applied sciences

MDPI

Article

Residual Dense Swin Transformer for Continuous-Scale Super-Resolution Algorithm

Jinwei Liu, Zihan Gui, Chenghao Yuan, Guangyi Yang * and Yi Gao

School of Electronic Information, Wuhan University, Luojia Mountain Road, Wuhan 430072, China; ljw0416@whu.edu.cn (J.L.); syro_gzh@whu.edu.cn (Z.G.); 2021302121401@whu.edu.cn (C.Y.); 2017301200229@whu.edu.cn (Y.G)

* Correspondence: ygy@whu.edu.cn

Abstract: The single-image super-resolution task benefits has a wide range of application scenarios, so has long been a hotspot in the field of computer vision. However, designing a continuous-scale super-resolution algorithm with excellent performance is still a difficult problem to solve. In order to solve this problem, we propose a continuous-scale SR algorithm based on a Transformer, which is called residual dense Swin Transformer (RDST). Firstly, we design a residual dense Transformer block (RDTB) to enhance the information flow before and after the network and extract local fusion features. Then, we use multilevel feature fusion to obtain richer feature information. Finally, we use the upsampling module based on the local implicit image function (LIIF) to obtain continuous-scale super-resolution results. We test RDST on multiple benchmarks. The experimental results show that RDST achieves SOTA performance in the fixed scale of super-resolution tasks in the distribution, and significantly improves (0.1∼0.6 dB) the arbitrary scale of super-resolution tasks out of distribution. Sufficient experiments show that our RDST can use fewer parameters,and its performance is better than the SOTA SR method.

Keywords: Transformer; super-resolution; continuous-scale

check for updates

Citation: Liu, J.; Gui, Z.; Yuan, C.; Yang, G.; Gao, Y. Residual Dense Swin Transformer for Continuous-Scale Super-Resolution Algorithm. *Appl. Sci.* **2024**, *14*, 3678. https://doi.org/10.3390/app14093678

Academic Editor: Andrea Prati

Received: 19 February 2024
Revised: 31 March 2024
Accepted: 2 April 2024
Published: 25 April 2024

Copyright: © 2024 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

Single-image super-resolution (SISR) refers to the technical means to restore a low-resolution image to a high-resolution image. It is widely used in the fields of medical imagery [1,2], remote sense image [3,4], monitoring, and security [5,6]. Therefore, this technology has long been a research hotspot in the field of computer vision. In most of today's application scenarios, people expect to enlarge an image to any scale without losing the high-frequency details of the image. However, because a low-resolution image can correspond to multiple different high-resolution images, SISR becomes an ill posed problem. How to use the single model to approximate the optimal solution in the super-resolution space of arbitrary scale amplification is still a difficult problem. Therefore, it is of great significance to study a continuous-scale super-resolution algorithm with excellent performance.

SISR algorithms can be divided into two categories: traditional methods and deep-learning-based methods. Yang [7] drew on the idea of compressed sensing, performed sparse representation of low-resolution images, and used prior knowledge to complete the dictionary learning of high-resolution images to achieve super-resolution reconstruction; Gao et al. [8] used locally linear embedding in manifold learning. To achieve linear mapping from low-resolution space to high-resolution space, both Glasner [9] and Huang [10] proposed the example-based super-resolution method; the difference between the methods being that the latter transforms the patch to find a more similar patch in the low-resolution to high-resolution images. However, the super-resolution effect of these traditional methods is limited, and they struggle to meet application requirements in real life. Algorithms based

on deep learning, especially based on convolutional neural networks, exhibit excellent performance that traditional methods do not have. Since Dong [11] first brought CNN into the SR field, countless SISR algorithms have been developed, and the SRCNN structure has been improved. Most of them use residual connections, dense connections, and iterative supervision to continuously deepen the CNN [12–15]. Although this approach solves the problem of limited receptive field caused by CNN fixed-size convolution to a certain extent, it still does not fundamentally solve the problem of global information loss. In addition, there are very few studies on super-resolution at any scale. Lim et al. [16] used multiple upsampling modules for training to achieve integer multiples of multiscale super-resolution. Refs. [17,18] used the pooling layer and local implicit functions to achieve arbitrary scales of super-resolution, but they both focused on building modules that can achieve arbitrary scales of upsampling, while ignoring the importance of feature extraction.

The development of the Transformer [19] in the field of CV, especially the emergence of the ViT [20] and Swin Transformer [21], has provided new ideas for scholars in the field of SISR. ViT was the first method to successfully apply Transformer to the computer field and achieve the same or even surpassing the effect of CNN. It slices the image and performs patch embedding, using it as the input sequence of the Transformer. Based on this, some scholars [22–24] have proposed performing super-resolution tasks and realized new SOTA performance at that time. On this basis, the Swin Transformer uses the idea of CNN to introduce a shift window to enhance the performance of network local feature extraction and reduce the amount of calculation. Based on this, ref. [25] introduced the idea of partial windows to further improve the super-resolution effect. It is obvious that none of these Transformer-based methods can make full use of the low-level and high-level information and cannot achieve image super-resolution on a continuous scale.

Benefiting from the inspiration of the Swin Transformer and LIIF, we propose a residual dense Swin Transformer to solve the continuous-scale super-resolution with excellent performance. We propose the residual dense Transformer block (RDTB) structure on the basis of the Swin transfomer. By introducing residual connections and dense connections, we realize information interaction between all levels, propose local feature fusion (LFF) to promote feature fusion within the block, and design global feature fusion (GFF) to achieve information flow between blocks. Through the information complementation between the bottom and high levels, the network can pay attention to the low-frequency and high-frequency information of the image at the same time; the Transformer's self-attention mechanism can be used to take into account the local and global information in the image. We combine the patch-embedded characteristic of the Transformer with the implicit local continuous expression of the image, and we better combine feature extraction with the upsampling module to achieve continuous-scale super-resolution reconstruction.

In summary, our contributions are as follows:

(1) A high-performance super-resolution network RDST is proposed. The network makes full use of the low- and high-level information in the image and is combined with an LIIF upsampling module to achieve continuous-scale super-resolution reconstruction of a single model.
(2) A novel RDTB structure is proposed, which uses LFF to perform local information fusion on features within blocks and uses GFF to perform global information fusion on the features between blocks. At the same time, it combines the shallow information to fully explore the information expression in low-resolution images.
(3) Through a comparison experiment with a fixed multiple in the distribution and a continuous-scale super-resolution experiment on the benchmark, it is shown that RDST is equal to the state-of-the-art (SOTA) method in the super-resolution results for the fixed multiple, and the super-resolution results at magnification outside the dataset distribution are greatly improved.

2. Related Work

This section gives a brief review of the CNN-based and Transformer-based SISR methods.

2.1. CNN-Based Super-Resolution Method

With the rise of deep learning, especially convolutional neural networks, the SISR algorithm based on CNN has made brilliant achievements. The SRCNN [7] proposed by Dong et al. is the pioneering work of CNN applied to SR. With the help of sparse coding, they introduced CNN to SR tasks, creating a precedent for the study of SISR based on deep learning. Later, in response to the slower speed of SRCNN [7], they proposed FSRCNN [26], which uses postsampling and deconvolution layers to reduce network parameters, which greatly improves the speed of the algorithm, but the super-resolution effect is not improved compared to that of SRCNN [7]. The VDSR [12] of Kim et al. enhances the super-resolution effect by deepening the network structure, but the problem that it creates is that the network parameters are greatly increased. DRCN [13] uses the idea of iteration to deepen the network, and residual learning and recursive supervision strategies are used to stabilize the network training process. Although the network parameters are reduced, the calculation amount is not, and there is also the problem that the network is difficult to train. In order to overcome the training difficulties caused by network deepening, SRResNet introduces the idea of local residuals. DRRN [27] combines the ideas of local residuals, global residuals, and convolutional layer recursion to reduce the computational cost and improve the effect of the algorithm. In EDSR, the BN layer in the residual block is not needed, and it also stacks deeper networks by reducing the computational cost. SRDenseNet draws on the idea of DenseNet [28] and uses the complementary fusion of features of different depths for super-segmentation tasks. RDN [15] is a further improvement on DRRN. It applies a dense residual block and introduces local residual learning and global residual learning to improve the effect of the model. RCAN [29] introduces channel attention into the residual block and uses the RCAB structure to improve network expression ability. MSRN [30] extracts rich feature information from a multiscale perspective. Although the CNN-based methods have made many achievements, the characteristics of the convolution kernel always limit the global feature extraction ability of such networks, which cannot fundamentally achieve the effective fusion of global and local features.

2.2. Transformer-Based Super-Resolution Method

After [19,31,32] made brilliant achievements in the field of NLP, scholars have tried to apply Transformer to the field of computer vision, challenging the dominance of CNN in computer vision. With the introduction of ViT, DeiT [33], Swin Transformer, etc., scholars have proposed a Transformer-based SISR. IPT [22] introduces Transformer into the underlying visual tasks and uses ImageNet pretraining and multitask learning and performs well on the dataset of SISR tasks; ESRT [23] combines the backbone of CNN with Transformer and uses Transformer's powerful global modeling capabilities to enhance the CNN. Swin-IR [25] includes the RSTB structure based on Swin Transformer, effectively using the sliding window mechanism to achieve long-distance modeling and using fewer parameters to obtain better performance. Although these algorithms have achieved varying degrees of improvement, they are currently based on Transformer. All of the super-resolution algorithms focus on how to apply Transformer to a fixed-multiple super-resolution image task. They fail to solve the super-resolution task of any multiple, and they fail to make full use of the low-level information and high-level information in the network.

3. Methodology

3.1. Network Architecture

As can be seen in Figure 1, the RDST proposed in this paper is composed of three main parts: shallow feature extraction, multilevel feature extraction, and an upsampling module. Multilevel feature extraction consists of several RDTBs and multilevel feature fusion composition. The input of the model is a low-resolution RGB image, and the output is a continuous-scale super-resolution image.

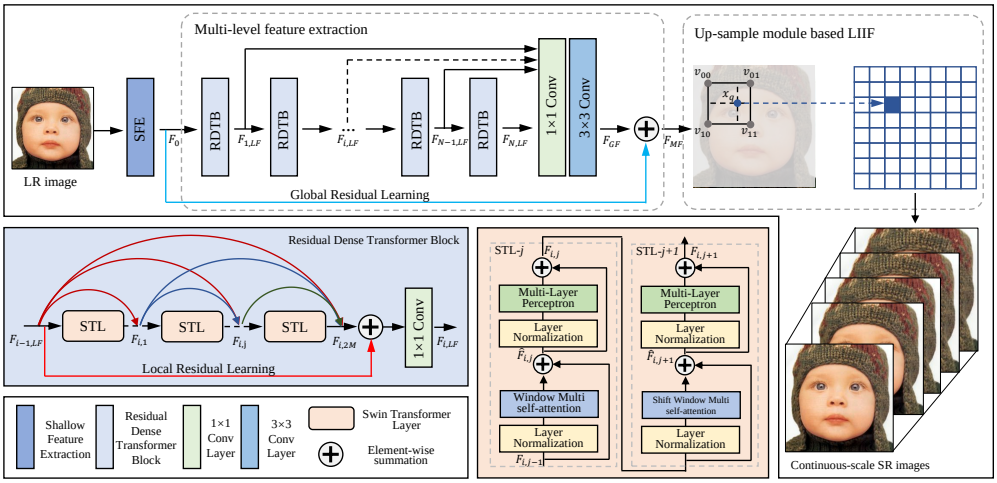

Figure 1. Flowchart of residual dense Swin Transformer.

3.1.1. Shallow and Multilevel Feature Extraction

First, we use a 3 × 3 convolutional layer to extract the shallow features from low-resolution images $I_{LR} \in \mathbb{R}^{H \times W \times C_{in}}$, which is expressed in Formula (1):

$$F_0 = H_{SFE}(I_{LR}) \tag{1}$$

where $H_{SFE}(\cdot)$ refers to the shallow feature extraction module, and H, W, C_{in}, and C_{out} are the length, width, number of channels and the number of output channels of the shallow features, respectively. On the one hand, the application of the convolutional layer can make good use of the underlying features of the image to restore an image that is more in line with the perception of the human eye. On the other hand, it is conducive to subsequent global residual learning and stabilizes the training process of the network. Subsequently, we use multiple RDTBs to extract each level's features $F_{i,LF} \in \mathbb{R}^{H \times W \times C_{out}}$, which is expressed in Formula (2):

$$F_{i,LF} = H_{RDTB-i}(F_{i-1,LF}) i \in 1, 2, 3, \cdots, N \tag{2}$$

where $H_{RDTB-i}(\cdot)$ represents the ith RDTB, and $F_{i-1,LF}$ is the feature extracted by the ith RDTB. Each RDTB block takes the output of the previous RDTB block as the input, uses the Swin Transformer layer (STL) in the block to extract image features, and uses local feature fusion. To enhance the feature interaction within the block, local residuals are introduced to connect the training process of stabilizing the network and strengthen the feature expression ability of the network. Finally, the final feature expression is obtained through the multilevel feature fusion module $H_{MLFF}(\cdot)$, which is expressed in Formula (3):

$$\begin{aligned} F_{MF} &= H_{MLFF}(F_0, F_{1,LF}, \cdots, F_{N,LF}) \\ &= H_{GFF}(F_{1,LF}, \cdots, F_{N,LF}) + F_0 \end{aligned} \tag{3}$$

Here, $H_{GFF}(\cdot)$ represents the global feature fusion function between blocks. Through multilevel feature fusion and the introduction of global residuals, the network makes full use of the low-level and high-level features in the image to improve the network's reconstruction effect.

3.1.2. Upsampling Module Using LIIF

Inspired by [18], we use the local image implicit function $f_\theta(\cdot)$ in the upsampling module to express the discrete image continuously, namely, $I = f_\theta(z, x)$. The input of the function is any coordinate x to be predicted, the corresponding feature vector is z, and the output is the RGB value I at this coordinate. The corresponding eigenvectors of the actual predicted coordinates cannot be obtained directly, so they are estimated by using the eigenvectors of the four nearest coordinates around the predicted coordinates.

The specific super-resolution process is as follows: We first perform feature unfolding on the fusion feature F_{MF} in the upsampling module, and we use our own information to enrich each feature vector in the feature map. The specific method is expressed in the following formula:

$$F_{MF}^{(n,i,j)} = Concat\left(\left\{F_{MF}^{(n,i+k,j+k)}\right\}\right) \ k \in -1,0,1 \tag{4}$$

where $F_{MF}^{(n,i,j)}$ represents the nth feature vector in fusion feature F_{MF} and its coordinates are (i,j). Then, we use the nearby feature vector to predict the RGB value of the corresponding coordinate x_q; the specific process is as follows:

$$I^{(n)}(x_q) = \sum_{t \in 00,01,10,11} \frac{S_t}{S} \cdot f_\theta(F_{MF}^{(n,i,j)}, x_q - v_t^{(n)}) \tag{5}$$

where $v_t^{(n)}$ represents the coordinates of the corresponding feature vector $F_{MF}^{(n,i,j)}$, S_t is the rectangular area of the diagonal coordinates of x_q, $v_t^{(n)}$, S is the total area corresponding to the four eigenvector coordinates, and $f_\theta(\cdot)$ represents the function of the RGB value of the predicted coordinates. Considering that the relationship between the position of the pixel to be predicted and its surrounding pixels is different when the actual magnification is different, the cell parameter is also introduced into the function $f_\theta(\cdot)$, which refers to the size of the pixel under different magnifications. In the actual prediction process, a five-layer MLP can be used to achieve a super-resolution, continuous-scale of image.

3.2. Residual Dense Transformer Block

As can be seen in Figure 1, RDTB is composed of several STLs and a convolutional layer. Taking the *i*th RDTB as an example, for the input fusion feature $F_{i-1,LF}$, feature extraction and learning are performed through the multilayer STL, and local feature fusion is used to interactively flow features at different levels to enhance RDTB's local information extraction capacity. Finally, the residual connection is introduced to obtain the fusion feature $F_{i,LF}$. It is expressed in Formula (6).

$$F_{i,LF} = H_{LLF}(F_{i-1,LF}) \tag{6}$$

where $H_{LLF}(\cdot)$ represents the local feature fusion function in the block.

Swin Transformer Layer

STL is improved from the Transformer structure based on self-attention. This specific structure is shown in Figure 1. It uses window multihead self-attention (W-MSA) to calculate the global attention within the window and solves the problem of the huge computational cost of the Transformer for the image; it also uses shift window–multihead self-attention (SW-MSA) to realize window information interaction between the two so as to achieve global information modeling. The specific process is expressed in the following formula:

$$\hat{F}_{i,j} = W_M SA\left(LN\left(F_{i,j-1}\right)\right) + F_{i,j-1} \tag{7}$$

$$F_{i,j} = MLP\left(LN\left(\hat{F}_{i,j}\right)\right) + \hat{F}_{i,j} \tag{8}$$

$$\hat{F}_{i,j+1} = SW_M SA\left(LN\left(F_{i,j}\right)\right) + F_{i,j} \tag{9}$$

$$F_{i,j+1} = MLP\left(LN\left(\hat{F}_{i,j+1}\right)\right) + \hat{F}_{i,j+1} \tag{10}$$

where $F_{i,j-1}$ represents the output feature of the jth STL in the ith RDTB, $\hat{F}_{i,j}$ is the output feature of W-MSA, and $j \in 2,4,\cdots,2M$, $LN(\cdot)$ is layer normalization; since STL calculates the self-attention of the patch in the window, its position coding method is also different

from that of the traditional ViT. Using relative position coding, the calculation of the self-attention mechanism in the window can be expressed as

$$Attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d}} + B)V \quad (11)$$

where Q, K, and V are query, key, and value matrices, respectively; and B is the relative position weight that can be learned.

3.3. Multilevel Feature Fusion

It can be seen in Figure 1 that after each RDTB extracts local fusion features, we propose the use of multilevel feature fusion, which makes full use of the low-level information and high-level information extracted from the network to enhance the feature expression ability of the network. Multilevel feature fusion can be divided into two steps: global feature fusion and global residual learning.

3.3.1. Global Feature Fusion

Global feature fusion performs further information exchange on the local fusion features extracted from each level of RDTB. By concat splicing each level of fusion features $F_{i,LF}$, first use 1×1 convolution to achieve channel-dimensional information interaction and reduce network parameters, and then use 3×3 convolution to enhance local context information to obtain global fusion features F_{GF}, which can be expressed as

$$F_{GF} = Conv_{3\times3}Conv_{1\times1}(Concat([F_{1,LF}, \cdots, F_{i,LF}])) \quad (12)$$

where $Concat(\cdot)$ represents the splicing of channel dimensions.

3.3.2. Global Residual Learning

In order to introduce more high-frequency information from the image, before upsampling, we use global residual learning to connect the shallow features F_0 extracted above and the global fusion features F_{GF} with long jumps to obtain the final multiscale fusion feature F_{MF}. The application of long-hop connections enables the network to learn residual information at a coarse-grained level, which further improves the ability to express features. The specific process can be expressed as

$$F_{MF} = F_{GF} + F_0 \quad (13)$$

4. Experiments

4.1. Dataset and Metrics

During the training process, we used 800 high-definition images in DIV2K [34] as the model's training set; in the testing phase, we evaluated the model on several recognized benchmarks: Set5 [35], Set14 [36], BSD100 [37], Urban100 [36], and Manga109 [38]. At the same time, in order to evaluate and compare SR algorithms more objectively, we used PSNR and SSIM [39] as indicators to measure model performance. It is worth noting that the Transformer-based SR method needs to process the image block, so the algorithm in this paper had the same data boundary processing as SwinIR in the experiment.

4.2. Implementation Details

During the training process, we set the RDTB number, STL number, window size, embed dim and attention head number to 6, 6, 8, 64 and 8, respectively. We randomly cropped low-resolution images into 48×48 tiles as the input. We used the Adam optimizer to train the model for 1000 rounds, the batch size was set to 64, the initial learning rate was set to 0.0001, and the learning rate was halved every 200 rounds. In the training phase, it was ensured that the magnification of each batch of images was the same, and the value of the magnification was randomly distributed from 1 to 4. Our model was implemented

based on the pytorch framework and trained on 4 Tesla V100 GPUs. In this study, the L1 loss function was used to optimize and learn the parameters of RDST. The formula is as follows:

$$Loss = \|I_{HR} - I_{SR}\| \tag{14}$$

where I_{HR} and I_{SR} represent high-resolution images (gt) and reconstructed super-resolution images, respectively.

4.3. Comparative Experiment

We compared the algorithm in this paper with several typical fixed multiple SISR algorithms, including SRCNN, DRRN, SRDenseNet, EDSR, and RCAN. Each algorithm was tested on 5 benchmarks. It should be noted that the comparison algorithm indicators were from the original paper. The SRCNN and EDSR indicators in the Manga109 data were from RCAN, and the DRRN indicators were from RDN. RDST-s* means the RDST-s model trained on Div2K + Flickr2K

4.3.1. In-Distribution

Table 1 shows the PSNR index of each algorithm's ×2, ×3, and ×4 fixed multiples. It can be seen that the RDST in this paper achieved the best performance. Compared with the previous classic neural network algorithms SRCNN, DRRN, and SRDenseNet, RDST shows powerful feature extraction capabilities.

Table 1. Comparison with classical SISR methods. Best and second best performance are in red and blue colors, respectively.

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	33.67	0.9299	30.32	0.8688	29.55	0.8431	26.87	0.8403	30.82	0.9339
SRCNN	×2	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.5	0.8946	35.6	0.9663
DRRN	×2	37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188	37.6	0.9736
SRDenseNet	×2	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
EDSR	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.1	0.9773
RCAN	×2	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
RDST-s*	×2	38.32	0.9617	34.41	0.9243	32.44	0.9025	33.4	0.9398	39.73	0.9793
Bicubic	×3	30.4	0.8682	27.63	0.7742	27.2	0.7385	24.45	0.7349	26.95	0.8556
SRCNN	×3	32.75	0.909	29.3	0.8215	28.41	0.7863	26.24	0.7989	30.48	0.9117
DRRN	×3	34.03	0.9244	29.96	0.8349	28.95	0.8004	27.53	0.8378	32.42	0.9359
SRDenseNet	×3	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
EDSR	×3	34.65	0.928	30.52	0.8462	29.25	0.8093	28.8	0.8653	34.17	0.9476
RCAN	×3	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
RDST-s*	×3	34.82	0.9304	30.77	0.8501	29.36	0.812	29.28	0.8742	34.82	0.9519
Bicubic	×4	28.43	0.8104	26.09	0.7027	25.95	0.6675	23.14	0.6577	24.9	0.7866
SRCNN	×4	30.48	0.8628	27.5	0.7513	26.9	0.7101	24.52	0.7221	27.58	0.8555
DRRN	×4	31.68	0.8888	28.21	0.7721	27.38	0.7284	25.44	0.7638	29.18	0.8914
SRDenseNet	×4	32.02	0.8934	28.5	0.7782	27.53	0.7337	26.05	0.7819	-/-	-/-
EDSR	×4	32.46	0.8968	28.8	0.7876	27.71	0.742	26.64	0.8033	31.02	0.9148
RCAN	×4	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
RDST-s*	×4	32.66	0.9013	28.99	0.791	27.82	0.745	27.07	0.8147	31.8	0.9232

Figure 2 shows the visual effects of the algorithm in this paper and the classic SR algorithms with for super-resolution and fixed-size images. Figure 1 shows the results for four times, three times, and two times scale factors from top to bottom. For the "img078" in Urban100 and the "zebra" in Set14, the super-segmentation result of RDST preserves the texture details in the image. Compared with the other methods, it has fewer artifacts and is more suitable for human perception. For the "bird" in Set5, our super-score results are also

very close to the original HR results. The good visual effects show that RDST makes full use of multilevel features and Transformer's global modeling capabilities.

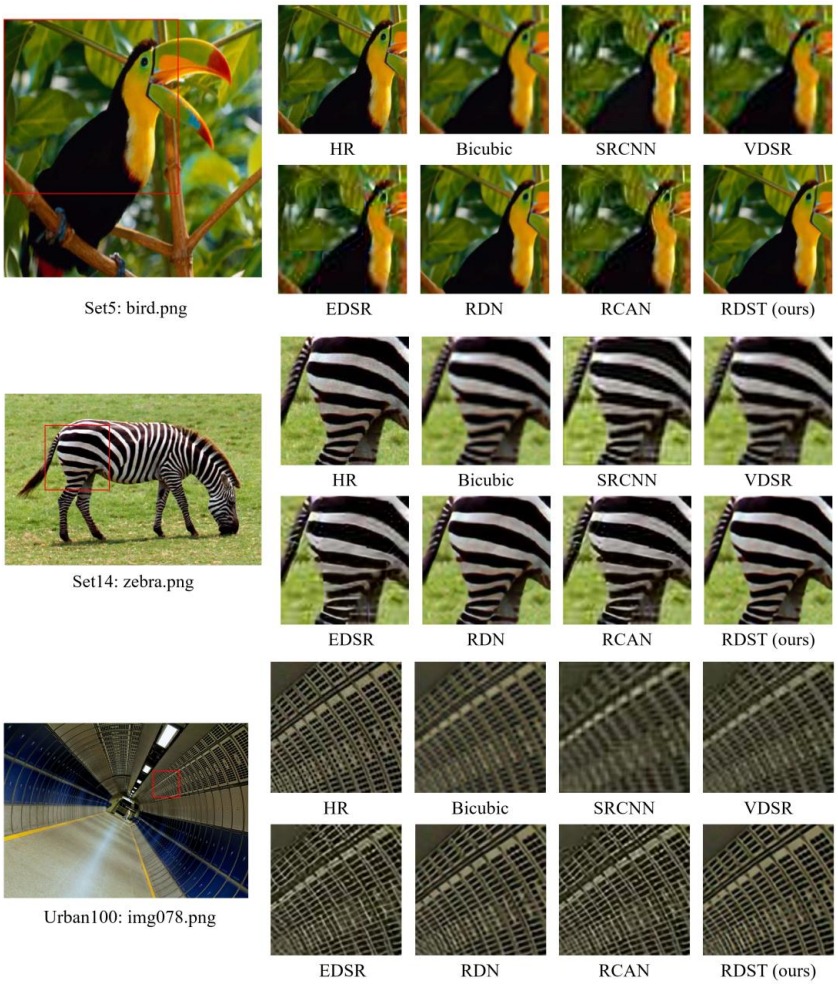

Figure 2. Visual results with a scale factor of 4, 3, and 2.

4.3.2. Out of Distribution

Different from the ordinary fixed magnification SR method, our proposed RDST can achieve super-resolution effects for any multiple with the help of LIIF. In order to further explore the combination capabilities of different encoders with LIIF, CNN-based models were selected, including EDSR baseline (EDSR$_{(b)}$) and RDN, and compared with the proposed RDST. RDST-t, RDST-s, and RDST-b refer to the tiny, small, and base versions of RDST, respectively. The number of RDTBs and the number of STLs in the block were four, six, and eight, respectively.

As Table 2 shows, RDST-s and RDST-b almost captured the best PSNR indicators for each scale. Especially for the PSNR index outside the distribution, our method is generally 0.1∼0.6 dB higher than the model based on CNN combined with LIIF. This finding fully proves the excellent generalization ability of RDST and the powerful feature extraction ability of the RDTB that we designed. The powerful extra-distribution performance is also due to the combination of Transformer's unique encoding and LIIF for continuous image expression. Figures 3–5, respectively, show the visual effects of 6 times, 18 times, and 30 times the super score. It can be clearly seen from the figures that our proposed RDST can also achieve good visual effects even at multiples outside of the hyperdivision distribution. Compared with other methods, RDST can better retain texture details such as "glass boundary" and "railing shape", retain more high-frequency details of the image, and produce super-resolution high-quality images that are more suitable for the human eye.

Table 2. Quantitative comparison (average PSNR) with CNN methods on benchmark datasets.Best and second best performance are in red and blue colors, respectively.

Dataset	Method	Metric	In Distribution			Out of Distribution					
			×2	×3	×4	×6	×8	×12	×18	×24	×30
Set5	bicubic	PSNR	33.67	30.40	28.43	25.93	24.40	22.56	20.95	20.03	19.32
		SSIM	0.9299	0.8682	0.8104	0.7206	0.6582	0.5948	0.5605	0.5485	0.5468
	Dense-LIIF	PSNR	37.74	34.13	31.89	28.65	26.69	24.39	22.36	21.26	20.42
		SSIM	0.9594	0.9250	0.8911	0.8245	0.7665	0.6846	0.6032	0.5752	0.5694
	EDSR(b)-LIIF	PSNR	37.99	34.40	32.21	28.94	27.01	24.60	22.51	21.40	20.49
		SSIM	0.9603	0.9271	0.8950	0.8316	0.7764	0.6949	0.6136	0.5803	0.5710
	RDN-LIIF	PSNR	38.17	34.68	32.50	29.15	27.14	24.86	22.66	21.50	20.57
		SSIM	0.9610	0.9292	0.8988	0.8361	0.7809	0.7070	0.6175	0.5829	0.5713
	RDST-t	PSNR	38.09	34.51	32.37	29.35	27.55	24.96	22.95	21.76	21.11
		SSIM	0.9607	0.9280	0.8971	0.8404	0.7916	0.7090	0.6277	0.5891	0.5795
	RDST-s	PSNR	38.17	34.69	32.52	29.58	27.71	25.15	23.10	21.91	21.18
		SSIM	0.9610	0.9293	0.8991	0.8447	0.7952	0.7171	0.6403	0.5978	0.5796
	RDST-b	PSNR	38.20	34.67	32.60	29.68	27.78	25.22	23.15	21.83	21.17
		SSIM	0.9612	0.9293	0.8998	0.8458	0.7969	0.7208	0.6414	0.5919	0.5822
Set14	bicubic	PSNR	30.32	27.63	26.09	24.34	23.19	21.72	20.44	19.60	19.02
		SSIM	0.8688	0.7742	0.7027	0.6174	0.5667	0.5145	0.4807	0.4637	0.4526
	Dense-LIIF	PSNR	33.35	30.13	28.41	26.26	24.77	23.01	21.50	20.55	19.68
		SSIM	0.9150	0.8381	0.7772	0.6893	0.6333	0.5662	0.5156	0.4881	0.4674
	EDSR(b)-LIIF	PSNR	33.60	30.34	28.63	26.47	24.93	23.13	21.61	20.66	19.81
		SSIM	0.9173	0.8430	0.7827	0.6963	0.6393	0.5715	0.5197	0.4919	0.4701
	RDN-LIIF	PSNR	33.97	30.53	28.80	26.64	25.15	23.24	21.73	20.78	19.85
		SSIM	0.9209	0.8470	0.7875	0.7028	0.6465	0.5779	0.5237	0.4955	0.4719
	RDST-t	PSNR	33.81	30.50	28.76	26.74	25.31	23.51	21.95	20.96	20.38
		SSIM	0.9199	0.8457	0.7854	0.7063	0.6527	0.5843	0.5312	0.5005	0.4834
	RDST-s	PSNR	33.98	30.62	28.88	26.87	25.46	23.62	22.06	21.00	20.46
		SSIM	0.9214	0.8478	0.7883	0.7108	0.6564	0.5876	0.5343	0.5020	0.4848
	RDST-b	PSNR	33.92	30.64	28.91	26.91	25.49	23.63	22.03	20.98	20.48
		SSIM	0.9209	0.8481	0.7889	0.7119	0.6576	0.5881	0.5345	0.5015	0.4852
B100	bicubic	PSNR	29.55	27.20	25.95	24.53	23.66	22.50	21.34	20.57	19.93
		SSIM	0.8431	0.7385	0.6675	0.5871	0.5440	0.5031	0.4746	0.4597	0.4514
	Dense-LIIF	PSNR	32.02	28.97	27.46	25.73	24.69	23.40	22.13	21.31	20.63
		SSIM	0.8968	0.8017	0.7315	0.6426	0.5907	0.5375	0.4981	0.4766	0.4642
	EDSR(b)-LIIF	PSNR	32.18	29.11	27.60	25.84	24.80	23.48	22.22	21.39	20.68
		SSIM	0.8992	0.8059	0.7368	0.6484	0.5960	0.5413	0.5009	0.4785	0.4653
	RDN-LIIF	PSNR	32.32	29.26	27.74	25.98	24.91	23.57	22.29	21.45	20.74
		SSIM	0.9010	0.8098	0.7420	0.6547	0.6018	0.5454	0.5033	0.4808	0.4669
	RDST-t	PSNR	32.24	29.18	27.66	26.03	25.13	23.81	22.70	21.68	21.29
		SSIM	0.8999	0.8078	0.7397	0.6579	0.6131	0.5534	0.5131	0.4842	0.4744
	RDST-s	PSNR	32.31	29.27	27.75	26.11	25.21	23.88	22.77	21.76	21.37
		SSIM	0.9009	0.8100	0.7426	0.6607	0.6161	0.5559	0.5147	0.4858	0.4758
	RDST-b	PSNR	32.34	29.29	27.77	26.13	25.22	23.89	22.77	21.76	21.37
		SSIM	0.9012	0.8104	0.7430	0.6615	0.6169	0.5565	0.5150	0.4862	0.4759
Urban100	bicubic	PSNR	26.87	24.45	23.14	21.63	20.73	19.61	18.63	18.03	17.61
		SSIM	0.8403	0.7349	0.6577	0.5635	0.5137	0.4658	0.4376	0.4260	0.4193
	Dense-LIIF	PSNR	31.50	27.72	25.72	23.47	22.20	20.71	19.47	18.72	18.18
		SSIM	0.9212	0.8421	0.7729	0.6687	0.6018	0.5250	0.4708	0.4461	0.4320
	EDSR(b)-LIIF	PSNR	32.15	28.21	26.16	23.80	22.48	20.91	19.63	18.84	18.30
		SSIM	0.9284	0.8538	0.7879	0.6848	0.6167	0.5357	0.4772	0.4496	0.4345
	RDN-LIIF	PSNR	32.87	28.82	26.68	24.20	22.79	21.15	19.80	19.00	18.44
		SSIM	0.9351	0.8662	0.8039	0.7029	0.6340	0.5488	0.4852	0.4548	0.4377
	RDST-t	PSNR	32.42	28.45	26.39	24.15	22.77	21.22	19.91	19.15	18.52
		SSIM	0.9310	0.8588	0.7950	0.7005	0.6320	0.5526	0.4911	0.4627	0.4414
	RDST-s	PSNR	32.82	28.82	26.71	24.38	22.98	21.40	20.03	19.27	18.61
		SSIM	0.9349	0.8660	0.8044	0.7104	0.6416	0.5605	0.4957	0.4665	0.4438
	RDST-b	PSNR	32.93	28.90	26.79	24.47	23.01	21.42	20.05	19.27	18.65
		SSIM	0.9356	0.8676	0.8065	0.7130	0.6436	0.5620	0.4960	0.4657	0.4444

Table 2. *Cont.*

Dataset	Method	Metric	In Distribution			Out of Distribution				
			×2	×3	×4	×6	×8	×12	×18	×24	×30
Manga109	bicubic	PSNR	30.82	26.95	24.90	22.69	21.45	19.98	18.76	17.99	17.46
		SSIM	0.9339	0.8556	0.7866	0.6958	0.6460	0.5977	0.5722	0.5624	0.5571
	Dense-LIIF	PSNR	38.12	32.93	29.96	26.22	24.14	21.77	19.97	18.93	18.25
		SSIM	0.9757	0.9407	0.9018	0.8242	0.7638	0.6830	0.6209	0.5909	0.5744
	EDSR(b)-LIIF	PSNR	38.67	33.53	30.58	26.77	24.57	22.04	20.14	19.06	18.34
		SSIM	0.9770	0.9450	0.9096	0.8380	0.7791	0.6954	0.6287	0.5954	0.5768
	RDN-LIIF	PSNR	39.26	34.21	31.20	27.33	25.04	22.36	20.35	19.20	18.44
		SSIM	0.9781	0.9487	0.9170	0.8508	0.7948	0.7099	0.6386	0.6014	0.5806
	RDST-t	PSNR	39.06	33.99	31.00	27.10	24.86	22.35	20.46	19.37	18.44
		SSIM	0.9779	0.9475	0.9151	0.8463	0.7894	0.7084	0.6412	0.6057	0.5810
	RDST-s	PSNR	39.33	34.32	31.33	27.44	25.16	22.57	20.61	19.49	18.53
		SSIM	0.9784	0.9496	0.9189	0.8532	0.7980	0.7165	0.6470	0.6100	0.5837
	RDST-b	PSNR	39.39	34.42	31.45	27.53	25.23	22.62	20.65	19.51	18.55
		SSIM	0.9785	0.9500	0.9198	0.8545	0.7997	0.7188	0.6490	0.6110	0.5844

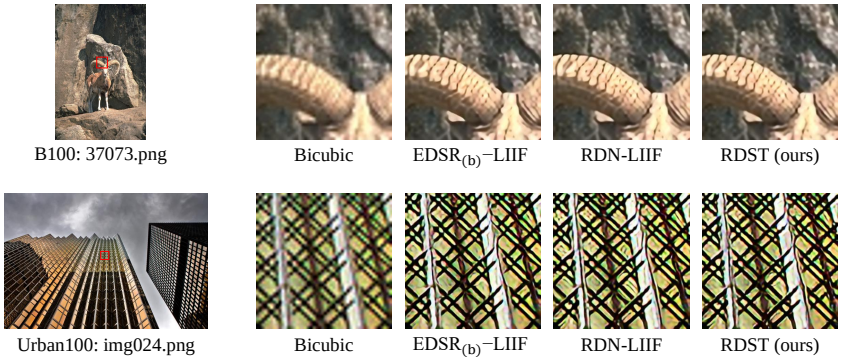

Figure 3. Visual results with a scale factor of 6.

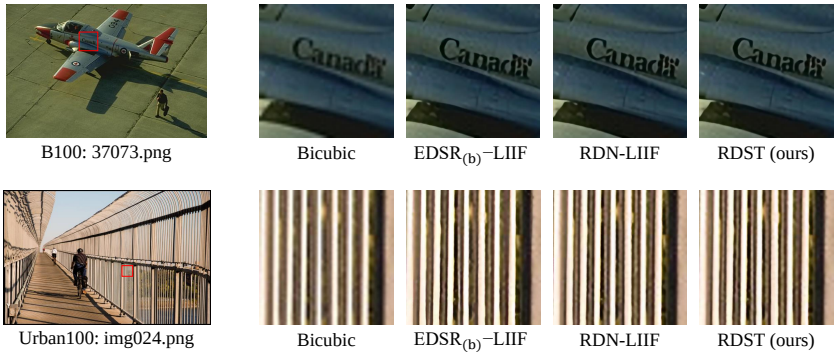

Figure 4. Visual results with a scale factor of 18.

Figure 5. Visual results with a scale factor of 18.

4.4. Ablation Experiment and Discussion

4.4.1. Impact of LFF and GFF

Table 3 shows the impact of LFF and GFF on the performance of the model. The four models in the table have the same RDTB number (6), STL number (6), window size (8), channel number (64), and attention head number (8), and the models were all tested on Manga109. It can be found from the PSNR indicators in the table that the addition of LFF and GFF enhances the flow of information before and after the network, improves the performance of the model, and verifies the effectiveness of LFF and GFF.

Table 3. Add Ablation study of LFF and GFF. Best performance are in red color, respectively.

LFF	GFF	Metric	×2	×3	×4	×6	×8	×12	×18	×24	×30
×	×	PSNR	39.32	34.26	31.24	27.36	25.09	22.51	20.58	19.47	18.51
		SSIM	0.9785	0.9491	0.9179	0.8515	0.7957	0.7141	0.6453	0.6088	0.5827
✓	×	PSNR	39.38	34.36	31.35	27.44	25.14	22.54	20.59	19.46	18.5
		SSIM	0.9785	0.9497	0.919	0.8535	0.7982	0.7166	0.6471	0.6094	0.5826
×	✓	PSNR	39.26	34.22	31.18	27.34	25.05	22.48	20.55	19.45	18.49
		SSIM	0.9783	0.9488	0.9171	0.8507	0.7949	0.7138	0.6452	0.6087	0.5827
✓	✓	PSNR	39.33	34.32	31.33	27.44	25.16	22.57	20.61	19.49	18.53
		SSIM	0.9784	0.9496	0.9189	0.8532	0.7980	0.7165	0.6470	0.6100	0.5837

It is worth noting that we also found a very interesting phenomenon. For the results within the distribution, the model that only adds LFF obtained the best effect at each magnification; for the results out of distribution, the model that combines LFF and GFF obtained the best effect at each magnification. We guess that this is because for a super-resolution image of multiples within the distribution, more attention is paid to the high-level features of the network, and LFF can provide enough local semantic information to reconstruct the image. For a super-resolution image for multiples outside the distribution, each level of the network needs to complement each other to achieve a better reconstruction effect.

4.4.2. Imapact of Head Number

Table 4 shows the influence of the number of multihead attentions in the Transformer structure on the performance of the model, and the models were all tested on Manga109. In order to more intuitively compare the impact of different numbers of attention heads on RDST. We also drew a scatter line chart for the PSNR indicators of the three zoom scales within and outside the distribution, as shown in Figure 3. For the convenience of presentation, we denote the models as RDST1, RDST2, RDST4, and RDST8.

Combining the data in Table 4 with the broken line in Figure 6, we can clearly see that for the over-score effect within the distribution, RDST8 obtains the best PSNR value; for the over-score effect outside the distribution, RDST2 obtains the best PSNR value. Through a large number of previous studies, it is known that different numbers of attention heads in the same layer of Transformer can learn information in different subspaces, but the attention patterns of most heads are the same. Therefore, we guess that for the super-resolution tasks in the subdivisions, the scaling factor is small, different feature information can be obtained from different heads, and feature information can be supplemented from heads with similar attention patterns, thereby improving the performance of the model. When the model performs an out-of-distribution super-resolution task, the scaling factor is large, and the heads with the same attention mode cannot achieve good information complementation. On the contrary, unnecessary information similar to noise is introduced, resulting in a large number of heads, causing the model's performance to decline.

Table 4. Ablation study of head number. Best performance are in red color, respectively.

Heads	×2	×3	×4	×6	×8	×12	×18	×24	×30
1	38.92	33.87	30.94	27.12	24.9	22.38	20.47	19.38	18.45
2	39.00	33.94	31.00	27.15	24.93	22.4	20.50	19.41	18.47
4	38.99	33.92	30.95	27.09	24.87	22.34	20.45	19.37	18.44
8	39.06	33.99	31.00	27.10	24.86	22.35	20.46	19.37	18.44

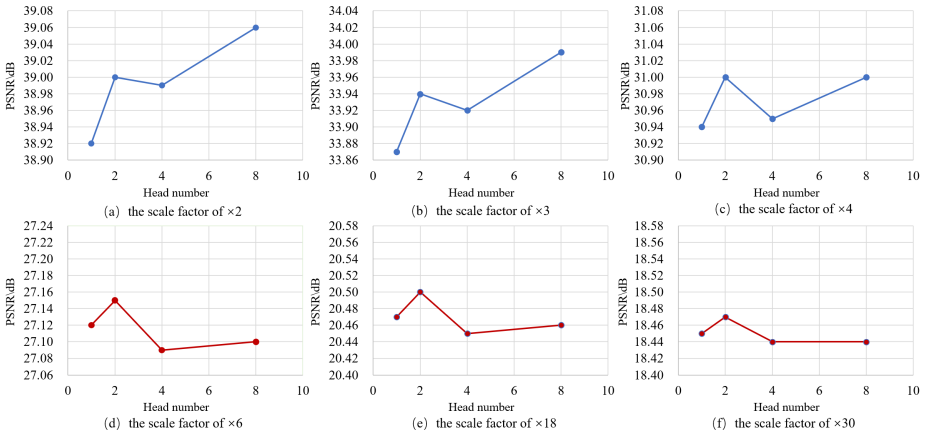

Figure 6. PSNR for different numbers of attention heads.

5. Conclusions

This paper proposed a Transformer super-division model RDST that can perform continuous-scale super-resolution tasks and has excellent performance. Based on Transformer, we introduced dense connection and local residual learning, and we designed RDTB with better feature extraction capabilities. Through multilevel feature fusion, we make full use of the information of each layer of the model, and then LIIF continuously expresses the fused features to obtain continuous-scale super-score results. The proposed RDST was tested on multiple benchmarks and achieved performance close to or even better than SOTA methods in fixed multiples within the segment, especially for arbitrary multiples outside of the distribution, producing considerable improvements compared to the other methods. In general, the overall performance of RDST is better than that of state-of-the-art SR methods.

Author Contributions: All authors contributed equally. G.Y. conducted the experiments and drafted the manuscript. J.L. and Z.G. implemented the core algorithm and performed the statistical analysis. C.Y. designed the methodology. Y.G. modified the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 42071322, and the National Key Research and Development Program of China under Grant, 2022YFB3903501.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Huang, Y.; Shao, L.; Frangi, A.F. Simultaneous Super-Resolution and Cross-Modality Synthesis of 3D Medical Images Using Weakly-Supervised Joint Convolutional Sparse Coding. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5787–5796. [CrossRef]
2. Mahapatra, D.; Bozorgtabar, B.; Garnavi, R. Image super-resolution using progressive generative adversarial networks for medical image analysis. *Comput. Med Imaging Graph.* **2019**, *71*, 30–39. [CrossRef] [PubMed]
3. Zhang, H.; Yang, Z.; Zhang, L.; Shen, H. Super-Resolution Reconstruction for Multi-Angle Remote Sensing Images Considering Resolution Differences. *Remote Sens.* **2014**, *6*, 637–657. [CrossRef]
4. Dong, X.; Wang, L.; Sun, X.; Jia, X.; Gao, L.; Zhang, B. Remote Sensing Image Super-Resolution Using Second-Order Multi-Scale Networks. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *59*, 3473–3485. [CrossRef]
5. Liu, W.; Lin, D.; Tang, X. Hallucinating faces: TensorPatch super-resolution and coupled residue compensation. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 2, pp. 478–484. [CrossRef]
6. Wang, Y.; Fevig, R.; Schultz, R.R. Super-resolution mosaicking of UAV surveillance video. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 345–348. [CrossRef]
7. Yang, J.; Wright, J.; Huang, T.; Ma, Y. Image super-resolution as sparse representation of raw image patches. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008; pp. 1–8. [CrossRef]
8. Gao, X.; Zhang, K.; Tao, D.; Li, X. Image Super-Resolution With Sparse Neighbor Embedding. *IEEE Trans. Image Process.* **2012**, *21*, 3194–3205. [CrossRef] [PubMed]
9. Glasner, D.; Bagon, S.; Irani, M. Super-resolution from a single image. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 349–356. [CrossRef]
10. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5206. [CrossRef]
11. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [CrossRef] [PubMed]
12. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654. [CrossRef]
13. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-Recursive Convolutional Network for Image Super-Resolution. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645. [CrossRef]
14. Tong, T.; Li, G.; Liu, X.; Gao, Q. Image Super-Resolution Using Dense Skip Connections. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4809–4817. [CrossRef]
15. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual Dense Network for Image Super-Resolution. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2472–2481. [CrossRef]
16. Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K.M. Enhanced Deep Residual Networks for Single Image Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1132–1140. [CrossRef]
17. Hu, X.; Mu, H.; Zhang, X.; Wang, Z.; Tan, T.; Sun, J. Meta-SR: A Magnification-Arbitrary Network for Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1575–1584. [CrossRef]
18. Chen, Y.; Liu, S.; Wang, X. Learning Continuous Image Representation with Local Implicit Image Function. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8624–8634. [CrossRef]
19. Vaswani, A.; Ramachandran, P.; Srinivas, A.; Parmar, N.; Hechtman, B.; Shlens, J. Scaling Local Self-Attention for Parameter Efficient Visual Backbones. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 12889–12899. [CrossRef]
20. 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 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, 3–7 May 2021.
21. 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 (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
22. Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; Gao, W. Pre-Trained Image Processing Transformer. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 12294–12305. [CrossRef]
23. Lu, Z.; Liu, H.; Li, J.; Zhang, L. Efficient Transformer for Single Image Super-Resolution. *arXiv* **2021**, arXiv:2108.11084.

24. Yang, F.; Yang, H.; Fu, J.; Lu, H.; Guo, B. Learning Texture Transformer Network for Image Super-Resolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5790–5799. [CrossRef]
25. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1833–1844. [CrossRef]
26. Dong, C.; Loy, C.C.; Tang, X. Accelerating the Super-Resolution Convolutional Neural Network. In Proceedings of the Computer Vision—ECCV 2016, Cham, Switzerland, 11–14 October 2016; pp. 391–407.
27. Tai, Y.; Yang, J.; Liu, X. Image Super-Resolution via Deep Recursive Residual Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2790–2798. [CrossRef]
28. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
29. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Proceedings of the Computer Vision—ECCV 2018, Cham, Switzerland, 8–14 September 2018; pp. 294–310.
30. Li, J.; Fang, F.; Mei, K.; Zhang, G. Multi-scale Residual Network for Image Super-Resolution. In Proceedings of the ECCV 2018, Cham, Switzerland, 8–14 September 2018.
31. Radford, A.; Narasimhan, K. Improving Language Understanding by Generative Pre-Training. 2018, *in press.* Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 1 April 2024).
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
33. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; J'egou, H. Training data-efficient image Transformers & distillation through attention. In Proceedings of the ICML, 2021, Virtual, 18–24 July 2021.
34. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1122–1131. [CrossRef]
35. Bevilacqua, M.; Roumy, A.; Guillemot, C.; line Alberi Morel, M. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In Proceedings of the British Machine Vision Conference, Surrey, UK, 3–7 September 2012; pp. 135.1–135.10. [CrossRef]
36. Zeyde, R.; Elad, M.; Protter, M. On Single Image Scale-Up Using Sparse-Representations. In *Curves and Surfaces*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 711–730.
37. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada; 7–14 July 2001; Volume 2, pp. 416–423. [CrossRef]
38. Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Sketch-based manga retrieval using manga109 dataset. *Multimed. Tools Appl.* **2016**, *76*, 21811–21838. [CrossRef]
39. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]

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.