Computers & Graphics (2023)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Image super-resolution with multi-scale fractal residual attention network

Author 1[a,*]

[a]*Institution 1*
[b]*Institution 2*

ARTICLE INFO

Article history:

Keywords:
Super-resolution
Multi-scale
Enhanced channel attention
Multi-path learning

ABSTRACT

Deep neural networks can significantly improve the quality of super-resolution. However, previous work has made insufficient use of low-resolution scale features and channel-wise information, hence hindering the representational ability of CNNs. To address these issues, a multi-scale fractal residual attention network (MFRAN) is proposed. Specifically, MFRAN consists of fractal residual blocks (FRBs), dual-enhanced channel attention (DECA), and dilated residual attention blocks (DRAB). Among them, FRB applies multi-scale extension rule to continuously expand into a fractal structure that detects multi-scale features; DRAB constructs a combined dilated convolution to learn a generalizable and expressive feature space with a larger receptive field; DECA employs one-dimensional convolution to achieve cross-channel information interaction, and enhance the flow of information between groups by channel shuffling. Then, we integrate horizontal feature representations via local residual and feature fusion. Extensive quantitative and qualitative evaluations of benchmark datasets show that our proposed approach outperforms state-of-the-art methods in terms of quantitative metrics and visual results.

1. Introduction

Single image super-resolution (SISR), which aims to reconstruct a high-resolution (HR) image from a single low-resolution (LR) image, has gained increasing attention for decades in computer vision. In order to generate SR images with higher quality and clearer texture, a large number of SR methods have been proposed, including interpolation-based [1], reconstruction-based [2, 3], learning-based methods [4, 5, 6], nearest neighbor embedding [7], sparse coding representation [8, 9], and those based on deep learning [10].

In recent years, convolutional neural networks (CNNs) have higher image quality and are less restricted by application scenarios, making CNN-based SR models the most mainstream super-resolution reconstruction method at present. Dong et al. (SRCNN) [11] first introduced a convolutional neural network into SISR, which laid a foundation for a large number of network models later; FSRCNN [12] and ESPCN [13] directly used low-resolution images as input and used transposed convolutional layers and sub-pixel convolutional layers for reconstruction, respectively; VDSR [14] verified the effect of network depth and residual learning on image reconstruction; Lim et al. (EDSR) [15] optimized the structure of SRResNet [16], removed the BN layer and redundant relu layer; Zhang et al. designed the RDN [17] and RCAN [18], which constructed the dense residual structure and residual in residual with attention mechanism respectively; Yang et al. (DGRN) [19] proposed a double-gradient regression scheme to restore the structural information of objects in the image.

Existing convolutional neural network-based methods have achieved good results in SR tasks. From the available studies, it appears that building a deeper network model can effectively improve the quality of reconstruction, so researchers tend to use

*Corresponding author:
e-mail: --- (Author)

deeper convolutional neural networks to improve performance. As the depth of the network increases, parameters and flops of the model increases, which leads to more time-consuming and memory-intensive training of models, hence exploring a super-resolution method that balances performance and parameters is a research priority. To construct such models, we studied the structure of existing CNNs to get some ideas, which will be discussed in detail next.

1.1. Multi-scale Feature Extraction

Numerous studies have shown that exploiting the rich scale information in LR images is the key to reconstructing high-quality images. Li et al. [20] (MSRN) first introduced a multi-scale approach to the SR task, which used convolution kernels of different sizes, and increased the number of channels; Li et al. [21] proposed multi-scale dense cross block to enhance the information interaction between convolutions. Nevertheless, these methods have a limited receptive field of convolution kernels, while still failing to balance well the number of parameters and performance. Moreover, frequent adjustments to the number of channels will increase the cost of memory access, so exploring more effective multi-scale feature extraction modules is the focus of our research.

1.2. Attention

The results of existing methods demonstrate that the input features of LR contain rich low-frequency information, and treating these channels equally will hinder the expressiveness of the model. Therefore, we hope to use the attention to make the network pay more attention to the texture detail, and improve the problem of insufficient high-frequency information and texture detail in the results under large-scale factors. However, the attention proposed by Zhang et al. [18] and Liu et al. [22] has limited enhancement of the model, so we would like to propose a more effective attention to enhance the attention to key features, and generate more realistic detailed information.

1.3. Multi-scale & Feature Fusion

The combination of multi-path learning and feature fusion allows SR models to better extract image features from multiple scales. For example, Qin et al. [23] and Zhang et al. [24] have proposed a number of multi-path structure, and fused global features with hierarchical feature fusion structure (HFFS), yet these methods used only two paths resulting in limited extraction capability, and HFFS resulted a large computational burden for the end of the network. Therefore, we will explore the design of a module with more paths to enhance the network performance, and fuse them using a efficient approach that can reduce the computational effort to provide better modeling capabilities.

The contributions of our work are summarized as follows:

A Multi-Scale Fractal Residual Attention Network (MFRAN) is proposed for SISR, compared with the state-of-the-art, our method achieves better results with fewer parameters, and the detailed texture of the reconstructed image is more realistic and clearer.

We devise Dual-Enhanced Channel Attention (DECA), which can capture inter-channel dependencies more efficiently, and support cross-channel information interaction, enabling the model to reconstruct SR images with richer details under large-scale factors.

We design a Fractal Residual Block (FRB), which consists of dilated residual attention blocks with different size, and low-level features and high-level features are aggregated to provide richer information for reconstructing high-quality details.

2. Related work

2.1. CNN-based for SISR

SRCNN [11] was the first successful attempt to use CNNs for image super-resolution. Since then, efforts have been made to design better SR models to improve the reconstruction quality. Later, a series of SR methods were proposed to continuously increase the depth and width of the network (such as EDSR [15], RDN [17]), and satisfactory SR performance was achieved. Although these networks achieved SOTA performance, it was hard to balance model performance and flops.

2.2. Attention Mechanisms

The attention mechanism can effectively improve the quality of the reconstructed image, Dai et al. [25] proposed a second-order channel attention (SOCA) module, where SOCA adaptively scales the features of the channel approach; Zhang et al. [26] employed local and non-local attention blocks to capture long-term dependencies between pixels; Yang et al. [27] proposed a multi-scale grid attention module to refine horizontal features; Wang et al. [28] proposed a dense connection network based on attention to suppress redundant response; Liu et al. [29] used the enhanced attention module to adaptively enhance the high-frequency details.

2.3. Multi-scale & Mutil-path

Multi-scale feature extraction has been the focus of SR tasks, and multi-path learning is adopted to transfer and fuse features, and fuse them to provide better modeling capabilities finally.

Recently, Lv et al. [30] extracted different hierarchical features from multi-scale dense fusion blocks; Wang et al. [31] used multi-scale features to fuse details of images with different resolutions; Larsson et al. [32] proposed a new fractal structure, which made the gradient better back propagation in the training process; Feng et al. [33] designed multi-scale fractal residual block and multi-path structure, which can effectively extract multi-scale features.

Based on the above work, we construct a multi-scale fractal residual attention network for more efficient super-resolution, in which the fractal residual block based on the fractal structure is used for feature extraction, and we design a dilated residual attention block composed of dual-enhanced channel attention to further improve the network performance.

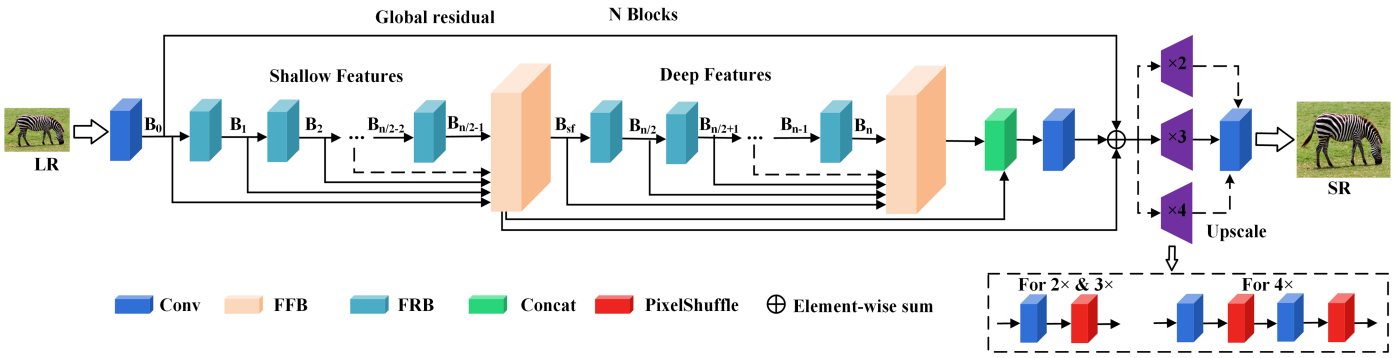

Fig. 1. The architecture of MFRAN, where the first $\frac{n}{2}$ FRBs are called shallow features and the last $\frac{n}{2}$ FRBs are called deep features, low-level spatial features (shallow features) and high-level semantic features (deep features) are fused separately at the end; B_n(n=0,1,...,N) indicates the nth FRB output feature.

3. Method

The purpose of SISR is to reconstruct the corresponding HR image I^{HR} from the LR image I^{LR}, which is usually obtained from the HR image by bicubic operation, and for the scale factor w can be formulated as

$$I^{LR} = R\left(I^{HR}, w\right) \tag{1}$$

where R records the degradation process, we aim to design a super-resolution network to recover the predicted image I^{SR} close to the HR image I^{HR} from the LR image I^{LR} with arbitrary scale factor w.

The common SR loss functions are the L_1 function, L_2 function, MSE function, etc., we choose the L_1 function as the loss function due to its greater convergence [34], which is described as follows

$$L_1 = \frac{1}{N}\sum_{i=1}^{N}\left\|I_i^{HR} - I_i^{SR}\right\|_1 \tag{2}$$

where $\|\cdot\|_1$ indicates L_1 norm, N.

3.1. Network Architecture

The MFRAN architecture is shown in Fig. 1. In the beginning, we apply a standard convolution for feature extraction of LR images formulated as

$$B_0 = Conv\left(I^{LR}\right) \tag{3}$$

where B_0 denotes the extracted features, which will also be used as the initial feature input for the model. We set up a total of N cascaded FRBs for feature extraction, n denotes the nth FRB(n=1,...,N), and we divide the features into two parts, we call the first $\frac{N}{2}$ FRBs shallow features and the last $\frac{N}{2}$ FRBs deep features. The shallow features B_{sf} is denoted as

$$B_{sf} = FFB\left(\left[B_0, B_1, \ldots B_{\frac{n}{2}-1}\right]\right) \tag{4}$$

the deep feature B_{df} is denoted as

$$B_{df} = FFB\left(\left[B_{sf}, B_{\frac{n}{2}}, B_{\frac{n}{2}+1}, \ldots B_n\right]\right) \tag{5}$$

where FFB(·) represents feature fusion block. In the process of feature transfer, the output of each FRB contains different feature representations, so it is crucial how each FRB extracts the different levels of LR features and transfers them to the end for reconstruction.

The shallow network has a relatively small receptive field and a strong ability to represent spatial detail information; the deep networks have a relatively large receptive field and the features contain more edge, and texture details, so we fuse them separately, and the processed feature B_{sf} is used as the input of the deep network. Then, we aggregate the shallow features and deep features at the end, so that the interaction between spatial and semantic information is increased, and then the SR images generated by the model are richer, the global feature B_{out} can be expressed as

$$B_{out} = FFB\left(B_{sf}, B_{df}\right) \tag{6}$$

The final features are reconstructed by dynamic upscaling module, when the scale factor is ×2 or ×3, we perform one upscaling; when the scale factor is ×4, we apply two upscaling, dynamic means that the upscaling blocks with different reconstruction factors are arranged in parallel, and the features flow into the block corresponding to during training. For other arbitrary-magnification such as ×3.7, ×7, the Meta-SR [35] can be used. The final reconstructed SR image can be expressed as

$$I^{SR} = Upscale\left(B_{out}\right) \tag{7}$$

3.2. Fractal Residual Block (FRB)

SR is a pixel-level regression task, previous work has tended to build deep network architectures on a single path, the network requires an effective multi-scale feature representation to accurately predict detailed information. Moreover, the width of the network is as important as the depth of the network, and neural networks with rectified linear unit activation functions need to be wide enough to maintain general approximation properties as the depth increases, so we shifted the focus from the previous deeper and narrower architectures to deeper and wider architectures.

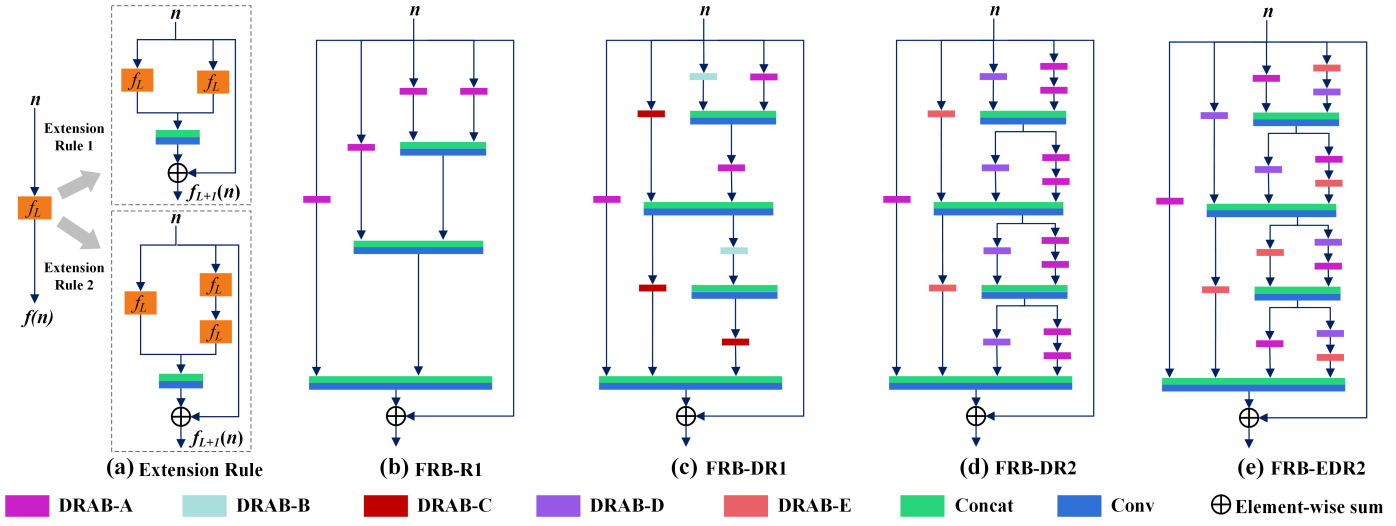

Fig. 2. (a) is the multi-path expansion rule; R1, DR1, DR2, EDR2 denote using rule 1 and standard convolution, using rule 1 and dilated convolution, using rule 2 and dilated convolution, using rule 2 and enhanced dilated convolution respectively, the suffix ABCDE of DRAB denotes a combination of dilated convolution at different rates.

From this perspective, we propose two multi-path extension rules. Firstly, let $f_L(\cdot)$ represents the DRABs for different combinations of scales, expressed as

$$f_L(n) = DRAB - \varphi \tag{8}$$

where φ indicates the type of DRAB and n denotes the input of the nth FRB, we have set a total of five combinations of DRAB-A to DRAB-E, and we construct a multi-path fractal structure with $f_L(\cdot)$ as the base block. Specifically, in extension rule 1 (R1), we add an $f_L(\cdot)$ to each path, and then the features extracted on the two paths are fused, as shown in Fig. 2 (a). Secondly, we construct FRB-R1 and FRB-DR1 based on R1, each FRB has four paths, with the difference that a standard convolution is used in FRB-R1, while FRB-DR1 uses the dilated convolution and an additional $f_L(\cdot)$ operation is performed after the features are fused, the structure is shown in Fig. 2 (b) and Fig. 2 (c).

In extension rule 2 (R2), we add a path to the original path of $f_L(\cdot)$, and the number of $f_L(\cdot)$ on the new path is doubled to extract finer features, we define successive fractals recursively

$$f_{L+1}(n) = \left[\left(f_L \circ f_L\right)(n)\right] \oplus \left[f_L(n)\right] \tag{9}$$

where $\circ$ indicates two consecutive $f_L(\cdot)$ operations, $\oplus$ means feature fusion operation, which is shown at the bottom of Fig. 2 (a). Similarly, we build FRB-DR2 and FRB-EDR2 based on R2, each path adopts the same type of $f_L(\cdot)$ in FRB-DR2, while the type of $f_L(\cdot)$ on each path is different in FRB-EDR2, and we use FRB-EDR2 as our final fractal residual block, as shown in Fig. 2 (d) and Fig. 2 (e). In general, FRB constructs a parallel multi-path structure that enables the simultaneous utilization of images at different scales, we collapse the adjacent connections into a single column spanning multiple columns, and the concatenation operation is used to combine all input features into a single output feature, and we add global residuals connections on FRBs to help training finally, which can be expressed as

$$B_{n+1} = B_n + L_{FRB}(n) \tag{10}$$

where n represents the nth FRB. Furthermore, we input low-resolution images into $f_L(\cdot)$ at different scales through mutil-path independently for feature extraction, and aggregate the generated local features to achieve coarse and fine horizontal feature representations.

From the structure, the depth of the network is defined as the number of layers of the longest path between the inputs and outputs, the number of $f_L(\cdot)$ on the first path through the fourth path is 1,2,4,8, so scales as 2^{L-1}, resulting in a total depth $t{\cdot}m{\cdot}2^{L-1}$, where t denotes the number of $f_L(\cdot)$, and m is the number of conv layers of DRAB. As for the feature fusion method, we tried both concatenation and element-wise sum operations for feature fusion between two paths, exploited concatenation for feature fusion finally.

3.3. Dilated Residual Attention Block (DRAB)

To obtain efficient multi-scale feature representations, we propose DRAB, which is the basic block of the whole FRB structure, containing two convolutional layers, and an activation layer. Next, we will describe its specific architecture and the proposed dual-enhanced channel attention mechanism.

3.3.1. Architecture

In previous work, both MSRN [20] and MDCN [21] applied a convolution kernel of size 5×5 to expand the receptive field, MSRN doubled the number of features in the second stage, and MDCN used 128 channels at the beginning, 384 channels were fused in the first stage, and 576 channels were fused in the second stage, which limited the receptive field and increased the computational burden. To prevent the local information loss caused by the gridding effect [36] and the lack of correlation of information acquired at a distance, we design a stepped structure. Let the first kernel be x and the second kernel is y, x and

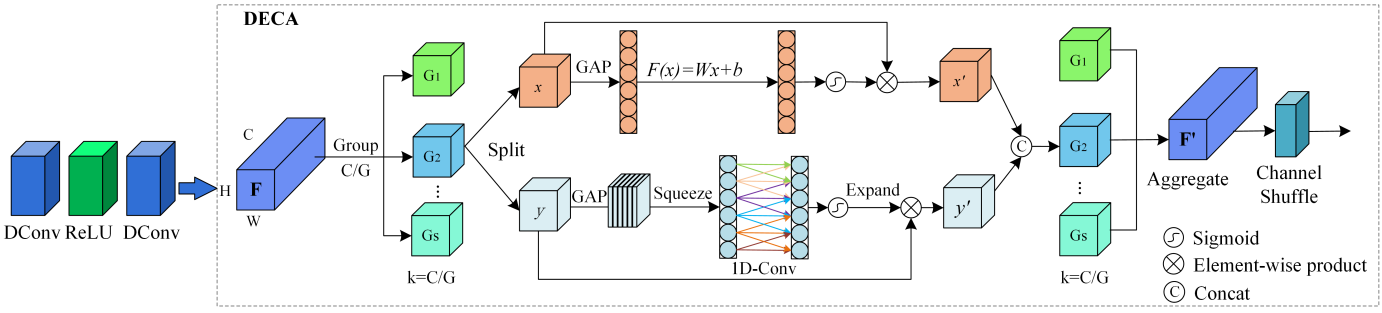

Fig. 3. Dilated Residual Attention Block structure, DConv represents the dilated convolution, the dashed box indicates the structure of dual-enhanced channel attention (DECA), where F is the input of DECA, which is divided into G (G is set to 8) groups by Split, and the number of channels in each group is $k(k=\frac{C}{G})$, GAP represents global average pooling, 1D-Conv represents one-dimensional convolution, and finally the G groups of features are integrated to obtain F' for output after channel shuffle.

y are combined using different dilated rates r (r=1,2,3). Denote the combination by φ

$$\varphi \in \{(x, y) \mid x \in \{1, 2, 3\}, y \in \{1, 2, 3\}\} \tag{11}$$

where (x, y) denotes the combination of different dilated rates, x denotes the dilated rate of the first kernel and y denotes the dilated rate of the second kernel, and we add a relu activation layer between x and y, which we call φ=(1,1) DRAB-A, φ=(2,2) DRAB-B, φ=(3,3) DRAB-C, φ=(1,2) DRAB-D, φ=(1,3) DRAB-E. Here, we avoid using the combination of dilated rate 2 and 3, because the experimental result proves that the combination of 2 and 3 would lead to local information loss, and the final reconstruction result is poor, hence we only used the above five combinations. Meanwhile, we use a local residual connection to sum the input and output of DRAB to further reduce the loss of local information. In this way, the DRAB can obtain information from a wider range of pixels avoiding the gridding problem, and the DRAB receptive field ranges on the four paths of FRB are different, thus allowing for the extraction of richer hierarchical features.

3.3.2. Dual-enhanced channel attention (DECA)

Feature representation plays an important role in model performance, while channel-wise attention helps the network to filter out better feature representation among the features, which facilitates high-quality image reconstruction work. Most of the attention mechanisms applied in the past are channel attention and spatial attention, while SR is a pixel-level task where each region in the feature mapping has the same importance, we tend to improve the model performance in terms of channel-wise. However, the channel attention used in RCAN will change the number of channels leading to MAC increase, which is independent of modeling channel dependencies at the same time. Consequently, we propose a dual-enhanced channel attention that both models channel dependencies and captures inter-channel information interactions across channels. The architecture of the DECA module is shown in the dashed box in Fig. 3., DECA is divided into four main steps as follows

Group. Given a feature mapping $F \in \mathbb{R}^{H\times W\times C}$ where H, W and C denote height, width, and channel, respectively, the input F is devided into G groups along the channel dimension, and the number of channels in each group is $k(k=\frac{C}{G})$, i.e., $F=[F_1,...,F_G]$, $F_i \in \mathbb{R}^{H\times W\times \frac{C}{G}}$, where each sub-feature G_i gradually captures specific features during the training process. Then, the corresponding importance coefficients are generated for each sub-feature by the attention module. Specifically, at the beginning of each attention module, the input of G_i is divided into two branches along the channel dimension, i.e., $x, y \in \mathbb{R}^{H\times W\times \frac{C}{2G}}$, one branch generates the channel attention graph by exploiting the interrelationships between channels, while the other branch captures local information interactions by way of cross-channel.

Channel attention. Given a feature mapping $x \in \mathbb{R}^{H\times W\times \frac{C}{2G}}$ of the branch above, we first embed global information via a simple global average pool (GAP) to generate channel statistics as $x \in \mathbb{R}^{\frac{C}{2G}\times 1\times 1}$, which can be calculated by shrinking x through spatial dimension $H \times W$:

$$x = f_{GAP}(x) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x(i, j) \tag{12}$$

Next, we employ a gating mechanism instead of 1×1 convolution to guide the adaptive selection of channels, thus avoiding changing the number of channels and better balancing the speed and accuracy of the model a gating mechanism guides the adaptive selection of channels. Subsequently, we apply the sigmoid layer to normalize the generated attention weight matrix and do multiplication with the initial feature mapping x. Finally, the final output of the channel attention x' can be obtained by

$$x' = \sigma\left(F(x)\right) \cdot x = \sigma(Wx + b) \cdot x \tag{13}$$

where σ means sigmoid, $W \in \mathbb{R}^{\frac{C}{2G}\times 1\times 1}$, $b \in \mathbb{R}^{\frac{C}{2G}\times 1\times 1}$.

Cross-Channel Attention. For the branch $y \in \mathbb{R}^{H\times W\times \frac{C}{2G}}$ located below, the global features are aggregated by GAP and generate a 1×1×C feature representation, and then convert the two-dimensional features into the one-dimensional feature representation of 1×C by squeeze, where the squeeze does not change the number of channels; after that, we design a

one-dimensional convolution to generate weights for each feature channel; compared with two-dimensional (2D) convolution, one-dimensional (1D) convolution enables multiple adjacent channels to participate in the adaptive channel selection process, enabling local cross-channel information interaction, while avoiding the effect of channel reduction on attention and reducing the MAC. After that, we obtain normalized weights by the sigmoid layer, recovering 1×C to 2D feature representation 1×1×C by expand, we finally weight the normalized attention weights into the features of each channel by the element product.

Channel shuffle. To efficiently train the deep neural network, we divide the features F into G groups, and each attention is applied only to the corresponding channel group, we aggregate the G group features at the end, and restore the number of channels to C after aggregation. Meanwhile, to make the connection between the different channels, we reshape C into (G, C) and disorder it, then transpose it into (C, G), and finally reshape it into C to realize the flow of information across groups along the channel dimension and increase the information representation capability.

4. Experiment

4.1. Datasets and Metrics

DIV2K [37], a publicly available high-quality (2K resolution) benchmark dataset is used for training. which is the widely used dataset for image restoration tasks in recent years, including 800 for training, 100 pictures for verification, and 100 pictures for testing. Then, five standard benchmark datasets Set5 [38], Set14 [39], B100 [40], Urban100 [41] and Manga109 [42] are used for testing. Among them, Set5, Set14, and B100 consist of natural images, Urban100 contains challenging urban scene images with details in different frequency bands, and Manga109 includes the cover of 109 Japanese comics. Later, downscale HR images with the desired scaling factor (×2, ×3, ×4) using bicubic interpolation, and use the downsampled image to simulate the low-resolution image. In order to make a fair comparison, we evaluated the SR results in terms of the peak signal-to-noise ratio (PSNR) and structural similarity image measurement (SSIM) [43] on the Y channel(luminance) in the YCbCr image space, because the visual system of the human is more sensitive to details in intensity than in color.

4.2. Implementation Details

Model. Now, we specify the implementation details of our proposed MFRAN. In the external cascade structure, the FRB-EDR2 number is set to 16, the size of all conv layers (including 1D-Conv) are set to 3×3, except for the fusion module, whose kernel size is 1×1, the zero-padding strategy is used to keep the size fixed, and the group number G in DECA is set to 8. Conv layers in shallow feature extraction and deep feature extraction have C = 64 filters, and the mean shift method is employed, while MFRAN can also process gray images. Furthermore, self-ensemble is used to further improve performance, which is denoted as MFRAN+, self-ensemble is a data augmentation method, which rotates and horizontally flips LR image at different angles (0°, 90°, 180°, 270°) to obtain a set of 8 images, then input these images into the model, perform corresponding inverse transformation of reconstructed HR images to obtain outputs, and finally obtain the final results by taking the average of these outputs.

Training. In training, we use the RGB input patches of size 48×48 from the LR image with the corresponding HR patches, and we randomly flip the patch horizontally and rotate it 90 degrees for data augmentation. We train the MFRAN model with ADAM [48] optimizer by setting β_1=0.9, β_2=0.999, and ε=10^{-8}. We set the mini-batch size as 16, the initial learning rate is set to 2×10^{-4} and the learning rate decay factor of step decay is set to 0.5, which decreases to half every 2×10^{-5} iterations. The models are implemented by PyTorch on the machine with 3.5GHz intel i9-11900K, 64G RAM, and Nvidia RTX 3090 GPU (24G memory).

4.3. Comparisons with the state-of-the-arts

In order to prove the effectiveness of the proposed MFRAN model, we have made a comprehensive comparison between our model and other 12 state-of-the-art SR methods from the perspective of quantitative evaluation, including Bicubic, EDSR [15], SRMDNF [44], MSRN [20], RDN [17], DGRN [19], CARN [45], MSFFRN [23], SeaNet [46], MDCN [21], TSAN [47] and MSFRN [33].

Quantitative comparison. In Table 1, we performed a quantitative comparison with some state-of-the-art methods on five benchmark datasets Set5, Set14, BSD100, Urban100, and Manga109, and we calculated the average value achieved by each method on the five datasets. The bold font indicates the best results and the underlined font indicates the second-best results. As can be seen in Table 1, MFRAN achieves outstanding results with all three scaling factors, and MFRAN is also higher than other methods in average PSNR/SSIM, which shows that MFRAN has better and more competitive super-resolution reconstruction results.

Visual Comparison. We perform the visual comparisons at scaling factors of ×2, ×3, and ×4, respectively, where the SR images of RDN and RCAN are generated by us based on the pre-trained models provided by the authors, and the SR images of the other methods are derived from those provided in the open-source repositories of the corresponding authors.

As shown in Fig. 4., we performed visual comparisons on small scaling factors (×2, ×3). When the scaling factor is ×2, the visual results are quite close on Set5, but MFRAN achieves the highest PSNR/SSIM; when the scaling factor is ×3, there is a large difference in the detailed comparison of img092 from Urban100. The SR image texture direction of CARN, MSRN, and MSFFRN is opposite to the real image, and the local details are blurred. Meanwhile, the details of the upper right corner of the SR images of EDSR and RCAN are blurred, and the lines cross in the lower-left corner of MDCN, while the texture of

Table 1
Quantitative comparisons with state-of-the-art methods. PSNR/SSIM at ×2, ×3, and ×4 scale factors and averages over all data sets are reported. Bolded fonts indicate the best results and underlined fonts indicate the second-best results

Methods	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM	Average PSNR/SSIM
Bicubic	×2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339	30.22/0.8832
EDSR [15]	×2	38.11/0.9601	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773	35.27/0.9387
SRMDNF [44]	×2	37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9761	34.51/0.9342
MSRN [20]	×2	38.08/0.9605	33.74/0.9170	32.23/0.9013	32.22/0.9326	38.82/0.9868	35.02/0.9396
RDN [17]	×2	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	39.18/0.9780	35.33/0.9395
CARN [45]	×2	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	N/A	N/A
MSFFRN [23]	×2	38.15/0.9610	33.88/0.9195	32.29/0.9010	32.60/0.9326	N/A	N/A
SeaNet [46]	×2	38.08/0.9609	33.75/0.9190	32.27/0.9008	32.50/0.9318	38.76/0.9774	35.07/0.9380
MDCN [21]	×2	38.19/0.9612	33.86/0.9202	32.32/0.9014	32.92/0.9355	39.09/0.9780	35.28/0.9393
MSFRN[33]	×2	38.02/0.9606	33.68/0.9184	32.19/0.8998	32.17/0.9287	38.59/0.9770	34.93/0.9369
DGRN [19]	×2	38.11/0.9600	33.74/0.9160	32.24/0.9000	32.47/0.9300	38.94/0.9770	35.10/0.9366
TSAN [47]	×2	38.22/0.9613	33.84/0.9196	32.32/0.9015	32.77/0.9345	N/A	N/A
MFRAN(Ours)	×2	38.24/0.9615	34.07/0.9215	32.37/0.9021	33.16/0.9373	39.35/0.9779	35.44/0.9401
MFRAN+ (Ours)	×2	**38.29/0.9616**	**34.16/0.9221**	**32.41/0.9025**	**33.34/0.9384**	**39.52/0.9783**	**35.51/0.9406**
Bicubic	×3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556	27.31/0.7963
EDSR [15]	×3	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476	31.48/0.8793
SRMDNF [44]	×3	34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	34.13/0.9484	31.49/0.8799
MSRN [20]	×3	34.38/0.9262	30.34/0.8395	29.08/0.8041	28.08/0.8554	33.44/0.9427	31.18/0.8757
RDN [17]	×3	34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484	31.49/0.8799
CARN [45]	×3	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	N/A	N/A
MSFFRN [23]	×3	34.65/0.9292	30.53/0.8463	29.23/0.8086	28.65/0.8619	N/A	N/A
SeaNet [46]	×3	34.55.0.9282	30.42.0.8444	29.17.0.8071	28.50.0.8594	33.73.0.9463	31.27.0.8771
MDCN [21]	×3	34.69/0.9294	30.54/0.8470	29.26/0.8095	28.83/0.8662	34.17/0.9485	31.50/0.8801
MSFRN [33]	×3	34.40/0.9272	30.34/0.8423	29.10/0.8052	28.19/0.8530	33.59/0.9447	31.12/0.8745
TSAN [47]	×3	34.64/0.9282	30.52/0.8454	29.20/0.8080	28.55/0.8602	N/A	N/A
MFRAN (Ours)	×3	34.76/0.9299	30.60/0.8477	29.30/0.8099	28.95/0.8682	34.43/0.9494	31.61/0.8810
MFRAN+ (Ours)	×3	**34.85/0.9305**	**30.72/0.8490**	**29.35/0.8109**	**29.16/0.8711**	**34.70/0.9506**	**31.76/0.8824**
Bicubic	×4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866	25.68/0.7250
EDSR [15]	×4	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148	29.33/0.8289
SRMDNF [44]	×4	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024	28.71/0.8161
MSRN [20]	×4	32.07/0.8903	28.60/0.7751	27.52/0.7273	26.04/0.7896	30.17/0.9034	29.13/0.8257
RDN [17]	×4	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151	29.32/0.8292
CARN [45]	×4	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	N/A	N/A
MSFFRN [23]	×4	32.44/0.8978	28.76/0.7860	27.74/0.7400	26.47/0.7980	N/A	N/A
SeaNet [46]	×4	32.33/0.8970	28.72/0.7855	27.65/0.7388	26.32/0.7942	30.74/0.9129	29.13/0.8257
MDCN [21]	×4	32.48/0.8985	28.83/0.7879	27.74/0.7423	26.69/0.8049	31.10/0.9163	29.37/0.8300
MSFRN [33]	×4	32.16/0.8947	28.62/0.7823	27.57/0.7362	26.09/0.7868	30.47/0.9082	28.98/0.8216
DGRN [19]	×4	32.57/0.8980	28.81/0.7870	27.70/0.7410	26.58/0.8010	31.11/0.9160	29.35/0.8286
TSAN [47]	×4	32.40/0.8975	28.73/0.7847	27.67/0.7398	26.39/0.7955	N/A	N/A
MFRAN (Ours)	×4	32.62/0.9004	28.87/0.7890	27.76/0.7429	26.82/0.8081	31.30/0.9183	29.47/0.8317
MFRAN+ (Ours)	×4	**32.73/0.9014**	**28.99/0.7908**	**27.82/0.7444**	**27.02/0.8122**	**31.60/0.9209**	**29.63/0.8339**

MFRAN is the clearest, and the PSRN of MFRAN is 0.42dB higher than MDCN.

Then, we also perform the visual comparison on the large scaling factor (×4) as shown in Fig. 5. In the img024 from Urban100, the SR images of CARN, MSRN, and RCAN for the railing are blurred and incorrectly oriented, and the right-side lines of MDCN appear crossed. In the img095 from Urban100, the upper left corner area of the floor tiles of the SR images reconstructed by other methods are blurrier, with incorrect edges and other phenomena. By contrast, the SR image reconstructed by MFRAN has sharper and clearer edges. Specifically, MFRAN's SR image in img024 has more evenly spaced and correctly oriented bars, which is the closest to the ground truth, especially noting that MFRAN is 0.37dB higher than RCAN and 0.67dB higher than MDCN. At the same time, the texture of MFRAN is the clearest in the SR image of img095, and the contrast is most obvious in the upper left region, and we exclusively selected the img095 with a slightly lower PSNR than MDCN to prove that MFRAN can achieve better visual effects and high-quality SR image.

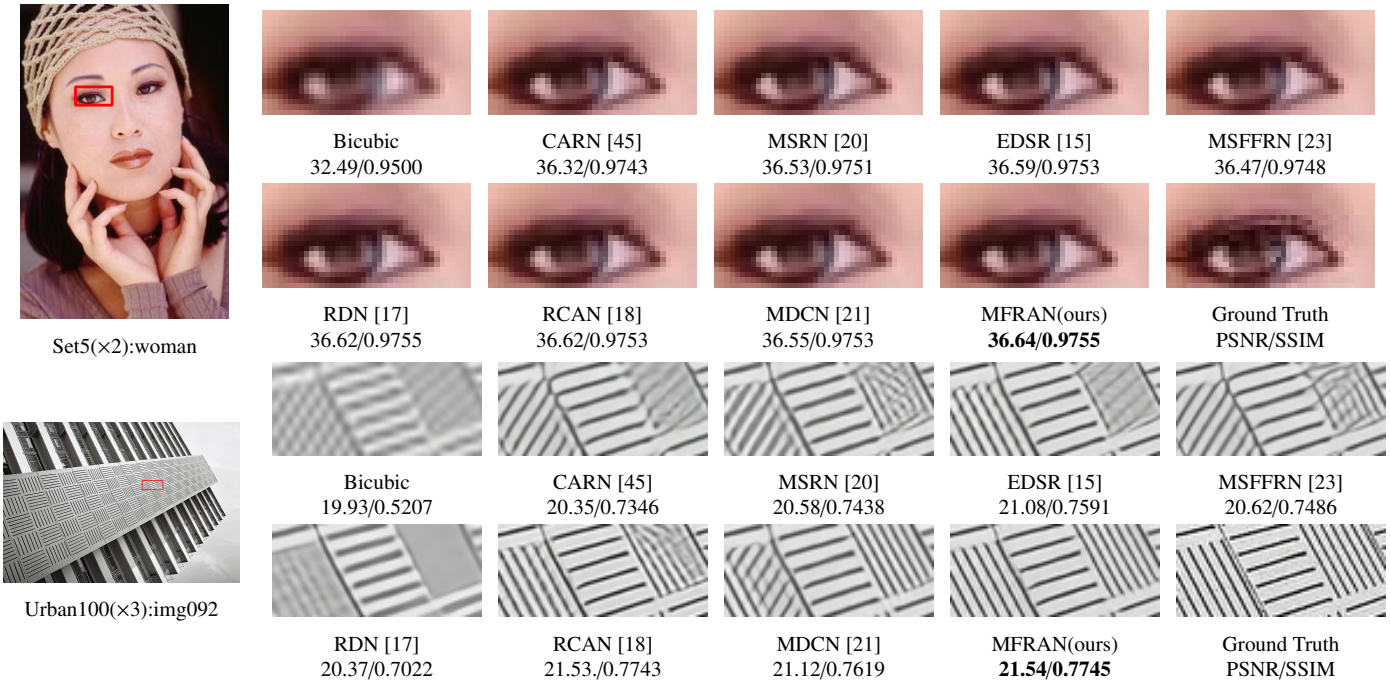

Fig. 4. Visual evaluation for a scale factor of ×2 & ×3 on the image "woman" from Set5 and the image "img092" from Urban100.

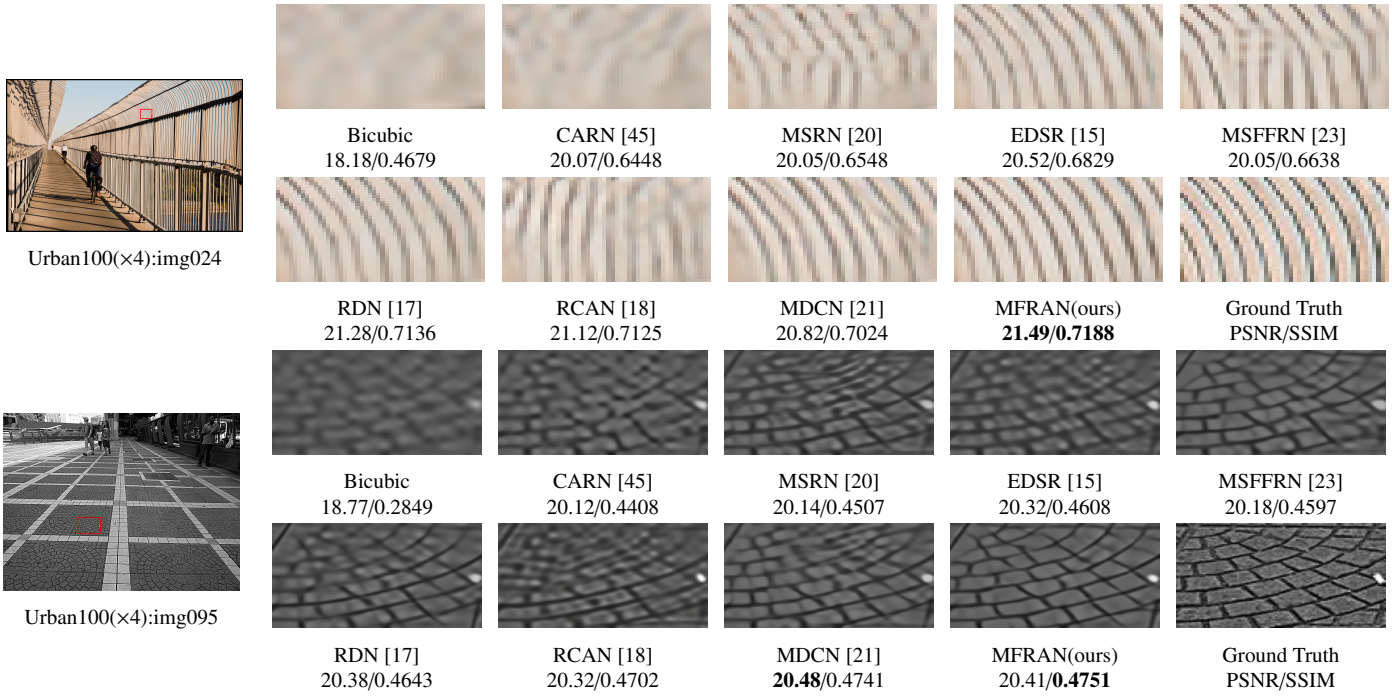

Fig. 5. Visual evaluation for a scale factor of ×4 on the image "img024" and the image "img095" from Urban100.

5. Model Analysis and discussion

5.1. Study of Fractal Residual Block

Fractal and feature fusion are significant components of the fractal residual block, hence we set up ten sets of experiments to verify the effects of different fractal structures and fusion methods on model performance. To ensure the accuracy and fairness of the experiments, all experiments are conducted in the same environment, and the fusion method adopts 1×1 convolution by default unless otherwise stated, and the components are arranged in order from top to bottom leftmost path, specifically: 1). R1-S: Using two standard convolutions as the basic blocks of FRB-R1 in Fig. 2 (b); 2). R1-D: Using dilated convolution with the same dilated rate in FRB-R1; 3). DR1-S: Applying two standard convolutions for the basic blocks in FRB-DR1 of Fig. 2 (c); 4). DR1-D: Adopting the same dilated rate of convolution in FRB-DR1; 5). R2-S: Employing two standard con-

Table 2
The comparison experiments of different fractal structures, fusion methods, and convolution methods, R1 denotes the experimental group with the R1 rule for expansion, R2 denotes the experimental group with R2 expansion, and bold font indicates the best results

Components		Different Combinations of Components and Rules									
Fractal Residual Block Components		R1-S	R1-D	DR1-S	DR1-D	R2-S	R2-D	EDR2 Mean	EDR2 Path=2	EDR2 Path=3	EDR2
Benchmark Datasets (PSNR)	Set5	38.08	38.11	38.1	38.13	38.17	38.21	38.19	38.09	38.13	**38.24**
	Set14	33.79	33.87	33.89	33.92	34.01	34.03	34.01	33.71	33.88	**34.07**
	B100	32.26	32.29	32.26	32.28	32.29	32.31	32.27	32.16	32.29	**32.37**
	Urban100	32.35	32.32	32.61	32.58	33.07	33.11	33.07	32.25	32.83	**33.16**
	Manga109	38.79	38.74	38.85	38.83	39.17	39.26	39.12	38.66	38.94	**39.35**
	Average	35.05	35.07	35.14	35.15	35.34	35.38	35.33	34.97	35.21	**35.44**

Table 3
The FRB quantity comparison experiment. Bolded fonts indicate the best results

Methods	Scale	Set5	Set14	BSD 100	Urban 100	Manga 109
MFRAN	×2	38.24	34.07	32.37	33.16	39.35
MFRAN-L		**38.25**	**34.15**	**32.39**	**33.28**	**39.47**
MFRAN	×3	34.76	30.6	29.3	28.95	34.43
MFRAN-L		**34.79**	**30.66**	**29.33**	**29.05**	**34.52**
MFRAN	×4	32.62	28.87	27.76	26.82	31.3
MFRAN-L		**32.64**	**28.89**	**27.79**	**26.87**	**31.37**

Table 4
Comparison experiments using different attention modules. Bolded fonts indicate the best results.

Methods	Scale	Set5	Set14	BSD 100	Urban 100	Manga 109
RB	×2	38.16	33.86	32.32	32.81	39.07
DRB		38.19	33.92	32.31	32.86	39.11
DRB+CA		38.18	33.96	32.34	32.98	39.34
DRB+CBAM		38.18	34.01	32.35	32.92	39.17
DRB+DEMA		38.19	34.06	32.34	32.96	39.21
DRAB		**38.24**	**34.07**	**32.37**	**33.16**	**39.35**
RB	×4	32.51	28.78	27.71	26.61	31.01
DRB		32.53	28.86	27.74	26.72	31.24
DRB+CA		32.55	28.85	27.73	26.69	31.22
DRB+CBAM		32.52	28.86	27.74	26.67	31.08
DRB+DEMA		32.55	28.84	**27.77**	26.75	31.14
DRAB		**32.62**	**28.87**	27.76	**26.82**	**31.3**

volutions for the basic blocks in FRB-DR2 of Fig. 2 (d); 6). R2-D: Convolving the basic blocks in FRB-DR2 with the same dilated convolution of the same dilated rate; 7). EDR2 with Mean: Using the EDR2 in Fig. 2 (e). as architecture, the block adopts the dilated convolution with varying dilated rate for feature fusion by the arithmetic mean; 8). EDR2 with Path = 2: we set the path in EDR2 to 2 to verify the impact of the number of paths on performance; 9). EDR2 with Path = 3: set the number of paths to 3; 10). EDR2: 1×1 convolution is adopted as the fusion method, and the number of paths is set to 4.

As can be seen from the Table 2, the performance of the components with dilated convolution is better than standard convolution, as shown in the 1) to 6); Meanwhile, it can be inferred that dilated convolution can extract multi-scale information to a certain extent and thus improve the performance, but the successive stacks of convolutions with the same dilated rate also cause local information loss on large-scale images, so the results on Urban100 and Manga109 are degraded; secondly, the experimental comparison of the components extended by the R1 and R2 rules demonstrates that the R2 extension rule achieves better results on benchmark datasets, which leads to the conclusion that our proposed extension rule is more effective in extracting features; thirdly, to study the effect of fusion operation on the model, we train two models based on the arithmetic mean and 1×1 convolution, the comparison between 7) (EDR2 with Mean) and 10) (EDR2 with Conv) appears that the 1×1 convolution can better fuse the multi-level features extracted from the fractal structure; from the comparison of 8) (EDR2 with path = 2); 9) (EDR2 with path = 3) and 10) (EDR2 with path = 4), it can be observed that FRB has the strongest extraction ability when the number of paths is set to 4. In case the path is set to 5, it is not considered due to the model memory occupation rate and the excessive number of parameters. Accordingly, we use EDR2 with 1×1 convolution and path = 4 as the final architecture of FRB.

Without considering the number of parameters, etc., deepening the network model by increasing the number of FRBs can improve performance, and to further confirm this, we design a set of comparison experiments on the number of FRBs, as shown in Table 3, we compare MFRAN, and MFRAN-L under three scale factors on five benchmark datasets, where FRB = 16 in MFRAN and FRB = 24 in MFRAN-L. MFRAN-L improves the PSNR under three factors compared to MFRAN, which proves that increasing the number of FRBs does improve the network performance. It is worth noting that the number of parameters for the large model MFRAN-L is less than that of EDSR, RDN, etc., and MFRAN's results are also better than EDSR and RDN for the large-scale factor (×4), while the number of parameters is 59% of RDN and 36% of EDSR, which also indicates that our model has greater potential at large-factor.

Fig. 6. Visual comparison of combinations in Table 4. The image "YumeiroCooking" of Manga109 is enlarged by ×4 and the best results are shown in bold font.

5.2. Study of Dilated Residual Attention Block

To investigate the role played by the dilated convolution and attention mechanism in the SR task, we conduct six sets of experiments: 1) Applying two standard convolutions and a relu layer, called Residual Block (RB); 2) Using a combined dilated rate strategy in RB, called Dilated Residual Block (DRB); 3) Adding the channel-wise attention used in RCAN to DRB, called Channel-wise Attention (CA), with the expansion rate in CA set to 16; 4) Employing the convolutional block attention module (A module consisting of concatenated channel attention and spatial attention) to DRB, called CBAM; 5) Replacing the two branches in DECA with mixed attention consisting of channel attention and spatial attention, called Dual Enhanced Mixed Attention (DEMA); 6) DRB with DECA, i.e. Dilated Residual Attention Block (DRAB). To ensure fairness and generalizability, the training environment, and hyperparameters are kept consistent for all groups of experiments, which are conducted under small-scale factor (×2) and large-scale factor(×4), which is shown in Table 4.

First of all, the comparison between 1) and 2) demonstrates that dilated convolution can expand receptive field, and thus improve the model performance. Secondly, regardless of the scale factor of ×2 or ×4, the DRAB has a great improvement in metrics on the five benchmark datasets. Concretely, when the scale factor is ×2, compared with groups 2) and 6), the PSNR of 6) improves 0.3dB and 0.24dB on Urban100 and Manga109, respectively, which verifies the efficiency of the DECA; between group 3) and group 4), the average PSNR of 3) is higher than 4), hence channel-wise attention plays a more significant role than spatial-wise attention; compared with groups 3), 4), 5) and 6), the PSNR of 6) is higher than models that using other attentionn on each dataset, which illustrates that DECA improves model performance better than other attention modules.

Moreover, we provide a visual comparison at ×4 on the Manga109, as shown in Fig. 6, which shows that the model with DRAB reconstructed images not only has the highest PSNR and SSIM, but also the lines of the local zooms and details of the reconstructed images are closer to the real images.

Table 5
Experiments on the number of FRBs and Channels, C indicates the number of channels

Metrics	C=64 FRB=8	C=64 FRB=24	C=64 FRB=16	C=32 FRB=16	C=96 FRB=16
Params(M)	4.97	14.08	9.45	2.4	21.58
Madd(G)	25.89	67.41	45.24	12.05	108.03
Flops(G)	12.97	33.76	22.66	6.05	54.08
Set5(PSNR)	38.13	38.23	38.21	37.98	38.24

5.3. Study of Multi-scale Fractal Residual Attention Network

The performance of the multi-scale fractal residual attention the network is correlated with the number of FRBs and channels, so we set up two sets of comparison experiments. First, keep the number of channels as 64, and then verify the PSNR of the model when the number of FRBs is 8, 16, and 24; second, keep the number of FRBs as 16, and then verify the PSNR of the model when the number of channels is 32, 64, and 96, respectively.

The experimental results are shown in Table 5. After training the same epoch, the PSNR of the model gradually improves with the increase of the number of FRBs when the number of channels is set to 64. Later, when the number of FRBs is set to 16, the PSNR of the model is proportional to the number of channels. Although increasing the number of channels and FRBs can enhance the fitting ability of the network and thus obtain higher PSNR, the number of parameters, flops and MAdd will also increase rapidly. Especially when the number of channels is increased to 96, the PSNR of the model only improves 0.03dB, while parameters, flops and MAdd increase significantly, which leads to more time-consuming training. Therefore, in order to save computational resources while maintaining model performance, we selected FRB = 16 and C = 64 as the final model.

5.4. Compare with the best methods

Table 6 shows the comparison of parameters and performance of MFRAN, IPT[49], and SwinIR[50], among which IPT and SwinIR are based on Transformer, and MFRAN is based on CNN. Although IPT and SwinIR are slightly higher than MFRAN, the parameters of IPT are more than 10 times that of MFRAN. Meanwhile, the MAdd of SwinIR reached 600G, while MFRAN was only 48G. Therefore, the time spent to complete the training of IPT and SwinIR was much longer than that of MFRAN, which further indicated that MFRAN had better-balanced model performance and complexity.

5.5. Differences with similar methods

Comparison with MSRN & MDCN. MSRN [20] and MDCN [21] mainly extract features through two-path, while MFRAN uses four-path to extract richer horizontal features. Furthermore, MSRN and MDCN use more channels to expand the receptive field, compared to DRAB, which obtains a larger receptive field and avoids an increase in the number of parameters, reducing the computational burden.

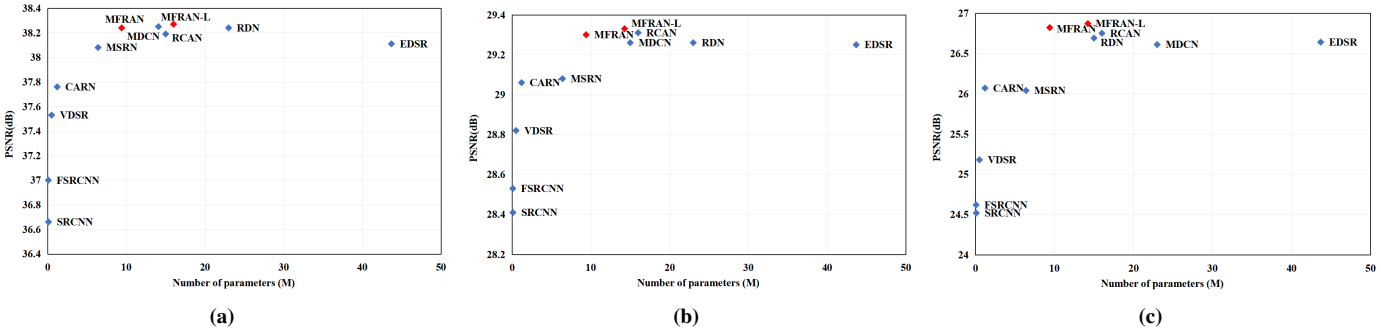

Fig. 7. PSNR performance versus the number of parameters. (a).comparison of parameters on Set14 at ×2; (b).Comparison of parameters on B100 at ×3; (c).Comparison of parameters on Urban100 at ×4, the red diamonds represent our results.

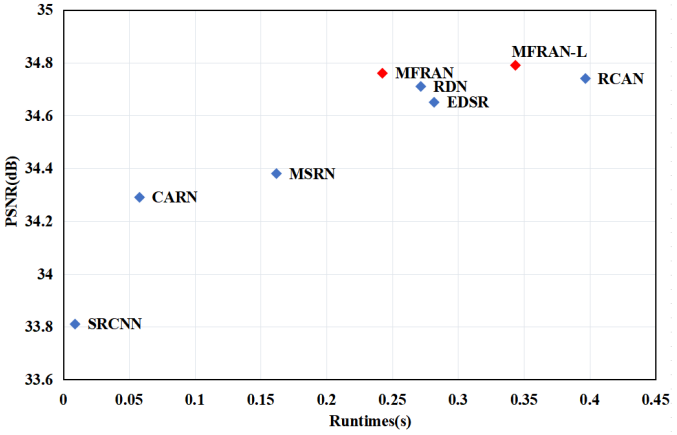

Fig. 8. PSNR performance versus runtime. Results are evaluated on the Set5 dataset with a scale factor of ×3, the red diamond represents our method.

Table 6
Comparison experiment with IPT and SwinIR parameters and performance at ×4

Methods	Params	Set5	Set14	BSD 100	Urban 100	Manga 109
IPT	115.5M	32.64	29.01	27.82	27.26	N/A
SwinIR	11.7M	32.72	28.94	27.83	27.07	31.67
MFRAN	9.6M	32.62	28.87	27.76	26.82	31.30

Comparison with MSFRN. The PSNR of MFRAN(ours) is 0.49dB higher than MSFRN [33] on average at three factors (×2, ×3 and ×4), which indicates that the performance of MFRAN (ours) is better than that of MSFRN, which is demonstrated in Table 1.

Structurally, MSFRN uses a four-path fractal structure of 3×3 convolution and 5×5 convolution to extract features. In contrast, the structure of MFRAN(ours) is more complex, the DRAB is used as the basic block. Compared with the standard 5×5 convolution, DRAB not only has a larger receptive field, but also does not increase the number of model parameters, so that the model can extract more abundant LR multi-scale features. In addition, DECA enhances the feature extraction ability of DRAB, which makes the network work more excellent performance in detail, texture reconstruction.

5.6. Model parameters and inference speed

Increasing the network depth by stacking modules is the easiest and most effective way to improve model performance, but it also increases the number of model parameters, resulting in more computational resources being used. As shown in Fig. 7., we provide the comparison between MFRAN, MFRAN-L, and existing good models in terms of performance and the number of network parameters. First, it can be seen that the MFRAN significantly outperforms networks such as CARN and MSRN; second, compared to large networks such as EDSR, RDN and RCAN, MFRAN achieves better results with a fewer number of parameters. Furthermore, we also conduct comparative experiments on inference speed on benchmark datasets, and for a fair comparison, all methods are tested on the same CPU. From Fig. 8, compared to EDSR, RDN, and RCAN, MFRAN has and faster inference speed at all scale factors, these comparisons illustrate that our model has a better balance between performance and model size.

6. Conclusion

In this paper, a multi-scale fractal residual attention network for SISR is proposed, in which FRB constructs a multi-path feature extraction structure and fuses horizontal features to integrate information under different depth paths; DRAB has a larger receptive field compared with the traditional residual block, which can extract multi-scale features and obtain more accurate SR images; DECA can not only achieve an adaptive selection of channels compared with the traditional channel attention, but also increase the information interaction between channels. A comprehensive evaluation of the benchmark dataset demonstrates that the MFRAN has advantages in reconstruction accuracy and visual quality, and can obtain competitive results with fewer parameters, thus achieving a good balance between model size and performance.

The pixel-level loss ignores the structural information of the image, which will lead to smooth reconstructed images and thus loss of detail information. In the future, we will investigate mixed loss such as perceptual loss, content loss to further improve the model performance. Moreover, we will devote ourselves to extending MFRAN to other restoration tasks (e.g., im-

age deblurring, denoising, and deraining), such as helping other tasks to extract multi-scale features through FRB, and helping the network to focus on key information such as rain spots and blurred regions through DECA.

References

[1] Zhang, L, Wu, X. An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE transactions on Image Processing 2006;15(8):2226–2238.

[2] Tai, YW, Liu, S, Brown, MS, Lin, S. Super resolution using edge prior and single image detail synthesis. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE; 2010, p. 2400–2407.

[3] Sun, J, Xu, Z, Shum, HY. Image super-resolution using gradient profile prior. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2008, p. 1–8.

[4] Ma, C, Yu, P, Lu, J, Zhou, J. Recovering realistic details for magnification-arbitrary image super-resolution. IEEE Transactions on Image Processing 2022;31:3669–3683.

[5] Zhang, W, Liu, Y, Dong, C, Qiao, Y. Ranksrgan: Super resolution generative adversarial networks with learning to rank. IEEE Transactions on Pattern Analysis and Machine Intelligence 2021;44(10):7149–7166.

[6] Wang, L, Kim, TK, Yoon, KJ. Joint framework for single image reconstruction and super-resolution with an event camera. IEEE Transactions on Pattern Analysis and Machine Intelligence 2021;44(11):7657–7673.

[7] Chang, H, Yeung, DY, Xiong, Y. Super-resolution through neighbor embedding. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.; vol. 1. IEEE; 2004, p. I–I.

[8] Yang, J, Wright, J, Huang, T, Ma, Y. Image super-resolution as sparse representation of raw image patches. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE; 2008, p. 1–8.

[9] Yang, J, Wright, J, Huang, TS, Ma, Y. Image super-resolution via sparse representation. IEEE transactions on image processing 2010;19(11):2861–2873.

[10] Lee, J, Jin, KH. Local texture estimator for implicit representation function. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 1929–1938.

[11] Dong, C, Loy, CC, He, K, Tang, X. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 2015;38(2):295–307.

[12] Dong, C, Loy, CC, Tang, X. Accelerating the super-resolution convolutional neural network. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer; 2016, p. 391–407.

[13] Shi, W, Caballero, J, Huszár, F, Totz, J, Aitken, AP, Bishop, R, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 1874–1883.

[14] Kim, J, Lee, JK, Lee, KM. Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 1646–1654.

[15] Lim, B, Son, S, Kim, H, Nah, S, Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017, p. 136–144.

[16] Ledig, C, Theis, L, Huszár, F, Caballero, J, Cunningham, A, Acosta, A, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 4681–4690.

[17] Zhang, Y, Tian, Y, Kong, Y, Zhong, B, Fu, Y. Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 2472–2481.

[18] 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 European conference on computer vision. 2018, p. 286–301.

[19] Yang, H, Zhang, Y, Cui, Z, Xu, Y, Yang, Y. Dgrn: Image super-resolution with dual gradient regression guidance. Computers & Graphics 2023;110:141–150.

[20] Li, J, Fang, F, Mei, K, Zhang, G. Multi-scale residual network for image super-resolution. In: Proceedings of the European conference on computer vision. 2018, p. 517–532.

[21] Li, J, Fang, F, Li, J, Mei, K, Zhang, G. Mdcn: Multi-scale dense cross network for image super-resolution. IEEE Transactions on Circuits and Systems for Video Technology 2020;31(7):2547–2561.

[22] Liu, Y, Wang, Y, Li, N, Cheng, X, Zhang, Y, Huang, Y, et al. An attention-based approach for single image super resolution. In: 2018 24Th international conference on pattern recognition. 2018, p. 2777–2784.

[23] Qin, J, Huang, Y, Wen, W. Multi-scale feature fusion residual network for single image super-resolution. Neurocomputing 2020;379:334–342.

[24] Zhang, D, Shao, J, Liang, Z, Liu, X, Shen, HT. Multi-branch networks for video super-resolution with dynamic reconstruction strategy. IEEE Transactions on Circuits and Systems for Video Technology 2020;31(10):3954–3966.

[25] Dai, T, Cai, J, Zhang, Y, Xia, ST, Zhang, L. Second-order attention network for single image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 11065–11074.

[26] Zhang, Y, Li, K, Li, K, Zhong, B, Fu, Y. Residual non-local attention networks for image restoration. arXiv preprint arXiv:190310082 2019;.

[27] Yang, Y, Qi, Y. Hierarchical accumulation network with grid attention for image super-resolution. Knowledge-Based Systems 2021;233:107520.

[28] Wang, Z, Lu, Y, Li, W, Wang, S, Wang, X, Chen, X. Single image super-resolution with attention-based densely connected module. Neurocomputing 2021;453:876–884.

[29] Liu, H, Cao, F, Wen, C, Zhang, Q. Lightweight multi-scale residual networks with attention for image super-resolution. Knowledge-Based Systems 2020;203:106103.

[30] Lv, X, Wang, C, Fan, X, Leng, Q, Jiang, X. A novel image super-resolution algorithm based on multi-scale dense recursive fusion network. Neurocomputing 2022;489:98–111.

[31] Wang, C, Lv, X, Ding, W, Fan, X. No-reference image quality assessment with multi-scale weighted residuals and channel attention mechanism. Soft Computing 2022;26(24):13449–13465.

[32] Larsson, G, Maire, M, Shakhnarovich, G. Fractalnet: Ultra-deep neural networks without residuals. arXiv preprint arXiv:160507648 2016;.

[33] Feng, X, Li, X, Li, J. Multi-scale fractal residual network for image super-resolution. Applied Intelligence 2021;51(4):1845–1856.

[34] Zhao, H, Gallo, O, Frosio, I, Kautz, J. Loss functions for image restoration with neural networks. IEEE Transactions on computational imaging 2016;3(1):47–57.

[35] 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 IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 1575–1584.

[36] Wang, P, Chen, P, Yuan, Y, Liu, D, Huang, Z, Hou, X, et al. Understanding convolution for semantic segmentation. In: 2018 IEEE winter conference on applications of computer vision. Ieee; 2018, p. 1451–1460.

[37] Agustsson, E, Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017, p. 126–135.

[38] Bevilacqua, M, Roumy, A, Guillemot, C, Alberi-Morel, ML. Low-complexity single-image super-resolution based on nonnegative neighbor embedding 2012;.

[39] Zeyde, R, Elad, M, Protter, M. On single image scale-up using sparse-representations. In: Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7. Springer; 2012, p. 711–730.

[40] 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 Eighth IEEE International Conference on Computer Vision. ICCV 2001; vol. 2. IEEE; 2001, p. 416–423.

[41] Huang, JB, Singh, A, Ahuja, N. Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 5197–5206.

[42] Matsui, Y, Ito, K, Aramaki, Y, Fujimoto, A, Ogawa, T, Yamasaki, T, et al. Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications 2017;76:21811–21838.

[43] Wang, Z, Bovik, AC, Sheikh, HR, Simoncelli, EP. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 2004;13(4):600–612.
[44] Zhang, K, Zuo, W, Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 3262–3271.
[45] Li, J, Yuan, Y, Mei, K, Fang, F. Lightweight and accurate recursive fractal network for image super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019, p. 0–0.
[46] Fang, F, Li, J, Zeng, T. Soft-edge assisted network for single image super-resolution. IEEE Transactions on Image Processing 2020;29:4656–4668.
[47] Zhang, J, Long, C, Wang, Y, Piao, H, Mei, H, Yang, X, et al. A two-stage attentive network for single image super-resolution. IEEE Transactions on Circuits and Systems for Video Technology 2021;32(3):1020–1033.
[48] Kingma, DP, Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 2014;.
[49] Chen, H, Wang, Y, Guo, T, Xu, C, Deng, Y, Liu, Z, et al. Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 12299–12310.
[50] Liang, J, Cao, J, Sun, G, Zhang, K, Van Gool, L, Timofte, R. Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 1833–1844.