remote sensing

MDPI

Article

KT-Deblur: Kolmogorov–Arnold and Transformer Networks for Remote Sensing Image Deblurring

Baoyu Zhu [1,2,3], Zekun Li [4], Qunbo Lv [1,2,3], Zheng Tan [1,3,*] and Kai Zhang [1,3]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, No.9 Dengzhuang South Road, Haidian District, Beijing 100094, China
[2] School of Optoelectronics, University of Chinese Academy of Sciences, No.19(A) Yuquan Road, Shijingshan District, Beijing 100049, China
[3] Department of Key Laboratory of Computational Optical Imagine Technology, CAS, No.9 Dengzhuang South Road, Haidian District, Beijing 100094, China
[4] National Computer Network Emergency Response Technical Team/Coordination Center of China, Yumin Road, Chaoyang District, Beijing 100020, China
* Correspondence: tanzheng@aircas.ac.cn; Tel.: +86-1591-063-9123

Abstract: Aiming to address the fundamental limitation of fixed activation functions that constrain network expressiveness in existing deep deblurring models, in this pioneering study, we introduced Kolmogorov–Arnold Networks (KANs) into the field of full-color/RGB image deblurring, proposing the Kolmogorov–Arnold and Transformer Network (KT-Deblur) framework based on dynamically learnable activation functions. This framework overcomes the constraints of traditional networks' fixed nonlinear transformations by employing adaptive activation regulation for different blur types through KANs' differentiable basis functions. Integrated with a U-Net architecture within a generative adversarial network framework, it significantly enhances detail restoration capabilities in complex scenarios. The innovatively designed Unified Attention Feature Extraction (UAFE) module combines neighborhood self-attention with linear self-attention mechanisms, achieving synergistic optimization of noise suppression and detail enhancement through adaptive feature space weighting. Supported by the Fast Spatial Feature Module (FSFM), it effectively improves the model's ability to handle complex blur patterns. Our experimental results demonstrate that the proposed method outperforms existing algorithms in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics across multiple standard datasets, achieving an average PSNR of 41.25 dB on the RealBlur-R dataset, surpassing the latest state-of-the-art (SOTA) algorithms. This model exhibits strong robustness, providing a new paradigm for image-deblurring network design.

Keywords: Kolmogorov–Arnold network (KAN); remote sensing; image deblurring; transformer; generative adversarial networks (GANs); deep learning

check for updates

Academic Editors: Silvia Liberata Ullo and Li Zhang

Received: 21 January 2025
Revised: 24 February 2025
Accepted: 25 February 2025
Published: 27 February 2025

Citation: Zhu, B.; Li, Z.; Lv, Q.; Tan, Z.; Zhang, K. KT-Deblur: Kolmogorov–Arnold and Transformer Networks for Remote Sensing Image Deblurring. *Remote Sens.* **2025**, *17*, 834. https://doi.org/10.3390/rs17050834

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Satellite remote sensing technologies have been widely applied in various fields, such as resource management, traffic monitoring, geological disaster early warning, and construction relating to national defense [1–4]. With the proliferation of smartphones and digital cameras, images have also become a primary medium for obtaining information in daily life [5]. However, motion blur can occur when the capturing platform used is subject to movement or random vibrations, and the rapid motion of objects further exacerbates this issue [6,7]. Moreover, in real-world scenarios, factors such as spatial variations in the blur kernel, imaging depth, and sensor noise complicate the blurring process, significantly

degrading image quality [8]. This not only affects visual perception but also introduces challenges for subsequent image analysis and task processing, especially in applications like object detection and understanding scenes. As a result, recovering clear, high-quality images from blurred ones has become a challenging task in the field of computer vision.

Motion blur in images typically arises when, during the exposure time, the displacement of the detector relative to the target exceeds the size of a pixel. In real-world scenarios, the blur kernel is often an unknown quantity and can typically be modeled as follows [9]:

$$B = S * k(M) + N, \tag{1}$$

where B is the blurred image; S is the corresponding clear image; $k(M)$ denotes the unknown blur kernel; M denotes the sparse matrix, each row of which contains a local blur kernel; $*$ is the discrete convolution; and N is noise.

Traditional image-deblurring methods, such as Wiener filtering and blind deconvolution, have made notable progress in the field [10–12]. These approaches rely on estimating the blur kernel to restore the sharpness of an image. However, their performance is limited when dealing with complex blur, especially in the presence of noise interference and intricate backgrounds. These methods suffer from the following main limitations. The blurring kernel in a realistic scene is often unknown and changes with the scene, and traditional methods cannot deal with complex blurring situations. Additionally, some deblurring algorithms need to estimate and optimize the blurring kernel repeatedly, leading to higher computational complexity.

In recent years, deep learning, particularly with respect to Convolutional Neural Networks (CNNs), has provided new solutions for image deblurring. Unlike traditional methods, which rely on explicit estimation of the blur kernel, deep-learning approaches leverage large-scale data to automatically learn how to restore the sharpness of blurred images. This ability has led to significant advancements in the field of image deblurring.

Among deep-learning-based deblurring models, the U-Net architecture has become one of the most widely used due to its outstanding image restoration capabilities. U-Net employs an encoder–decoder structure, with skip connections and multi-scale feature fusion, ensuring that low-level image features are effectively transmitted to higher layers. This design enables the model to retain fine details and improve overall image quality [13]. Generative Adversarial Networks (GANs) have a significant advantage in the image-deblurring task. GANs consist of a generator and a discriminator, which mutually optimize themselves by training against each other, enhancing the realism of the final image generated [14].

Although U-Net has achieved significant success in image deblurring, several challenges remain. First, U-Net relies on manual convolutional kernels and feature extraction methods, and it may be difficult to handle the substantial distribution gap between real-world image data and synthetically blurred data using these approaches [15]. This disparity limits the network's ability to adapt to various complex blur types in real-world scenarios. Additionally, traditional U-Net models typically overlook the dynamic changes in the blur kernel, which are crucial for restoring image quality. While some researchers have attempted to enhance U-Net's performance by introducing various attention mechanisms, the complexity and variability of the blur kernel, along with insufficient noise suppression, continue to pose significant challenges. Moreover, fixed activation functions, such as ReLU and Sigmoid, may fail to effectively capture the intricate features of input data when dealing with complex blur situations.

KANs [16,17], based on the Kolmogorov–Arnold machine learning theory, offer a promising solution to these issues. The key innovation of KANs lies in replacing traditional fixed activation functions with learnable ones. By introducing a dynamic, learnable activa-

tion function, KANs can adaptively adjust the nonlinear transformation during the training process, thereby improving their performance in complex tasks.

In recent years, transformer architectures based on self-attention mechanisms have been widely applied in image-deblurring tasks [18]. The self-attention mechanism is highly effective in modeling global dependencies within an image, leading to significant improvements in deblurring performance. When combined with CNNs, attention mechanisms complement the limitations of traditional CNNs, especially in handling complex backgrounds and fine-detail recovery [18–20]. However, the existing methods tend to perform poorly when dealing with novel blur types that were not present in the training datasets [15], and they may also be susceptible to overfitting.

Additionally, blur and noise in images often exhibit uneven distributions. Certain regions may contain strong texture information, while others might be impacted by noise or blur artifacts. Consequently, recovering details and textures in image-deblurring tasks often requires the precise capture of local features. Although transformers excel at modeling global dependencies, they may struggle to recover fine detail, particularly when dealing with small or subtle regions. Furthermore, the computational complexity of transformer architectures is relatively high, especially within the self-attention mechanism, where the time complexity is $O(N^2)$. This facet becomes a bottleneck in terms of efficiency when processing high-resolution images.

To address the limitations of existing image-deblurring methods, we introduce a novel image-deblurring framework by integrating KANs with transformer architectures: the KT-Deblur. KT-Deblur has a generative adversarial network architecture and integrates a U-Net structure to build generators. In the generator network, a neighborhood self-attention mechanism is employed, along with the proposed Minimal and Fast Feature Extraction Module (MFFEM) and the KAN module, forming a UAFE in the generator. KT-Deblur dynamically adjusts its focus across different regions within the local context, making it more adaptable to various blur types. This method has significant advantages, particularly in handling dynamic or non-uniformly blurred images, and improves detail recovery accuracy through the integration of the U-Net encoder–decoder structure. The specific contributions of this work are listed below.

1. This is the first application of KANs in image deblurring: Our team is the first to apply a KAN to image deblurring, exploring a novel method of integrating KANs with the U-Net architecture. By combining KANs with U-Net, we have created a new KT-Deblur model, which dynamically adjusts the activation function based on the features of each input image, enabling the model to more flexibly handle different types of blur and effectively recover image details.
2. Union Attention Feature Extraction Module (UAFE): In this paper, we propose the UAFE, which combines the neighborhood self-attention mechanism, linear attention mechanism, MFFEM, and KAN module. By adaptively learning to recognize local feature relationships in an image, this model can effectively differentiate between noise and important details, dynamically adjusting its focus based on the content of the image. This enhances deblurring performance, especially when dealing with multiple blur types and noise interference.
3. Experimental validation and comparative analysis: Through extensive experiments on several benchmark datasets, we have determined that the proposed method outperforms existing methods in terms of PSNR, SSIM, and other metrics. Particularly in handling complex blur scenarios, the model demonstrates strong robustness and effectively manages various blur types. The experimental results confirm that the proposed approach offers an efficient and effective solution for practical image restoration tasks.

2. Related Work

2.1. Image-Deblurring Methods

Traditional image-deblurring methods, including blind deconvolution, Wiener filtering, and non-blind deconvolution, primarily rely on the accurate estimation of the blur kernel. However, the performance of most of these techniques significantly drops in noisy environments. With the rise of deep learning, numerous CNN-based deblurring approaches have been proposed [14]. Among these is the U-Net architecture, which has achieved remarkable success in image deblurring by leveraging a multi-layer encoder–decoder structure and skip connections. In recent years, deblurring methods incorporating generative models, such as GANs, have also shown promising results, further enhancing the effectiveness of image restoration [21,22].

The fundamental principle of end-to-end deep-learning-based image deblurring methods is to directly learn the mapping from the blurred domain to the sharp domain. Specifically, given a blurred image I_{blur}, an end-to-end deblurring model can use a deep neural network to directly output the corresponding sharp image I_{sharp}. This process can be expressed as follows:

$$I_{sharp} = F(I_{blur}; \vartheta). \tag{2}$$

In this equation, I_{blur} represents the blurred image, while I_{sharp} is the sharp image restored by the deep-learning neural network model F, with ϑ denoting the network parameters. During training, an appropriate loss function is defined to measure the difference between the restored image and the sharp image. Using a corresponding paired dataset and the loss function, the neural network is trained through backpropagation, continuously adjusting and optimizing the network parameters to minimize the loss function. This enables the network to accurately restore a blurred image to its sharp counterpart. During inferencing, the blurred image to be processed is fed into the trained deep neural network, and the output is computed through a forward propagation algorithm.

DeepDeblur [23] employs an end-to-end approach for restoring blurred images, constructing a processing pipeline from coarse to fine through the stacking of multiple subnetworks, resulting in impressive deblurring performance. DeblurGAN [24], based on GANs, introduces a novel image-deblurring algorithm that leverages gradient penalties and perceptual loss to effectively restore blurred images. DeblurGAN-V2 [22] builds upon DeblurGAN, optimizing it by incorporating a lightweight backbone network and a feature pyramid network, thus achieving a balance between performance and efficiency. MPRNet [25] has a multi-stage network architecture that progressively learns image features. MAXIM [26] has a multi-axis Multilayer Perceptron (MLP)-based architecture, enhancing the model's learning capacity.

Inspired by the success of Vision Transformer (ViT) [27] in various computer vision tasks, several transformer-based models [18,28,29] have been developed, exhibiting promising performance. Uformer [18] employs a local-enhanced window transformer block and a multi-scale recovery modulator, reducing computational complexity while retaining a powerful capacity to capture both local and global dependencies for image restoration. Stripformer [19] combines a transformer architecture with intra-strip and inter-strip attention mechanisms, allowing the network to re-weight image features in both horizontal and vertical directions. Restormer [29] is an efficient transformer model capable of capturing long-range pixel interactions, making it suitable for large images.

2.2. Attention Model

The attention mechanism was initially introduced in the field of natural-language processing (NLP) to enable neural networks to focus on key parts of text sequences, thereby enhancing the model's processing capabilities. This mechanism is increasingly applied to

computer vision tasks, enabling neural networks to more accurately recognize and process images and videos, particularly in image-deblurring tasks. By using the attention mechanism, the network can dynamically focus on critical areas within an image, significantly improving the accuracy of image restoration.

In traditional convolution operations, the network can only process information within a limited receptive field, leading to the loss of global context. In contrast, attention mechanisms can effectively capture long-range dependencies by learning task-specific attention distributions. Attention can be viewed as a dynamic selection process that adaptively weights features to highlight the most important parts of the input. SENet [30] recalibrates channel feature importance by explicitly modeling the relationships between feature maps. GSoP-Net [31] enhances the ability to gather global information by employing global second-order pooling (GSoP).

In the field of aerospace remote sensing, attention mechanisms also play crucial roles in various ways [32–35]. LMNet [36] introduces the Residual Transformer 3D-Spatial Attention Module (RT3DsAM), which can learn feature representations globally and filter out important information. Uformer [18] replaces the convolutional layers in the U-Net model with transformer-based locally enhanced window transformer blocks, utilizing the self-attention ability of feature maps at multiple scales to recover more image details. The authors of [37] explored the characteristics of transformers in the frequency domain and developed an efficient frequency-domain-based self-attention solver, which estimates scaled dot-product attention using element-wise multiplication instead of matrix multiplication in the spatial domain. Neighborhood Attention (NA) [38] is an efficient and scalable sliding window attention mechanism for vision tasks. NA operates on a per-pixel basis, significantly reducing computational complexity by limiting self-attention to local neighboring pixels. With linear time and space complexity, NA differs from the windowed self-attention in the Swin Transformer, as its sliding window pattern allows the receptive field to gradually expand while maintaining translational invariance, without the need for additional pixel shifting. This feature makes NA both efficient and effective in maintaining good performance when processing images.

2.3. Kolmogorov–Arnold Network

As fundamental building blocks of deep learning models, MLPs demonstrate strong theoretical expressive power in nonlinear function approximation tasks, supported by the universal approximation theorem. However, traditional MLPs exhibit significant limitations in practical applications: In mainstream architectures such as transformers, MLPs typically account for the majority of non-embedding parameters. This not only reduces model transparency but also severely constrains model interpretability due to MLPs' black-box nature.

To overcome these limitations, researchers have begun exploring alternative architectures based on the Kolmogorov–Arnold representation theorem. This theorem states that any continuous multivariate function can be decomposed into a finite superposition of univariate functions, providing a theoretical foundation for network construction. However, existing works [39] are largely confined to the original theorem's two-dimensional network configuration (a network depth of 2 and a hidden layer neuron count of (2n + 1)), failing to incorporate modern training techniques such as backpropagation for optimization and extension. In this context, ExSpliNet [40] integrates concepts from Kolmogorov networks, probabilistic tree ensembles, and multivariate B-spline representations. It provides a probabilistic interpretation of a model and demonstrates its universal approximation properties. Furthermore, the associated study discusses how to leverage B-spline properties for efficient model encoding. In [41], the authors explain how the Kolmogorov Superposition

Theorem (KST) can address the curse of dimensionality when approximating multivariate continuous functions in high-dimensional spaces. In one study on the KST [42], the authors investigate the optimal expressive capability of linear modules, such as ReLU, in deep neural networks (DNNs), and demonstrate how this approach enhances the approximation power of ReLU DNNs, especially in the context of high-dimensional continuous functions.

Specifically, this model uses a function-based approximation that allows the activation function of each neuron to be obtained by learning about the following relationship:

$$z = f_\theta(x), \tag{3}$$

where $f_\theta(x)$ represents a learnable activation function, with θ being the set of parameters optimized during the training process. This activation function, $f_\theta(x)$, can be dynamically adjusted based on the complexity of the input data, thus allowing the network to automatically adapt the activation function according to the characteristics of the data, thereby enhancing its expressiveness.

In a study on KANs [16], the original Kolmogorov–Arnold representation was extended to arbitrary depths and widths and integrated with existing deep learning models. Through extensive empirical experiments, the authors of this study demonstrated the potential of KANs as foundational models for both AI and scientific applications, validating their accuracy and interpretability. The results of their study revealed that while MLPs can learn generalized additive structures, using ReLU activation functions to approximate exponential and sinusoidal functions is inefficient. In contrast, KANs efficiently learn composite structures and univariate functions, resulting in significantly superior performance compared to MLPs.

Inspired by the Kolmogorov–Arnold Theorem (KAT), the KAN treats multivariate functions as collections of learnable univariate spline functions defined at the edges [16]. A single KAN layer can thus be represented by a matrix of univariate functions, where each function is designed to capture the individual components of the multivariate input space.

$$\Phi = \{\phi_{q,p}\}, \tag{4}$$

where $p = 1, 2, \ldots, n_{in}$, $q = 1, 2, \ldots, n_{out}$. n_{in} is the input dimension, n_{out} is the output dimension, and $\phi_{q,p}$ is the learnable function. Using n_i to denote the neuron node in layer i and $x_{l,i}$ to denote the value of the ith neuron in layer l, ϕ is a function between the two layers, and the activation of each node in the $l+1$ layer is calculated as follows:

$$x_{l+1,\,j} = \sum_{i=1}^{n_i} \phi_{l,\,j,\,\,i}\,(x_{l,i}), \tag{5}$$

The implementation of Layer KAN [17] in the form of layer-to-layer connections can be described as follows:

$$KAN(X) = (\phi_{L-1} \circ \phi_{L-2} \circ \cdots \circ \phi_0)(x). \tag{6}$$

3. Method

As shown in Figure 1, the proposed KT-Deblur is based on a GAN framework and consists of two main components: (1) a generator (KT-Deblur G) and (2) a discriminator (KT-Deblur D). The generator serves as the core of the network, responsible for processing the blurred image and restoring it to a sharp image. The discriminator, on the other hand, evaluates the quality of the generated image, determining whether it is sufficiently realistic.

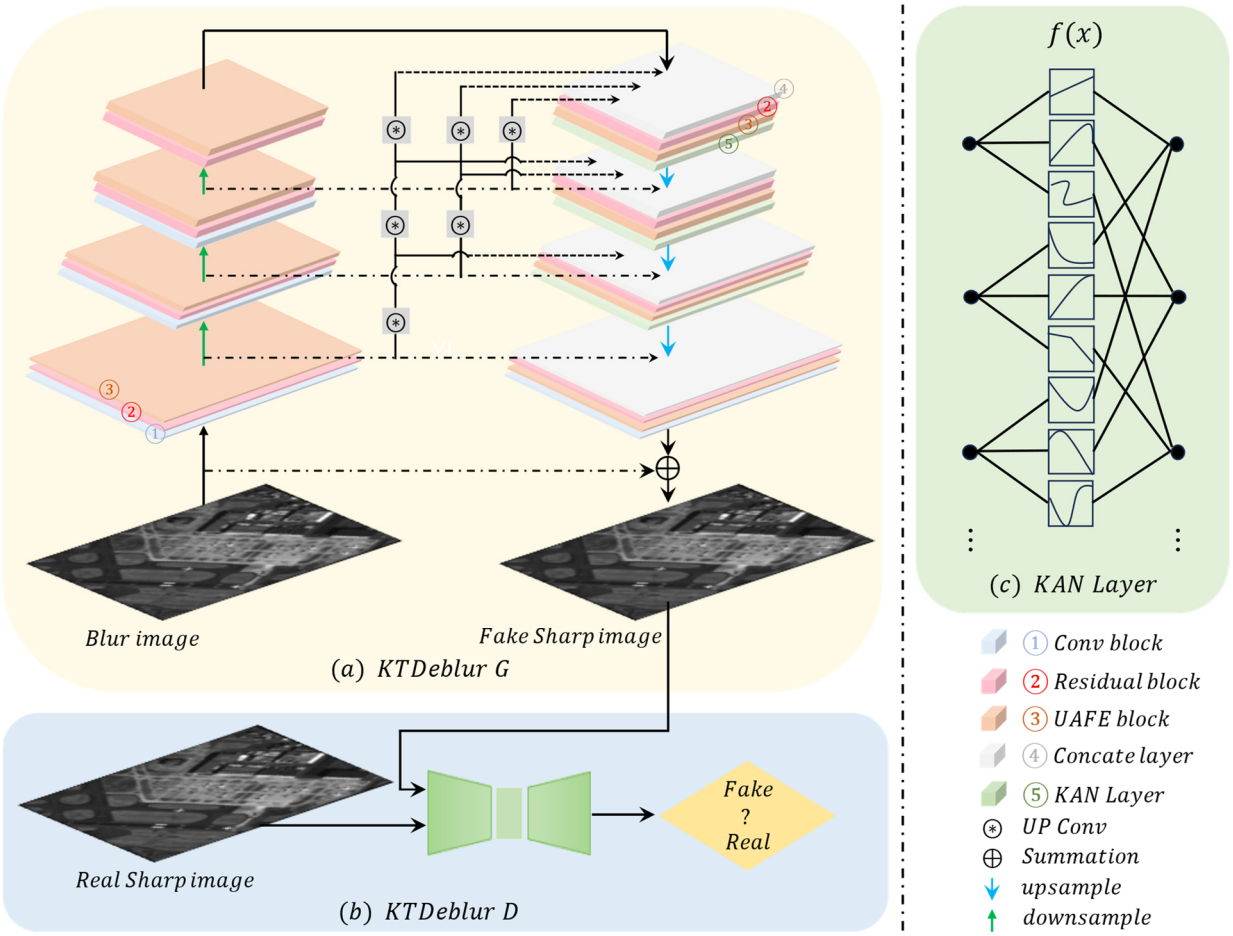

Figure 1. Schematic diagram of the network architecture of KT-Deblur. (**a**) the generator (KT-Deblur G); (**b**) the discriminator (KT-Deblur D); (**c**) the KAN layer.

The KT-Deblur framework proposed in this paper draws on the classic U-Net architecture and integrates the Union Attention Feature Extraction Module with a KAN to construct the generator network, thereby enhancing the model's ability to restore low-quality images.

U-Net is a well-established convolutional neural network architecture, utilizing an encoder–decoder structure with skip connections to preserve low-level feature information, making it highly effective at restoring image details, particularly in image-deblurring tasks. In this study, the KAN module was combined with U-Net to construct the generator. By adaptively learning activation functions and leveraging both global and local features, the model improves image recovery performance. The strong feature extraction and reconstruction capabilities of the U-Net structure allow for precise restoration of image details and structures, significantly enhancing the effectiveness of deblurring.

Unlike traditional MLP models, the proposed network replaces fixed activation functions with learnable ones, eliminating the dependence on linear weight matrices and fundamentally improving adaptability to various types of blurred images. Coupled with the powerful feature extraction and reconstruction abilities of U-Net, this network further optimizes deblurring results. Additionally, the network introduces a neighborhood attention module that computes localized attention to capture spatial dependencies within an image or sequence, reducing computational complexity while enhancing sensitivity to local structural information. The module adaptively computes the weights within the neighborhood and performs weighted fusion of features, enabling each position to effectively utilize its surrounding context, thereby improving feature representation and overall performance.

The key components of the proposed method are discussed in detail below.

3.1. Kolmogorov–Arnold Network

The KAN is the key module in the method described in this paper. It is mainly used to allow the adaptive learning of the activation function.

In existing deep learning neural networks, the activation functions, such as ReLU, Sigmoid, or Tanh, are usually fixed, assuming that the input of a certain neuron is x and its linearly transformed output is

$$y = W * x + b, \tag{7}$$

where W is the weight, b is the bias, and x is the input vector. Deep-learning neural networks typically map the output y nonlinearly through a fixed activation function $\sigma(y)$; for example,

$$\sigma(y) = RELU(x). \tag{8}$$

The use of fixed activation functions limits a network's ability to adapt its nonlinear transformations to different types of blur, thereby restricting its expressive power. To enhance the network's flexibility, we introduced learnable activation functions. By adaptively learning the shape and parameters of the activation functions, the network can optimize the nonlinear mappings at each layer based on the characteristics of the input data. This allows the network to better capture the complex relationships in the data, improving its overall performance in handling various types of blur.

According to the research conducted by Vladimir Arnold and Andrey Kolmogorov, based on the concept of multivariate continuous functions, if f is a multivariate continuous function defined on a bounded domain, it can be represented as a finite combination of univariate continuous functions and binary addition operations. This decomposition enables the expression of complex multivariate functions in terms of simpler univariate components, facilitating more efficient analysis and approximation. In particular, for smooth functions,

$$f(x) = f(x_1, x_1, \cdots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_q^n \phi_{q,p}(x_p) \right), \tag{9}$$

where $\phi_{q,p} : [0,1] \to \mathbb{R}, \Phi_q : \mathbb{R} \to \mathbb{R}$ denote continuous functions, each of which can be represented by a univariate function and a summation.

In an adaptive activation function network, the activation function $\sigma(x)$ is no longer a fixed function but is replaced by a learnable function $f_\theta(x)$, where θ is a network parameter indicating the shape of the activation function. That is,

$$\sigma(x) = f_\theta(x), \tag{10}$$

where $f_\theta(x)$ is an activation function that can be optimally learned through training and dynamically adjusted according to the complexity of the input data, and the parameter θ is dynamically adjusted by the optimization algorithm. The activation function $f_\theta(x)$ can be dynamically adjusted according to the complexity of the input data, through which the network is able to automatically adjust the activation function according to the characteristics of the data, thus enhancing the expressive ability. $f_\theta(x)$ can be expressed as a weighted sum of a set of basis functions:

$$f_\theta(x) = \sum_{i=0}^{N} \alpha_i g_i(z, \theta_i), \tag{11}$$

where $\{g_i(z, \theta_i)\}$ is a set of basis functions, and α_i and θ_i are parameters obtained through training.

Figure 2 illustrates the architecture of an N-layer KAN. In contrast to traditional neural networks, which rely on linear weight matrices, the KAN layer employs learnable

univariate functions to parameterize the weights. Each KAN node computes a weighted sum of the input signals without introducing any fixed nonlinear transformations. This design eliminates the dependency on conventional linear weight matrices and enables the network to adapt more flexibly to different types of image blurring by dynamically adjusting the shape and characteristics of the activation function. As a result, the KAN effectively overcomes the limitations of traditional activation functions when dealing with complex blur scenarios. Through this adaptive training mechanism, the KT-Deblur model, integrated with the KAN layer, can dynamically modify the activation function's shape according to the specific features of each input image, leading to enhanced image detail recovery and improved deblurring performance.

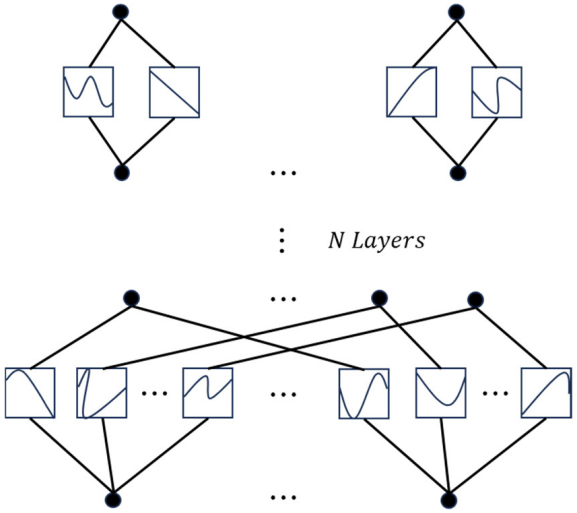

Figure 2. Schematic diagram of an N-layer KAN.

Compared to traditional deblurring methods, the KT-Deblur model incorporating the KAN layer offers greater flexibility in addressing various types of blur. This is particularly evident when handling complex dynamic types of blur, such as motion blur and defocus blur, wherein KT-Deblur significantly enhances the restoration of image details and overall clarity. The advantages of KT-Deblur become even more pronounced in complex scenarios, such as those involving space-varying dynamic blur and defocus blur, demonstrating its superior performance in these challenging contexts.

One of the common challenges in image-deblurring tasks is the issue of overfitting, especially when noise and unnecessary details are excessively fitted, leading to a decline in the quality of the restored image. KT-Deblur addresses this challenge by incorporating learnable activation functions and a neighborhood self-attention mechanism, enabling the network to focus more on the core features of an image while reducing the risk of overfitting. When trained on diverse datasets, the model can adaptively learn the most suitable restoration strategy, enhancing its generalization ability and preventing overfitting. This capability enables KT-Deblur to demonstrate greater robustness in practical applications, effectively handling a wide range of unknown and complex blur types and offering broader applicability and flexibility.

3.2. Union Attention Feature Extraction Module

The UAFE consists of three key components: the Global Importance Weighted Neighborhood Self-Attention Transformer Module (GNATM), the MFFEM, and the KAN layer.

The Neighborhood Attention Transformer (NAT) is a novel hierarchical Transformer based on NA, combining the global-context-modeling ability of self-attention with the advantages of local feature extraction. This approach allows the model to assign varying

weights to different regions of the input data adaptively, enabling the receptive field to focus on the neighborhood around each token. It introduces local inductive bias without requiring additional operations.

Building on this, to further enhance image-deblurring performance, we introduce the GNATM in the model's generator. By focusing on the local neighborhood information of each pixel, this module strengthens the network's ability to recover fine details and model complex visual information. Particularly at the edges and in the high-frequency regions of an image, it effectively addresses the issue of detail loss in image recovery while enhancing structural information. Unlike traditional self-attention mechanisms, this module integrates global information while controlling attention scope or adding neighborhood constraints, thereby avoiding excessive computational complexity.

The GNATM was combined with the MFFEM and the KAN module to construct the union attention feature extraction module of the generator. A flowchart depicting the flow of data is shown in Figure 3.

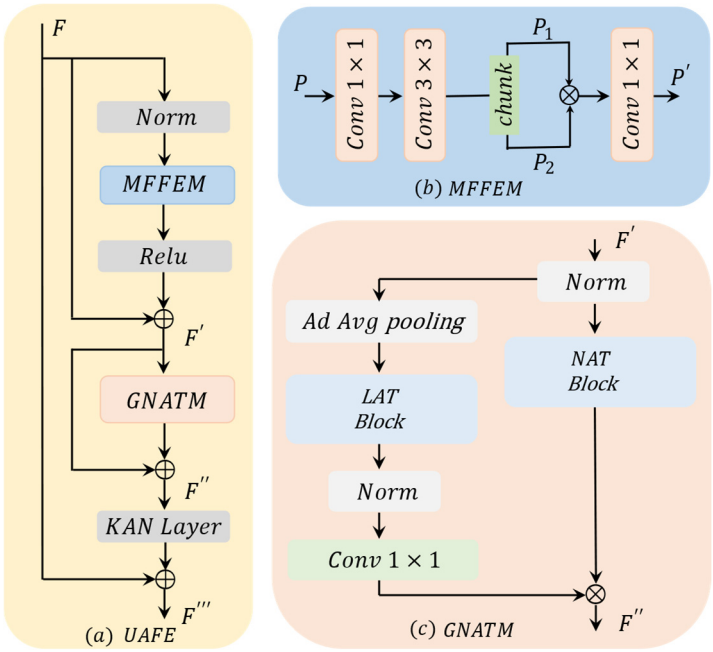

Figure 3. (**a**) Flowchart depicting the flow of data for the Union Attention Feature Extraction Module (UAFE), (**b**) is the schematic diagram of MFFEM, (**c**) is the schematic diagram of GNATM.

As shown in Figure 3a, the flow data for the features processed by the Union Attention Feature Extraction Module is as follows: The feature map F is normalized and fed into the MFFEM, which extracts the preliminary features through a series of convolutional operations and combines them with the ReLU activation function to achieve a nonlinear transformation. After residual concatenation, an enhanced feature representation F' is obtained; it contains key structural information from the image. Subsequently, the feature F' is passed into the GNATM, which evaluates the importance of local regions through a dynamic weighting mechanism to extract features with significant semantic meaning. Through residual concatenation, the feature F'', containing rich details and deep structural information, is obtained, based on which further nonlinear transformations are carried out via variable activation functions, and finally a high-quality feature map is generated.

As shown in Figure 3b, the MFFEM consists of three convolutional layers and a dimensional segmentation module. Specifically, the initial feature map P has a channel number of C. Firstly, the channel number is extended to 2C by two convolutional layers, and then the feature map is uniformly segmented into an equal number of tensor blocks P1

and P2 in the channel dimension. Finally, the final output feature map P' is obtained by the convolutional layers after dot-multiplication operations.

As shown in Figure 3c, the GNATM significantly enhances the network's sensitivity to detail and structural information in extracting image features by focusing on the local neighborhood information of each pixel point. Specifically, the feature map F' obtained by the MFFEM is fed in parallel to the Global Feature Importance Weight Generation Module and the Neighborhood Attention Module. Through these two modules, the global importance weights and the importance information of local neighborhood features, respectively, are extracted, and finally the two are fused via multiplication operations to obtain the enhanced feature map F''.

The process of global importance weight generation includes the following steps: first, the feature map F' is normalized and then subjected to an adaptive average pooling operation to obtain the global information of the image; then, the features are weighted by the linear attention module and further normalized; finally, the global importance weight is obtained through a 1×1 convolution operation. This weight is multiplied pixel-by-pixel by the local feature map output from the neighborhood attention module, finally yielding the fused feature map F'', thus effectively combining the global and local information and improving the detail expression and structure perception of the feature map.

The Adaptive Average Pooling operation (Adapool) [43] is an adaptive and exponentially weighted pooling method. It dynamically adjusts the size of the pooling window, allowing the model to process input data of varying sizes while maintaining a fixed output feature map size. This approach enhances the model's flexibility and generalization capability.

Adapool effectively preserves important image features while ensuring consistent output dimensions, mitigating the information loss typical of traditional pooling methods. On the other hand, exponential weighted pooling assigns higher weights to neighboring regions, emphasizing local feature details and increasing sensitivity to local variations. This technique smoothly responds to changes in input data, avoiding excessive smoothing or loss of details, making it particularly well suited to handling images with non-uniform distributions or complex backgrounds. By combining these two pooling strategies, this method simultaneously ensures image size consistency and retains local feature details, thereby improving the performance of deep-learning models in image deblurring and other image processing tasks.

The GNATM uses a linear attention module for feature weighting, which involves the use of the neighborhood attention mechanism to enhance detail-rich regions, smooth out unimportant regions, and preserve more structural information of an image during the recovery process. The data flow processes for the two modules are shown below.

The data flow of the Linear Attention Module (LAT Block) is illustrated in Figure 4a. Initially, the input feature map X is normalized to ensure numerical consistency and stability. A linear transformation is then applied to map the feature map into a new feature space, transforming it into a form suitable for attention calculation. To reduce computational complexity, low-rank approximation is employed, using a projection matrix $P \in \mathfrak{R}^{L \times K}$ (where $K \ll L$) to compress the attention matrix into smaller dimensions K. This reduces the traditional $O(L^2)$ computational complexity to $O(L \times k)$, overcoming the quadratic time complexity inherent in traditional attention mechanisms. Specifically, the projection matrix P is used to approximate the queries Q and keys K in low-rank space, yielding low-dimensional representations Q' and K'. By calculating attention weights in this lower-dimensional space, linear attention reduces computational complexity, enhances feature selection efficiency, and emphasizes important information via weighted aggregation, enabling the efficient capture of global dependencies in sequences while simultaneously

reducing memory and computational overhead. This mechanism is particularly effective for handling long sequence data, significantly improving computational efficiency while preserving the ability to model global information.

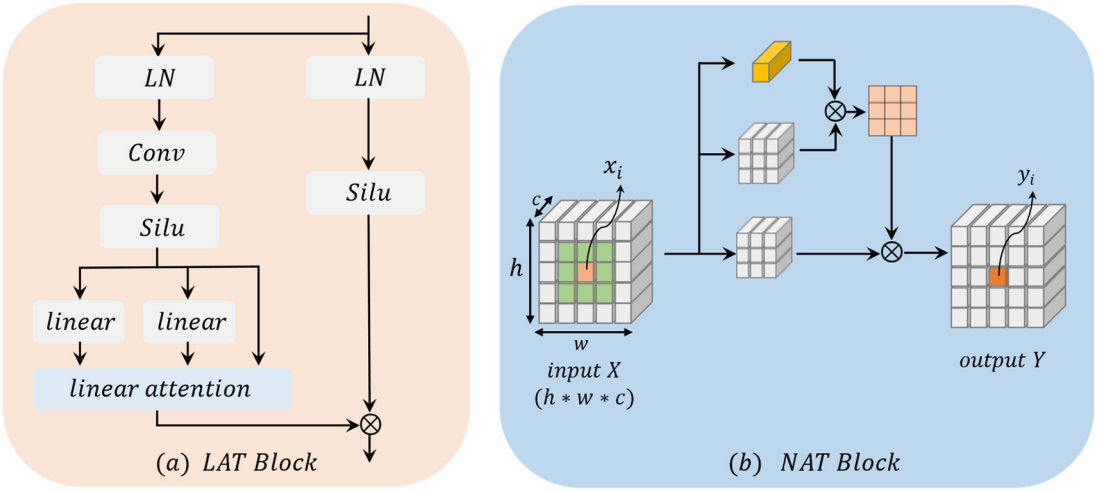

Figure 4. (**a**) Schematic of the data flows for the linear attention mechanism mechanism, (**b**) Schematic of the data flows for the neighborhood attention mechanism.

The Neighborhood Attention Block (NAT Block) dynamically adjusts the recovery weight of each pixel point by calculating the correlation between each pixel point and its neighborhood. Specifically, the network assigns a neighborhood attention weight to each pixel point; this weight reflects the importance of that pixel for detail recovery in the deblurring process. With neighborhood feature weighting, the network can enhance detail-rich regions while smoothing unimportant regions when recovering an image, thus preserving more of the image's structural information during the recovery process.

The Neighborhood Attention Module (NAM) is designed to capture local spatial or contextual dependencies, enhancing sensitivity to local information while reducing computational complexity through a localized attention mechanism. The data flow process of the NAM can be broken down into several key steps.

Initially, the input feature map $X \in \Re^{C \times H \times W}$ is fed into the module, where H and W represent the height and width of the image, and C denotes the feature dimensions. The feature map is subjected to preliminary normalization to ensure numerical stability and improve training robustness, addressing any input data bias and mitigating potential issues such as gradient vanishing or explosion. To capture contextual information from the local space, the Neighborhood Attention Module defines a fixed-size local window or neighborhood $N_{i,j}$ around each position. During computation, each position i interacts with other positions $j \in N_{i,j}$ within its neighborhood, modeling the local features surrounding it. The similarity between each position and other positions in its neighborhood is computed and weighted based on this similarity in order to aggregate important local features.

Specifically, the calculation of similarity within the neighborhood is achieved using queries (Q), keys (K), and values (V). First, the input feature map is mapped to the query ($Q \in \Re^{H \times W \times d}$), key ($K \in \Re^{H \times W \times d}$), and value ($V \in \Re^{H \times W \times d}$) spaces through linear transformations. A dot product operation is then applied to generate a similarity matrix $G \in \Re^{L \times L}$, which reflects the relative importance of each position in the input feature map and its neighboring positions. After computing the similarity matrix, a normalization process is applied to standardize it, producing a set of weights for each position relative to others in its neighborhood. This ensures that important local features are emphasized

while irrelevant features are suppressed, thereby enhancing the model's sensitivity to local structural information.

Feature map Y, after being weighted and normalized through neighborhood attention, is fused with the original input feature map via a residual connection. This fusion enables the network to retain local information while fully leveraging the enhanced features that have been weighted and normalized, leading to a better expression of the key features of the input data.

3.3. Loss Function

In order to optimize network performance, a composite loss function combining content loss and perceptual loss was designed. With this loss function, the network can better balance the need to recover details and reduce noise. The Adam optimizer was used for network training, and a suitable learning rate and batch size were chosen. In terms of data preprocessing, the training data were normalized to improve training stability and convergence speed.

During the training of the deblurring model, in order to compare the trained image, the reconstructed completed image, and the original paired clear image with respect to certain metrics, similar to AMD-GAN [44], we used a three-component loss function including adversarial loss, perceptual loss, and reconstruction loss: L, $w_g > 0.76$, $w_p \in (0.1, 0.2)$, and $w_x \in (0.05, 0.2)$. L is defined as follows:

$$L = w_g \mathcal{L}_g + w_p \mathcal{L}_p + w_x \mathcal{L}_x \tag{12}$$

$\mathcal{L}_p$ denotes the mean-square-error (MSE) loss, and $\mathcal{L}_g$ denotes the global and local discriminator loss function using RaGAN-LS loss, as shown in the following equation. I_X denotes the fuzzy image domain, and I_Y denotes the clear image domain. x is the fuzzy image, and y is the corresponding clear image. D is the discriminator, and G is the generator.

$$\begin{aligned} \mathcal{L}_g \quad &= \mathrm{E}_{x \sim I_X(x)} \left[\left(D(x) - \mathrm{E}_{y \sim I_Y(y)} D(G(y) - 1) \right)^2 \right] \\ &+ \mathrm{E}_{y \sim I_Y(y)} \left[\left(D(G(y)) - \mathrm{E}_{x \sim I_X(x)} D(G(x) + 1) \right)^2 \right]. \end{aligned} \tag{13}$$

$\mathcal{L}_x$ denotes the perceptual loss between the clear image and the recovered blurred image in the conv3.3 feature map of VGG-19.

$$\mathcal{L}_x = \frac{1}{W_{i,j} H_{i,j}} \sum_{w=1}^{W_{i,j}} \sum_{h=1}^{H_{i,j}} \left(\varnothing_{i,j}(Y)_{w,h} - \varnothing_{i,j}(G(X))_{w,h} \right)^2, \tag{14}$$

where i, j denotes the feature map sequence number of the i maximum pooling layer followed by the j convolution (after activation) within the VGG19 network; $\varnothing_{i,j}$ denotes the feature map; and $W_{i,j}$ and $H_{i,j}$ are the dimensions of this feature map.

4. Results

4.1. Experimental Setup

Experimental details: All the experiments were conducted on a computer equipped with NVIDIA RTX A5000 (24 G memory × 4 cards) (NVIDIA Corporation, Santa Clara, CA, USA), using an Intel Xeon Gold 5320 CPU (Intel Corporation, Santa Clara, CA USA) with 128 GB of RAM. The system used was a 64-bit Ubuntu 18.01.6, and the deep-learning framework consisted of PyTorch 1.9, Python 3.8, and CUDA 11.1. The model was trained using the Adam optimization algorithm, using a warmup strategy during training (15 epochs),

setting the initial learning rate to 0.02. and starting decay linearly to 10^{-6} at 100 epochs, for a total of 500 epochs of training. Using the progressive method of training, we first trained the model on 128*128 images for 200 epochs, followed by 256*256 images for 200 epochs and then 512*512 images for 100 epochs. The method used in this study takes 0.42 S to inference a 512*512 image.

The RSML_C and RSML_S datasets, containing remote sensing images with multiple blur levels, were constructed using AMD-GAN [44]. The RSML_C dataset includes images captured using a DMK 33GX290-GigE black-and-white industrial camera (The Imaging Source Europe GmbH, Bremen, Germany), which outputs blurred images at four exposure times ranging from short to long. For the RSML_S dataset, on the other hand, a random trajectory generation method was used to produce motion blur kernels of varying sizes, creating images with multiple levels of blur.

For the GoPro dataset [23], a GoPro Hero 4 was used to capture high-resolution video and extract sharp-blurred image pairs.

The images in the RealBlur dataset [45] were captured directly via a camera in a dimly lit environment, and the output images blurred when the filming equipment vibrated; a dataset acquired in this way more closely matches the real environment..

4.2. Experimental Comparison Results

Comparison metrics: We used the PSNR as well as SSIM as evaluation metrics for assessing the strengths and weaknesses of the algorithms.

Baseline models: We quantitatively analyzed the performance of DeblurGAN-v2 [22], MAXIM [26], ESTRNN [46], DBGAN [47], DBCPeNet [48], SDWNet [49], LEBMD [50], MPRNet [25], BANet [51], Stripformer [19], DeepRFT-v2 [52], Uformer-B [18], Restormer [29], and AdaRevD [53] for the datasets.

As with AMD-GAN [44], we compared the models' performance using two datasets, the RSML_S dataset and the RSML_C dataset, training the models on the GOPRO dataset and testing them on the RSML_S and RSML_C datasets. As shown in Table 1, the proposed algorithm exhibited high values for both PSNR and SSIM. Compared to AdaRevD, KT-Deblur achieved higher PSNR and SSIM values.

Table 1. Mean PSNR and SSIM values for different deblurring methods on the RSML_S dataset (trained using GOPRO). The best results are shown in bold.

Method	PSNR	SSIM
DeblurGAN-v2 [22]	23.83	0.722
MAXIM [26]	24.58	0.786
MPRNet [25]	24.74	0.785
Restormer [29]	25.11	0.809
DeepRFT-v2 [52]	26.82	0.831
AdaRevD [53]	27.95	0.836
Our method	**29.21**	**0.878**

As shown in Figure 5, we selected six representative image-deblurring methods for comparative experiments: DeblurGAN-v2, based on generative adversarial networks; MPRNet, employing multi-stage progressive restoration; AdaRevD, utilizing adaptive reversible networks; MAXIM, based on a multi-axis MLP; DeepRFT-v2, using residual fast Fourier transform; and Restormer, with a transformer architecture. There were significant differences in these methods' restoration effects. Due to the instability of adversarial training, DeblurGAN-v2 produced images with extensive blurring. While MPRNet and AdaRevD ameliorated global blurring, they still failed to address blurring issues when processing targets such as aircraft. Benefiting from their long-range dependency modeling capabilities,

MAXIM and Restormer yielded relatively clear image subjects, but they generated ghosting artifacts when processing small targets (e.g., vehicles), a result that may be attributed to their insufficient modeling of local motion blur. DeepRFT-v2 achieved better overall visual effects through the integration of frequency transformation but still had limitations in restoring fine textures. In contrast, the proposed method, via introducing the KAN and a joint neighborhood-global attention feature extraction module, not only maintained clear global structures but also better recovered image details and target textures, demonstrating significant advantages, particularly in handling complex blurring scenarios.

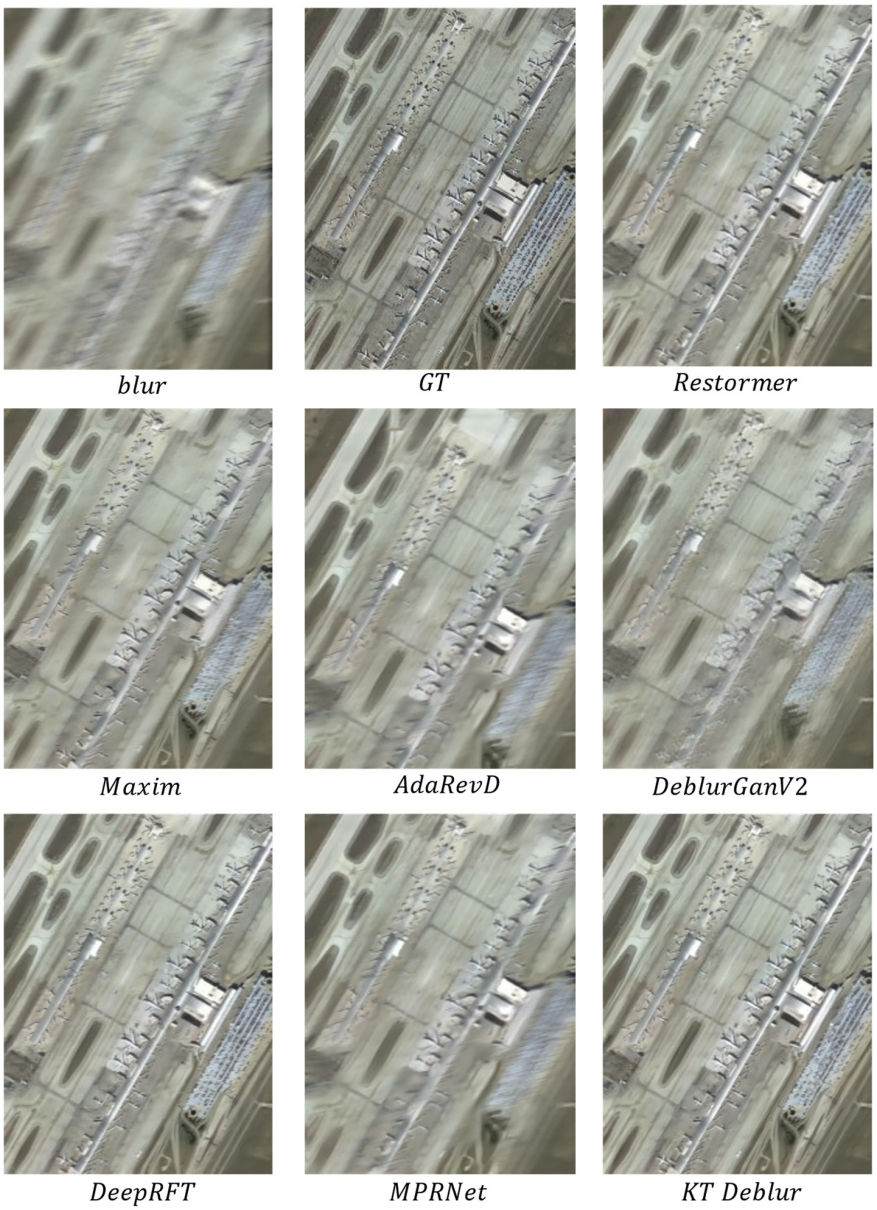

Figure 5. Recovery comparison results for the RSML_S dataset.

The test results for different algorithms with respect to the RSML_C dataset are shown in Table 2.

The recovery results for the RSML_C dataset are shown in Figure 6. The algorithm proposed in this paper performed outstandingly in these examples and largely succeeded in recovering the original details of the aircraft targets in the images. Compared with other state-of-the-art algorithms, KT-Deblur significantly reduces aircraft artifacts, can better preserve the detailed texture of images, and avoids visible target distortion. These results

demonstrate the robustness of KT-Deblur in dealing with complex blurring situations, especially in target detail recovery.

Table 2. Mean PSNR and SSIM for different deblurring methods applied to the RSML_C dataset (trained using GOPRO). The best results are shown in bold.

Method	PSNR	SSIM
DeblurGAN-v2 [22]	19.70	0.670
MAXIM [26]	21.78	0.685
MPRNet [25]	22.99	0.705
Restormer [29]	23.22	0.712
DeepRFT-v2 [52]	23.56	0.721
AdaRevD [53]	23.98	0.728
Our method	**24.45**	**0.792**

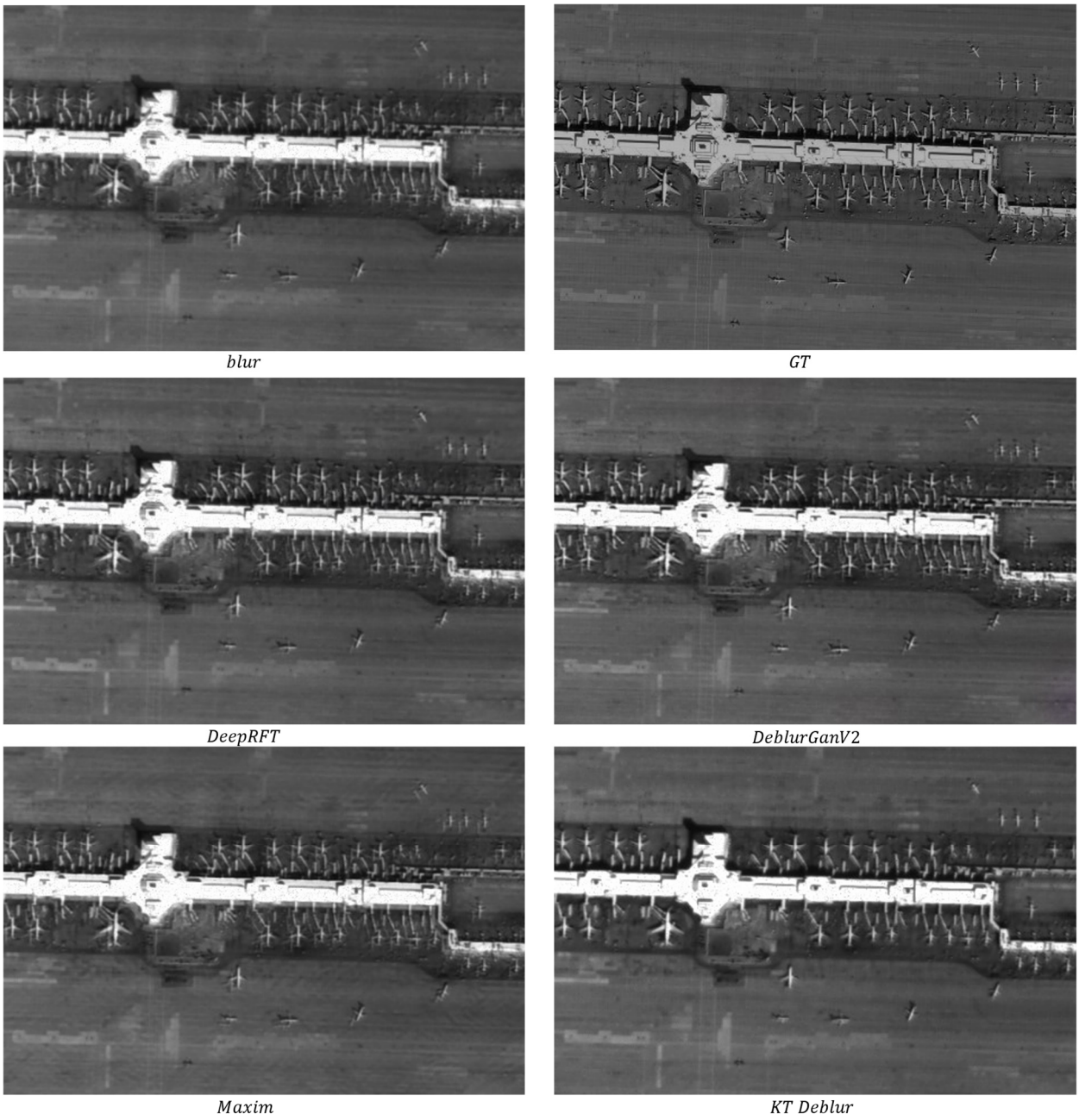

Figure 6. Recovery comparison results for the RSML_C dataset.

As shown in Tables 3 and 4, our method achieved 0.978/34.82 dB and 0.979/41.25 dB on the GOPRO dataset and RealBlur-R dataset, respectively, results that are superior to those obtained using the other deblurring methods. The algorithm in this paper achieved SOTA performance on the RealBlur-R dataset.

Table 3. Average PSNR and SSIM for different deblurring methods on the GoPro dataset. The best results are shown in bold.

Method	PSNR	SSIM
ESTRNN [46]	31.07	0.902
DBGAN [47]	31.10	0.942
DBCPeNet [48]	31.10	0.945
SDWNet [49]	31.36	-
LEBMD [50]	31.79	0.949
Restormer [29]	33.57	0.966
AdaRevD [53]	34.60	0.972
Our method	**34.82**	**0.978**

Table 4. Mean PSNR and SSIM for different deblurring methods on the RealBlur-R dataset. The best results are shown in bold.

Method	PSNR	SSIM
MPRNet [25]	39.31	0.972
MAXIM [26]	39.45	0.961
BANet [51]	39.55	0.971
Stripformer [19]	39.84	0.974
AdaRevD [53]	41.19	**0.979**
Our method	**41.25**	**0.979**

4.3. Ablation Study

4.3.1. The Effectiveness of the KAN Structures

The KAN was replaced with an MLP in the generator and tested on the public dataset RealBlur-R to verify the effectiveness of the KAN layer. The results are shown in Table 5, showing that the average PSNR improved by 0.8 dB.

Table 5. Results of ablation experiments on the RealBlur-R dataset in terms of PSNR and SSIM (KAN layer).

Method	PSNR	SSIM
KT-Deblur without KAN layer	40.45	0.968
KT-Deblur with KAN layer	41.25	0.979

4.3.2. The Effectiveness of the Union Attention Feature Extraction Module

The UAFE module in the generator structure was removed and tested on the public dataset RealBlur-R to verify the effectiveness of UAFE. The results are shown in Table 6, showing that the average PSNR improved by 2.13 dB.

Table 6. Results of ablation experiments on the RealBlur-R dataset in terms of PSNR and SSIM (UAFE).

Method	PSNR	SSIM
KT-Deblur without UAFE	39.12	0.954
KT-Deblur with UAFE	41.25	0.979

4.3.3. The Effectiveness of Loss Functions

Adversarial loss is the key to ensuring that KT-Deblur is capable of generative confrontation. Ablation tests were conducted on the public dataset RealBlur-R in regard to perceptual loss and reconstruction loss. The results are shown in Table 7.

Table 7. Results of ablation experiments on the RealBlur-R dataset in terms of PSNR and SSIM (loss functions).

Method	PSNR	SSIM
KT-Deblur without $\mathcal{L}_p$	29.67	0.854
KT-Deblur without $\mathcal{L}_x$	30.23	0.873
KT-Deblur	41.25	0.979

5. Conclusions

We proposed a novel image deblurring method, the KT-Deblur, which is the first to integrate KANs in image deblurring. Additionally, a UAFE module was innovatively designed. By dynamically adjusting the activation functions, the model can adapt to different types of blurring and autonomously learn the local features of images, significantly enhancing detail restoration and deblurring performance. The experimental results demonstrate the superior performance of the proposed method across both custom-built remote sensing datasets and publicly available deblurring datasets. Notably, on the RealBlur-R dataset, the average PSNR reached 41.25 dB, outperforming the latest SOTA algorithms.

During the experiments, it was found that the performance for all the methods decreased significantly in the case of extreme image blurring. Our method relies on the paired dataset. Future research will focus on optimizing the construction of the KAN combined with the self-supervised learning strategy in order to further enhance image-deblurring performance.

Author Contributions: Conceptualization, B.Z. and Q.L.; methodology, B.Z.; software, B.Z.; validation, B.Z., Z.L. and K.Z.; writing—original draft preparation, B.Z.; writing—review and editing, B.Z. and Z.T.; project administration, Q.L.; funding acquisition, Z.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (No. 2022YFB3904800).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare there are no conflicts of interest.

References

1. Hosseiny, B.; Rastiveis, H.; Homayouni, S. An Automated Framework for Plant Detection Based on Deep Simulated Learning from Drone Imagery. *Remote Sens.* **2020**, *12*, 3521. [CrossRef]
2. Cao, Z.; Kooistra, L.; Wang, W.; Guo, L.; Valente, J. Real-Time Object Detection Based on UAV Remote Sensing: A Systematic Literature Review. *Drones* **2023**, *7*, 620. [CrossRef]
3. Goncalves, V.P.; Werneck Ribeiro, E.A.; Imai, N.N. Mapping Areas Invaded by *Pinus* Sp. from Geographic Object-Based Image Analysis (GEOBIA) Applied on RPAS (Drone) Color Images. *Remote Sens.* **2022**, *14*, 2805. [CrossRef]
4. Feng, P.; Li, C.; Zhang, S.; Meng, J.; Long, J. Integrating Shipborne Images with Multichannel Deep Learning for Landslide Detection. *J. Earth Sci.* **2024**, *35*, 296–300. [CrossRef]
5. Zhao, X.; Wang, L.; Zhang, Y.; Han, X.; Deveci, M.; Parmar, M. A Review of Convolutional Neural Networks in Computer Vision. *Artif. Intell. Rev.* **2024**, *57*, 99. [CrossRef]
6. Zhang, J.; Li, J.; Li, J.; Sun, Y.; Liu, X.; Zheng, Z.; Lu, G. MBRVO: A Blur Robust Visual Odometry Based on Motion Blurred Artifact Prior. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8418–8425. [CrossRef]

7. Xing, W.; Tuo, W.; Li, X.; Wang, T.; Yang, C. Micro-Vibration Suppression and Compensation Techniques for in-Orbit Satellite: A Review. *Chin. J. Aeronaut.* **2024**, *37*, 1–19. [CrossRef]
8. Zhang, K.; Wang, T.; Luo, W.; Ren, W.; Stenger, B.; Liu, W.; Li, H.; Yang, M.-H. MC-Blur: A Comprehensive Benchmark for Image Deblurring. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 3755–3767. [CrossRef]
9. Bai, Y.; Cheung, G.; Liu, X.; Gao, W. Graph-Based Blind Image Deblurring From a Single Photograph. *IEEE Trans. Image Process.* **2019**, *28*, 1404–1418. [CrossRef]
10. Bai, Y.; Jia, H.; Jiang, M.; Liu, X.; Xie, X.; Gao, W. Single-Image Blind Deblurring Using Multi-Scale Latent Structure Prior. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 2033–2045. [CrossRef]
11. Blind Image Deblurring via Superpixel Segmentation Prior | IEEE Journals & Magazine | IEEE Xplore. Available online: https://ieeexplore.ieee.org/abstract/document/9410300 (accessed on 20 January 2025).
12. Wen, F.; Ying, R.; Liu, Y.; Liu, P.; Truong, T.-K. A Simple Local Minimal Intensity Prior and an Improved Algorithm for Blind Image Deblurring. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2923–2937. [CrossRef]
13. Gao, X.; Qiu, T.; Zhang, X.; Bai, H.; Liu, K.; Huang, X.; Wei, H.; Zhang, G.; Liu, H. Efficient Multi-Scale Network with Learnable Discrete Wavelet Transform for Blind Motion Deblurring. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 2733–2742.
14. Xiang, Y.; Zhou, H.; Li, C.; Sun, F.; Li, Z.; Xie, Y. Deep Learning in Motion Deblurring: Current Status, Benchmarks and Future Prospects. *Visual Comput.* **2024**. [CrossRef]
15. Pham, B.-D.; Tran, P.; Tran, A.; Pham, C.; Nguyen, R.; Hoai, M. Blur2Blur: Blur Conversion for Unsupervised Image Deblurring on Unknown Domains. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 2804–2813.
16. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. KAN: Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2404.19756.
17. Liu, Z.; Ma, P.; Wang, Y.; Matusik, W.; Tegmark, M. KAN 2.0: Kolmogorov-Arnold Networks Meet Science. *arXiv* **2024**, arXiv:2408.10205.
18. Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; Li, H. Uformer: A General u-Shaped Transformer for Image Restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17683–17693.
19. Tsai, F.-J.; Peng, Y.-T.; Lin, Y.-Y.; Tsai, C.-C.; Lin, C.-W. Stripformer: Strip Transformer for Fast Image Deblurring. In *European Conference on Computer Vision*; Springer Nature Switzerland: Cham, Switzerland, 2022.
20. Chen, L.; Chu, X.; Zhang, X.; Sun, J. Simple Baselines for Image Restoration. In Proceedings of the Computer Vision—ECCV 2022; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 17–33.
21. Zhu, B.; Lv, Q.; Yang, Y.; Sui, X.; Zhang, Y.; Tang, Y.; Tan, Z. Blind Deblurring of Remote-Sensing Single Images Based on Feature Alignment. *Sensors* **2022**, *22*, 7894. [CrossRef]
22. Kupyn, O.; Martyniuk, T.; Wu, J.; Wang, Z. Deblurgan-v2: Deblurring (Orders-of-Magnitude) Faster and Better. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 8878–8887.
23. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 3883–3891.
24. Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; Matas, J. Deblurgan: Blind Motion Deblurring Using Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8183–8192.
25. 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 2021 IEEE/Cvf Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; IEEE Computer Society, Los Alamitos, CA, USA, 2021; pp. 14816–14826.
26. Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; Li, Y. Maxim: Multi-Axis Mlp for Image Processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5769–5780.
27. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
28. Kim, T.H.; Sajjadi, M.S.M.; Hirsch, M.; Schölkopf, B. Spatio-Temporal Transformer Network for Video Restoration. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.

29. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5728–5739.
30. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
31. Gao, Z.; Xie, J.; Wang, Q.; Li, P. Global Second-Order Pooling Convolutional Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3024–3033.
32. Huang, W.; Deng, Y.; Hui, S.; Wang, J. Adaptive-Attention Completing Network for Remote Sensing Image. *Remote Sens.* **2023**, *15*, 1321. [CrossRef]
33. Wang, C.; Meng, L.; Gao, Q.; Wang, J.; Wang, T.; Liu, X.; Du, F.; Wang, L.; Wang, E. A Lightweight Uav Swarm Detection Method Integrated Attention Mechanism. *Drones* **2023**, *7*, 13. [CrossRef]
34. Cao, H.; Gao, Y.; Cai, W.; Xu, Z.; Li, L. Segmentation Detection Method for Complex Road Cracks Collected by UAV Based on HC-Unet++. *Drones* **2023**, *7*, 189. [CrossRef]
35. Chen, S.; Lan, J.; Liu, H.; Chen, C.; Wang, X. Helmet Wearing Detection of Motorcycle Drivers Using Deep Learning Network with Residual Transformer-Spatial Attention. *Drones* **2022**, *6*, 415. [CrossRef]
36. Kong, L.; Dong, J.; Ge, J.; Li, M.; Pan, J. Efficient Frequency Domain-Based Transformers for High-Quality Image Deblurring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5886–5895.
37. Hassani, A.; Walton, S.; Li, J.; Li, S.; Shi, H. Neighborhood Attention Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6185–6194.
38. Zhang, K.; Ren, W.; Luo, W.; Lai, W.-S.; Stenger, B.; Yang, M.-H.; Li, H. Deep Image Deblurring: A Survey. *Int. J. Comput. Vis.* **2022**, *130*, 2103–2130. [CrossRef]
39. Montanelli, H.; Yang, H. Error Bounds for Deep ReLU Networks Using the Kolmogorov–Arnold Superposition Theorem. *Neural Netw.* **2020**, *129*, 1–6. [CrossRef] [PubMed]
40. Fakhoury, D.; Fakhoury, E.; Speleers, H. ExSpliNet: An Interpretable and Expressive Spline-Based Neural Network. *Neural Netw.* **2022**, *152*, 332–346. [CrossRef] [PubMed]
41. Lai, M.-J.; Shen, Z. The Kolmogorov Superposition Theorem Can Break the Curse of Dimensionality When Approximating High Dimensional Functions. *arXiv* **2024**, arXiv:2112.09963.
42. He, J. On the Optimal Expressive Power of ReLU DNNs and Its Application in Approximation With the Kolmogorov Superposition Theorem. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, 1–14. [CrossRef]
43. Stergiou, A.; Poppe, R. AdaPool: Exponential Adaptive Pooling for Information-Retaining Downsampling. *IEEE Trans. Image Process.* **2023**, *32*, 251–266. [CrossRef] [PubMed]
44. Zhu, B.; Lv, Q.; Tan, Z. Adaptive Multi-Scale Fusion Blind Deblurred Generative Adversarial Network Method for Sharpening Image Data. *Drones* **2023**, *7*, 96. [CrossRef]
45. 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; Springer: Berlin/Heidelberg, Germany, 2020; pp. 184–201.
46. Zhong, Z.; Gao, Y.; Zheng, Y.; Zheng, B. Efficient Spatio-Temporal Recurrent Neural Network for Video Deblurring. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 191–207.
47. Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Stenger, B.; Liu, W.; Li, H. Deblurring by Realistic Blurring. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020; pp. 2734–2743.
48. Cai, J.; Zuo, W.; Zhang, L. Dark and Bright Channel Prior Embedded Network for Dynamic Scene Deblurring. *IEEE Trans. Image Process.* **2020**, *29*, 6885–6897. [CrossRef]
49. 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 2021), Montreal, BC, Canada, 11–17 October 2021; IEEE Computer Soc: Los Alamitos, CA, USA, 2021; pp. 1895–1904.
50. Jiang, Z.; Zhang, Y.; Zou, D.; Ren, J.; Lv, J.; Liu, Y. Learning Event-Based Motion Deblurring. In Proceedings of the 2020 IEEE/Cvf Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020; pp. 3317–3326.
51. 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]

52. Mao, X.; Liu, Y.; Liu, F.; Li, Q.; Shen, W.; Wang, Y. Intriguing Findings of Frequency Selection for Image Deblurring. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 1905–1913. [CrossRef]
53. Mao, X.; Li, Q.; Wang, Y. AdaRevD: Adaptive Patch Exiting Reversible Decoder Pushes the Limit of Image Deblurring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–24 June 2024; pp. 25681–25690.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.