\text{EB}_{\text{k}}^{\text{out}}\right)\downarrow$ feature, we introduced channel attention to pay more attention to important feature channels after passing through a 3×3 convolutional layer. Then, the $\text{SCM}_{\text{k}}^{\text{out}}$ was multiplied with the attention mask to refine the shallow feature from I_k, and then the multiplied feature passed through a 3×3 convolutional layer. Finally, we used an element-wise summation operation to fuse the refined $\text{SCM}_{\text{k}}^{\text{out}}$ and $\left(\text{EB}_{\text{k}}^{\text{out}}\right)\downarrow$ features.

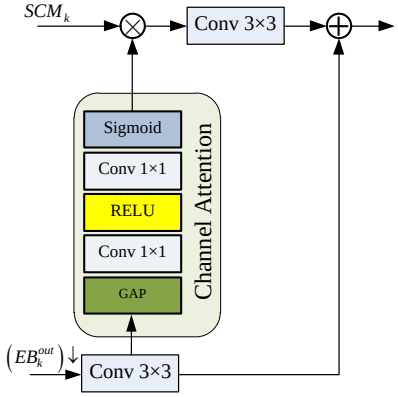


Figure 5. The structure of Channel Attention Fusion Block.

3.4. Supervised Attention Block

In the coarse-to-fine restoration strategy, we must predict the image at different scales. To improve the effectiveness of feature propagation, we designed a supervised-attention block by using the restoration image to refine the feature before being transferred to the next stage. As illustrated in Figure 6, the input feature is passed into two 3×3 convolutional layers in two different paths. One path is integrated with the degraded blur image I_k and generates the restored image $\hat{S}_k$. Then, the restored image passes through a 3×3 convolutional layer and follows a channel attention block. Next, the attention masks are applied to refine the feature of another path. We used the ground-truth image to supervise and optimize the restored result, and the restored image can be utilized to filter or empha-

size the features at the current stage in turns. Finally, we added a residual connection with the refined feature before transmitting the feature to the next stage.

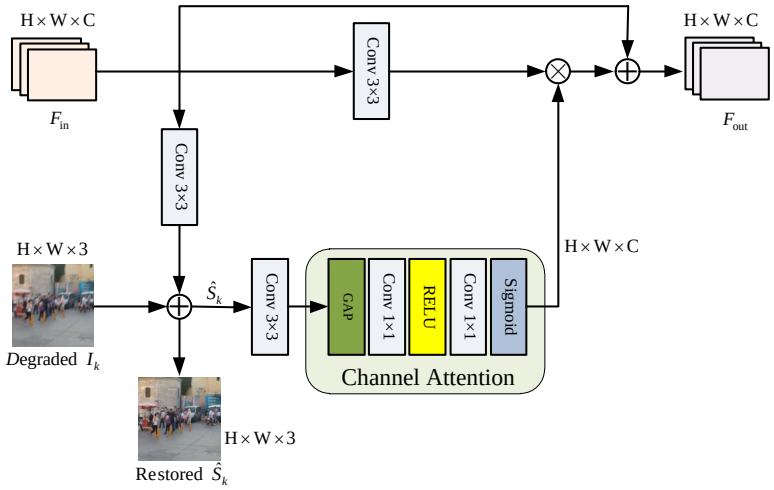


Figure 6. The structure of Supervised Attention Block.

3.5. Loss Function

Based on the coarse-to-fine strategy, our model consists of three stages, and each stage outputs a restored image. Thus, we optimized the model with multi-scale loss as well. The loss function adopts three kinds of losses: multi-scale content loss, multi-scale frequency reconstruction loss [8], and multi-scale perceptual loss. We can suppose that the I_k is a ground-truth image in stage k, and the corresponding $\hat{S}_k$ is the corresponding restored a more image in stage k.

(1) Multi-scale content loss: we minimized the content loss between the ground truth and the predicted image with the Charbonnier loss [29] function. Minimizing the content loss gradually can restore more accurate results. ε was set to 10^{-3}.

$$L_{content} = \sum_{k=1}^{3} \sqrt{\left\| \hat{S}_k - S_k \right\|_1 + \varepsilon^2}, \tag{14}$$

(2) Multi-scale frequency reconstruction loss: The blurry image mainly lost the high-frequency information [8]; therefore, it is also important to reduce loss in the frequency domain. In this case, we employed the fast Fourier transformer (referred to as F) to obtain the $L1$ loss as the frequency loss between the ground-truth image and the predicted output.

$$L_{fft} = \sum_{k=1}^{3} \left\| F(\hat{S}_k) - F(S_k) \right\|_1, \tag{15}$$

(3) Multi-scale perceptual loss: to further obtain perceptually satisfactory results, we used pre-trained VGG-19 [30] as the feature extractor. Like the multi-scale frequency reconstruction loss, we also adopted the $L1$ function to measure the percentage of loss between the two images.

$$L_{precent} = \sum_{k=1}^{3} \left\| \varphi(\hat{S}_k) - \varphi(S_k) \right\|_1, \tag{16}$$

Overall, the whole loss function can be expressed as follows, where λ_1 is set to 0.1 and λ_2 is set to 0.01.

$$L = L_{content} + \lambda_1 L_{fft} + \lambda_2 L_{precent}, \tag{17}$$

4. Experiments and Analysis

4.1. Dataset and Implementation Details

We used the GoPro dataset [7] and Realblur [31] dataset for training and testing. The GoPro dataset contains 2103 pairs of blurry and sharp images for training and 1111 pairs for testing. To assess the generalization performance of our model, we directly applied our GoPro-trained model on part of the ReaBlur test dataset, which included 980 pairs of images.

We trained all the models with the Pytorch framework. For data pretreatment, we randomly cropped the image to 256 × 256, and then horizontally flipped it with 0.5 probability. We trained 3000 epochs and the batch size was set to four. The initial learning rate was 1×10^{-4}, then we adopted the Cosine Annealing strategy [32] to steadily decrease the learning rate to 1×10^{-6}, with three epochs for warming up. Moreover, our experiments were conducted on a computer with one TITAN RTX GPU.

4.2. Experimental Results

(a) Quantitative Analysis: We adopted the peak-signal-to-noise-ratio (PSNR) and structural similarity (SSIM) to evaluate the image quality. Meanwhile, the parameters indicate that a lightweight and effective model is the current research trend. Considering our model is a kind of CNN-transformer combination structure, we compared our model with CNN-based models [2,7,8,10–17,19] and a transformer-based model [18]. Table 1 shows a comparison with the advanced models based on the GoPro [7] dataset and the RealBlur [31] dataset.

Table 1. Deblurring results of the advanced deblurring models, our model is trained on the GoPro [7] dataset and directly evaluates the RealBlur [31] test dataset.

Model	Method	GoPro		RealBlur		Params
		PSNR	SSIM	PSNR	SSIM	(M)
CNN-based models	Nah et al. [7]	29.08	0.914	27.87	0.827	11.7
	Zhang et al. [11]	29.19	0.931	27.80	0.847	9.2
	DeblurGAN-V2 [10]	29.55	0.934	28.70	0.866	60.9
	SRN [12]	30.26	0.934	28.56	0.867	6.8
	DBGAN [15]	31.10	0.942	24.93	0.745	11.6
	MT-RNN [14]	31.15	0.945	28.44	0.862	2.6
	DMPHN [13]	31.20	0.940	28.42	0.860	21.7
	BANet [17]	32.44	0.957	-	-	85.65
	SDWNet [16]	31.26	0.966	28.61	0.867	7.2
	FMD-cGAN [19]	28.33	0.962	-	-	1.98
	MIMO-UNet+ [8]	32.45	0.957	27.63	0.837	16.1
	MPRNet [2]	32.66	0.959	28.70	0.873	20.1
Transformer-based model	Restomer [18]	32.92	0.961	28.96	0.879	26.12
CNN-Transformer	Our model	32.68	0.962	28.73	0.881	15.6

From Table 1, we can see that our model outperforms SSIM results in the most deblurring models. Notably, we only trained our model on the GoPro dataset and then directly used it to test the RealBlur dataset, proving that our model has a good generalization ability. The PSNR results are also better than most of the CNN-based models. Compared with the transformer-based model Restomer [18], our model achieves better SSIM results with smaller model parameters. Moreover, some CNN-based models have small parameters, but the evaluation results are poor, and some achieve good results with larger model sizes. In contrast, our model balances performance and parameters; it has comparable evaluation results and relatively small model parameters.

(b) Qualitative Analysis: Figures 7 and 8 show the visual examples of comparison on the GoPro test set with [2,8,12,17,18] and the RealBlur test set with [7,8,10,12,18]. We

present local details to demonstrate the restoration results of each model. Some previous CNN-based models cannot restore the font counters well and the transformer-based Restomer [18] exhibited good performance. In Table 1, we can see that Restomer [18] has the best PSNR results, and our model outperforms SSIM results in most deblurring models. Compared with the previous models, our model can effectively restore the detail texture, and perform better than Restomer [18] in some scenes, For instance, in the first example in Figure 7 and the last example in Figure 8, our model present sharper results.

Figure 7. Some visual comparisons on the GoPro dataset. From top left to bottom right: blurry images, the ground truth images, the restoration of SRN [12], BANet [17], MPRNet [2], MIMO-Unet+ [8], Restomer [18], and our model.

4.3. Ablation Studies

To verify the effectiveness of each block, we reduced each block separately and trained these structures on the GoPro dataset [7]. The baseline was U-Net with a single input and a single output. It does not include CAFB, SAB, and LGFCB, and only adopts eight residual blocks in each encoder and decoder stage to extract features. We adopted the same loss function as mentioned in Section 3.5, and used the same training strategy. The Table 2 shows the results of our ablation experiment.

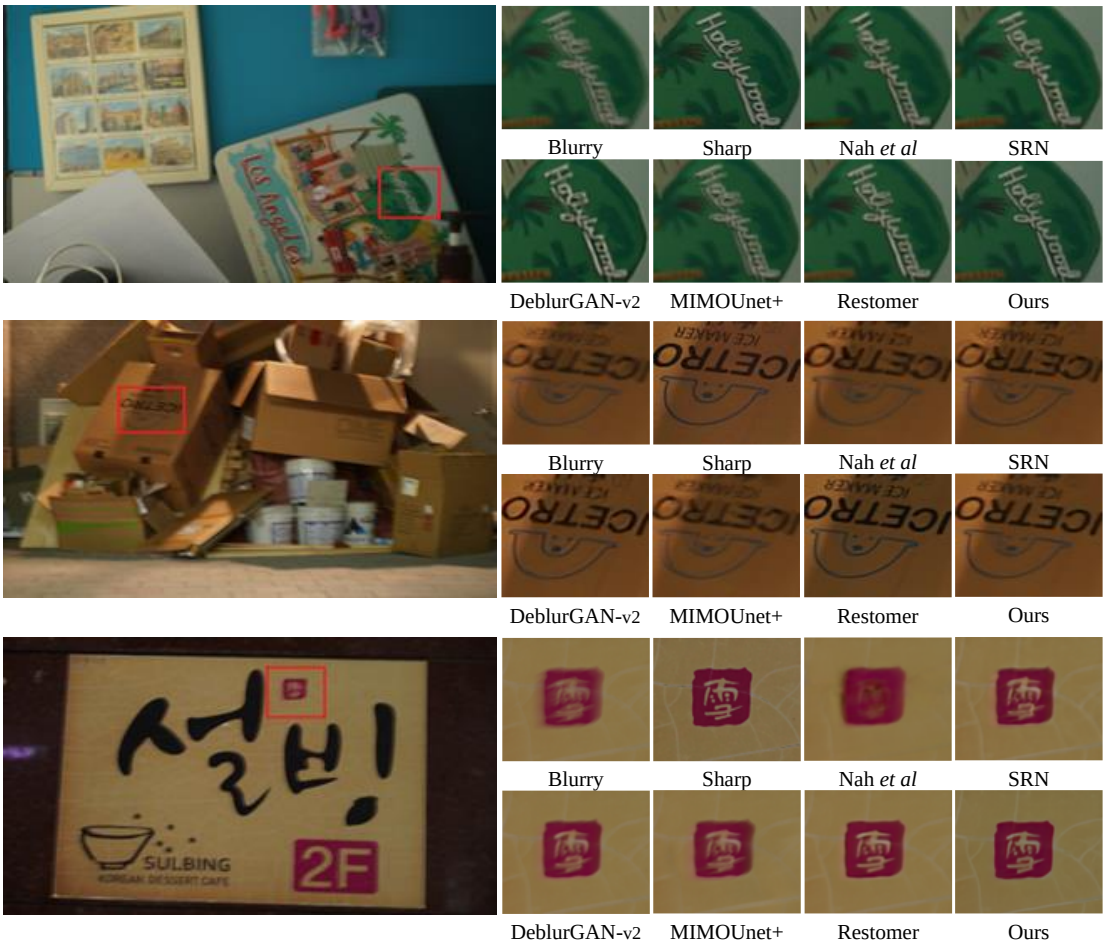


Figure 8. Some visual comparisons on the RealBlur dataset, our models only trained on the GoPro dataset. From top left to bottom right: blurry images, the ground-truth images, the restoration of Nah et al. [7], SRN [12], DeblurGAN-v2 [10], MIMO-Unet+ [8], Restomer [18], and our model.

Table 2. Ablation experiment on GoPro dataset [7]. TB indicates the transformer block. The PSNR and SSIM are average results per image. The '√' means that the block is selected for training.

Baseline	CAFB	SAB	LGFCB		PSNR	SSIM
			MRB	TB		
√					31.44	0.943
√	√		√	√	32.37	0.957
√		√	√	√	32.42	0.959
√	√	√		√	32.11	0.952
√	√	√	√		32.07	0.951
√	√	√	√	√	32.68	0.962

As shown in Table 2, we evaluated the PSNR and SSIM results of each structure. The evaluation results show that each proposed block is effective to some extent. The first and second rows in Table 2 indicate the contributions of SAB and CAFB blocks, which result in 0.31 dB and 0.26 dB improvement, respectively. Notably, we observe that MRB can improve the results by 0.57 dB and the transformer block can increase results by 0.61dB. This demonstrates that both the local feature of the CNN structure and the global feature of the transformer block play an important role in image deblurring. With the use of LGFCB, our model can focus on the global and local feature extraction simultaneously, drastically influencing performance.

4.4. Runtime Comparison

In recent years, deblurring models have made significant progress in accuracy and processing speed. In this paper, we balance efficiency and speed. Namely, we focus on exploring an efficient deblurring model to obtain relatively high accuracy and consume less testing time as well. In general, under the premise of similar accuracy, the model with smaller parameters is more suitable for resource-limited equipment. For a fair comparison, we tested all the compared models in the same environment. The runtime was measured by using the released code of each model on a single TIAN RTX GPU. Table 3 compares our model and some advanced deblurring models in terms of accuracy, model parameters, and process speed.

Table 3. Parameters and runtimes comparison. The runtimes are the average testing time per image in the GoPro test dataset [7]. All models are tested with a single TITAN RTX GPU. The units of parameters and runtime are millions and seconds, respectively.

	DeblurGAN-V2 [10]	**MIMO-Unet+ [8]**	**MPRNet [2]**	**SDWNet [16]**	**Ours**
Params (M)	60.9	16.1	20.1	7.2	15.6
Runtime (s)	0.21	0.017	0.18	0.14	0.012
PSNR (dB)	29.55	32.45	32.66	31.26	32.68
SSIM	0.934	0.957	0.959	0.966	0.962

As we can see, our model achieves competitive PSNR and SSIM performance to other models. It is fast and has relatively small parameters. Notably, SDWNet [16] has the highest SSIM but the processing speed is much slower than our model. Compared with DeblurGAN-V2, our model parameters are only one-third of DeblurGAN-V2's, while our model performs better. This shows that our model has significant advantages in terms of model parameters, process speed, and deblurring performance.

4.5. Object Detection Performance

As we mentioned above, blurry images are detrimental to subsequent vision tasks. In this section, we employed YOLOv4 [33] to analyze the influence of burry images in object-detection tasks and detect the restoration images of some deblurring models [2,8,12,18] to compare the naturalness performance. We input blurry images and restoration images of these deblurring models to YOLOv4 successively and obtained the evaluation outputs. Figure 9 shows the comparison of the detection results. To observe the detection performance of these deblurring models more intuitively, Table 4 lists the detection evaluation results of each object: "X" means this object failed to be detected in the current restoration image.

Table 4. The detection evaluation result of each object. "X" means this object cannot be detected in the restoration of the current deblurring model.

	Blur	**SRN [12]**	**MIMO-Unet+ [8]**	**MPRNet [2]**	**Restomer [18]**	**Ours**
car(left)	0.99	0.99	0.99	0.99	0.99	0.99
plant	X	0.78	0.56	0.74	0.75	0.66
person	0.88	0.76	0.86	0.77	0.86	0.89
chair	X	X	0.65	X	0.72	0.64
umbrella	X	0.54	0.51	0.58	0.59	0.54
car(right)	X	X	X	X	0.60	0.54

As shown in Table 4, the blurry image has unclear contours and only two objects can be detected. The restoration images of deblurring models all have better performance than blurry images. The restoration images of Restomer and our model can detect six objects using YOLOv4. However, the detection results of other restoration images miss some objects. This is probably because some detailed information is lost in the restoration process.

Notably, the car on the right side of the image is inconspicuous; only Restomer [18] and our model can produce sharper images so that YOLOv4 can recognize it accurately. In conclusion, our model is capable of achieving sharp images, and the object-detection performance is satisfactory as well. Moreover, this experiment indicates that image deblurring is suitable for high-level computer vision tasks (e.g., object detection) as a pretreatment technique.

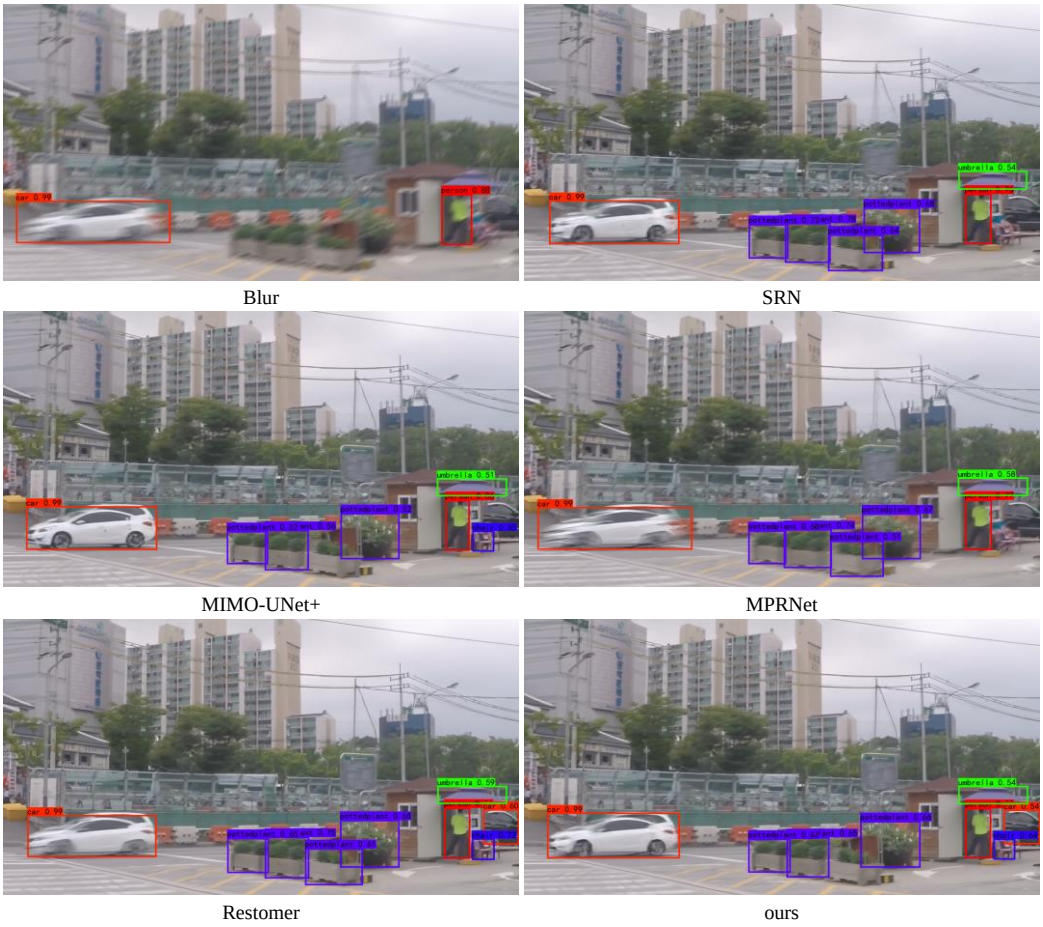


Figure 9. The visual example of object detection. From top left to bottom right: blurry images, the restoration of SRN [12], MPRNet [2], MIMO-Unet+ [8], Restomer [18], and our model.

5. Conclusions

In this paper, we proposed an improved CNN transformer combination network for image deblurring. It can extract richer local features and global features simultaneously, which can ameliorate the details loss and enlarge the receptive field. A series of experiments demonstrate that our model performs well on image deblurring and achieves competitive evaluation results (PSNR and SSIM). When compared with other models, our model does not achieve the best performance in every aspect, but on the whole, it also shows superiority in regard to accuracy and speed. Furthermore, our deblurring model can be considered a pretreatment technique for object detection to improve performance.

Author Contributions: Conceptualization, X.C., Y.W. (Yuanyuan Wan), D.W. and Y.W. (Yuqing Wang); methodology, X.C. and Y.W. (Yuanyuan Wan); software, X.C. and Y.W. (Yuanyuan Wan); validation, X.C. and D.W.; formal analysis, X.C. and Y.W. (Yuanyuan Wan); investigation, X.C. and D.W.; resources, X.C. and D.W.; data curation, X.C. and Y.W. (Yuqing Wang); writing—original draft preparation, X.C., Y.W. (Yuqing Wang), and Y.W. (Yuanyuan Wan); writing—review and editing, X.C., Y.W. (Yuqing Wang), and D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by "Research and application of key technologies for online detection of aerosol Particulate matter based on precision medicine" OF Jilin Province Science and Technology Department, grant number: 20210204134YY.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fergus, R.; Singh, B.; Hertzmann, A.; Roweis, S.T.; Freeman, W.T. Removing camera shake a single photograph. In Proceedings of the ACM SIGGRAPH 2006 Papers, Boston, MA, USA, 30 July–3 August 2006; pp. 787–794.
2. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H.; Shao, L. Multi-stage progressive image restoration. In Proceedings of the CVPR, Online, 19–25 June 2021; pp. 14816–14826.
3. Chakrabarti, A. A neural approach to blind motion deblurring. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 221–235.
4. Hradiš, M.; Kotera, J.; Zemčík, P.; Šroubek, F. Convolutional neural networks for direct text deblurring. In Proceedings of the British Machine Vision Conference, Swansea, UK, 7–10 September 2015; Volume 10, p. 2.
5. Schuler, C.J.; Hirsch, M.; Harmeling, S.; Schölkopf, B. Learning to deblur. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1439–1451. [CrossRef] [PubMed]
6. Sun, J.; Cao, W.; Xu, Z.; Ponce, J. Learning a convolutional neural network for non-uniform motion blur removal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 769–777.
7. Nah, S.; Kim, T.H.; Lee, K.M. Deep multi-scale convolutional neural network for dynamic scene deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3883–3891.
8. Cho, S.-J.; Ji, S.-W.; Hong, J.-P.; Jung, S.-W.; Ko, S.-J. Rethinking coarse-to-fine approach in single image deblurring. In Proceedings of the ICCV, Montreal, QC, Canada, 11–17 October 2021.
9. Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; Matas, J. Deblurgan: Blind motion deblurring using conditional adversarial networks. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 January 2018.
10. Kupyn, O.; Martyniuk, T.; Wu, J.; Wang, Z. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In Proceedings of the ICCV, Seoul, Republic of Korea, 27 October–2 November 2019.
11. Zhang, J.; Pan, J.; Jimmy; Ren, S.J.; Song, Y.; Bao, L.; Rynson; Lau, W.H.; Yang, M.-H. Dynamic scene deblurring using spatially variant recurrent neural networks. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 January 2018.
12. Tao, X.; Gao, H.; Shen, X.; Wang, J.; Jia, J. Scale-recurrent network for deep image deblurring. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 January 2018.
13. Zhang, H.; Dai, Y.; Li, H.; Koniusz, P. Deep stacked hierarchical multi-patch network for image deblurring. In Proceedings of the CVPR, Long Beach, CA, USA, 16–20 January 2019.
14. Park, D.; Kang, D.U.; Kim, J.; Chun, S.Y. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020; pp. 327–343.
15. Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Stenger, B.; Liu, W.; Li, H. Deblurring by realistic blurring. In Proceedings of the CVPR, Online, 14–19 January 2020.
16. Zou, W.; Jiang, M.; Zhang, Y.; Chen, L.; Lu, Z.; Wu, Y. SDWNet: A straight dilated network with wavelet transformation for image deblurring. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1895–1904.
17. Tsai, F.-J.; Peng, Y.-T.; Tsai, C.-C.; Lin, Y.-Y.; Lin, C.-W. BANet: A blur-aware attention network for dynamic scene deblurring. *IEEE Trans. Image Process.* **2022**, *31*, 6789–6799. [CrossRef] [PubMed]
18. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H. Restormer: Efficient transformer for high-resolution image restoration. In Proceedings of the CVPR, New Orleans, USA, 19–24 June 2022; pp. 5718–5729.
19. Kumar, J.; Mastan, I.D.; Raman, S. FMD-cGAN: Fast motion deblurring using conditional generative adversarial networks. In *Communications in Computer and Information Science*; CVIP; Springer: Berlin/Heidelberg, Germany, 2021.
20. Gao, G.; Xu, Z.; Li, J.; Yang, J.; Zeng, T.; Qi, G.-J. CTCNet: A CNN-transformer cooperation network for face image super-resolution. *arXiv* **2022**, arXiv:2204.08696.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
22. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.

23. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; pp. 10347–10357.
24. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
25. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
26. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
28. Huang, P.; Han, S.; Zhao, J.; Liu, D.; Wang, H.; Yu, E.; Kot, A.C. Refinements in motion and appearance for online multi-object tracking. *arXiv* **2020**, arXiv:2003.07177.
29. Charbonnier, P.; Blanc-Feraud, L.; Aubert, G.; Barlaud, M. Two deterministic half-quadratic regularization algorithms for computed imaging. In Proceedings of the ICIP, Austin, TX, USA, 13–16 November 1994; p. 3.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Rim, J.; Lee, H.; Won, J.; Cho, S. Real-world blur dataset for learning and benchmarking deblurring algorithms. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 184–201.
32. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
33. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.