Enhancing Single Image Super Resolution

Chen, Zheyi; Chang, Xiang; Chao, Fei; Chen, Yanjie; Shang, Changjing; Shen, Qiang

Published in:
UKCI 2024

Publication date:
2024

Citation for published version (APA):
Chen, Z., Chang, X., Chao, F., Chen, Y., Shang, C., & Shen, Q. (2024). Enhancing Single Image Super Resolution: A Galerkin-type Attention Mechanism-Based Approach with Residual Channel Attention. In *UKCI 2024: 23rd UK Workshop on Computational Intelligence*

Enhancing Single Image Super Resolution: A Galerkin-type Attention Mechanism-Based Approach with Residual Channel Attention Networks

Zheyi Chen[1], Xiang Chang[2], Fei Chao[1,2], Yanjie Chen[2,3], Changjing Shang[2], and Qiang Shen[2]

[1] Xiamen University, Fujian 361005, China,
fchao@xmu.edu.cn
[2] Aberystwyth University, Wales, SY23 3DB, UK,
[3] Fuzhou University, Fujian, 350116, China

Abstract. Deep learning networks effectively address the challenge of transforming low-resolution images into high-resolution images by learning from a series of LR-HR sample pairs. However, most network models are specifically trained for certain scales, and each set of network parameters is only applicable to a particular scale of super-resolution problems. To address this issue, this study introduces an arbitrary-scale super-resolution neural operator network based on a Galerkin attention mechanism, integrating Residual Channel Attention Networks as a replacement for the original feature extraction module. Furthermore, it investigates the impact of different loss functions, training epochs, and feature extraction modules on the performance of the super-resolution neural operator. Experimental results validate the performance of the proposed feature extraction module. The findings indicate that, under the same loss functions and training epochs, the improved module exhibits smaller losses on the training set compared to the original module, demonstrating enhancements. Even with significantly more training epochs, the visual effects of the original network using EDSR-Baseline as the feature extraction module still fall short of those achieved by the improved network.

Keywords: Super-resolution, Galerkin-type attention mechanism, Low-level vision processing, Deep neural network

1 Introduction

With the continuously enriching social demands, people's requirements for image resolution are also increasing [9, 11]. However, in many fields, the image resolution fails to meet these demands. For instance, numerous surveillance imaging systems do not possess optical zoom capabilities [12], thus unable to provide high-resolution images of critical targets. Presently, there are two primary methods to enhance image quality: hardware and software. The hardware route is

expensive and offers limited room for improvement. Conversely, the software approach utilizes super-resolution technology to enhance image quality, which is more cost-effective and yields better results. Super-resolution technology finds extensive technical applications in various practical scenarios, such as medical imaging [5], remote sensing imaging [6], and video surveillance and security [13]. Super Resolution refers to the process of restoring High-Resolution images from low-resolution images through a series of computer algorithms [10, 8]. The super-resolution problem has evolved into a significant research direction in computer vision and image processing. In real life, due to the limitations of camera sensors and image compression losses, the images people obtain are often not clear enough. High-resolution images, with their higher pixel density and more detailed features, provide additional information and play a crucial role in practical applications. As a fundamental research task in the field of computer vision, super-resolution reconstruction has gradually become a data augmentation and preprocessing method for other computer vision tasks [14].

The field of deep learning-based super-resolution algorithms has seen rapid advancements, with numerous high-performing methods emerging. These approaches have addressed many issues present in traditional super-resolution techniques, yet some challenges remain: Most current super-resolution reconstruction algorithms have moved away from traditional methods [15], resulting in a lack of sufficient interpretability. Deep learning methods combined with traditional techniques are easier to explain and their decision-making logic is more comprehensible, providing a foundation for further algorithmic improvements.

Thus, this paper presents a novel arbitrary integer upsampling super-resolution algorithm, based on the Local Implicit Image Function model and the Enhanced Deep Residual Networks model, integrated with the "Galerkin" attention mechanism. The proposed network employs an innovative neural operator architecture, which enhances both generalization and interpretability, outperforming existing continuous super-resolution methods in terms of accuracy and runtime.

The remainder of this paper is organized as follows: Section 2 introduces an arbitrary scale single image super-resolution algorithm incorporating the Galerkin attention mechanism. The super-resolution task is modeled based on the training process and optimization objectives. Section 3 specifies the datasets used for network training and testing, and provides detailed descriptions of the experimental training parameters and training methods. The section also introduces evaluation methods for super-resolution results. Section 4 concludes the paper and points out important future work.

2 Method

The proposed network architecture of the arbitrary scale super-resolution neural operator with the Galerkin attention mechanism is shown in Figure 1. The network consists of three parts: (1) Feature Extraction Module L, responsible for generating feature maps of low-resolution images, typically a super-resolution network without the upsampling module, such as EDSR-Baseline. (2) Attention

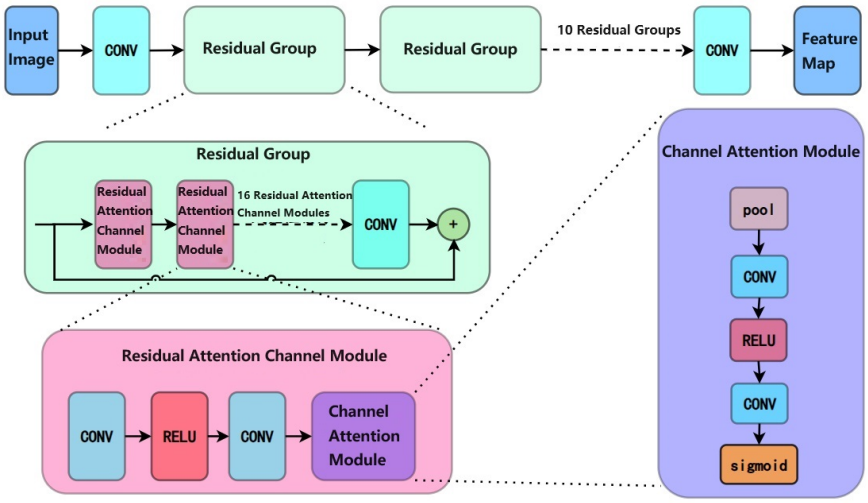

Fig. 1. Proposed network architecture

Module K, composed of a series of iterative kernel integrals, responsible for amplifying and enhancing the feature maps. (3) Projection Module P, which is the upsampling module, is responsible for generating the final SR image.

An image can be considered as a constant function $f(x)$. L and P learn mappings between approximate spaces $H(\Omega_{hc})$ and $H(\Omega_{hf})$, corresponding to the grid sizes of the function (analogous to the image dimensions) hc and hf. Specifically, these two modules implement feature extraction and projection in the experiments. The K module is responsible for enhancing the feature maps by approximating the multi-head attention function through a series of independently trainable kernel integrals.

2.1 Feature Extraction Module

The operator L is responsible for learning the mapping of the approximate space $H(\Omega_{hc})$ with grid size hc. The functionality of the feature extraction module can be described as follows:

$$H(x) = L(f(x)) \tag{1}$$

where $H(x)$ represents the information contained within the image, termed as the Hidden Representation, and is transmitted in the form of feature maps within the network.

The feature extraction module consists of a super-resolution network devoid of an upsampling module. This paper employs two types of upsampling modules: the original EDSR-Baseline module and the improved RCAN module developed in this study. The specific structures of these modules will be described later.

Fig. 2. Network Structure of the Attention Module

2.2 Feature Extraction Module Based on RCAN Network

The feature extraction module based on the RCAN network is divided into two parts: a series of Residual Groups (RG) and two 3×3 convolution blocks at the beginning and end of the module. This module's innovation lies in the adoption of the Channel Attention Mechanism, which, by learning to adjust the weights of different channels, enhances the effective channels while suppressing the ineffective ones, saving computational resources and enhancing feature extraction.

In Figure 1, the input image first undergoes feature extraction through a 3×3 convolution block. This is followed by a series of residual groups, each containing 16 Residual Channel Attention Blocks (RCAB). Finally, the features are output through another 3x3 convolution block. Each Residual Channel Attention Module includes a sequence of a 3×3 convolution block, a Relu activation function, another 3×3 convolution block, and a Channel Attention Layer (CA).

In the channel attention module, a global average pooling layer compresses the dimensionality of the input features into a one-dimensional format. Subsequently, the network learns the importance of different channels through two convolution kernels. The weights assigned to each channel, calculated by a unit comprising sigmoid and Relu functions, are then multiplied by the initial input information to produce the features processed by the channel attention module, achieving feature enhancement.

2.3 Attention Module

The attention module, denoted as K in the neural operator structure, is tasked with learning the latent basis functions suitable for super-resolution tasks. The structure of this module is as follows:

In this experiment, the attention module comprises an LLIF module, a convolution kernel, and two Galerkin attention mechanism modules connected in series.

The LIIF module is responsible for magnifying the feature map to the size of the final generated SR image. The Galerkin-type attention modules are tasked with basic enhancement based on super-resolution tasks, i.e., further extracting features from the image. The Galerkin attention modules employ kernel integral operators to approximate the multi-head self-attention mechanism function through the superposition of finite test functions. Each kernel integral operator is independent, and the network parameters within each operator can be derived through training.

The structure of the kernel integral operator is depicted in the following figure:

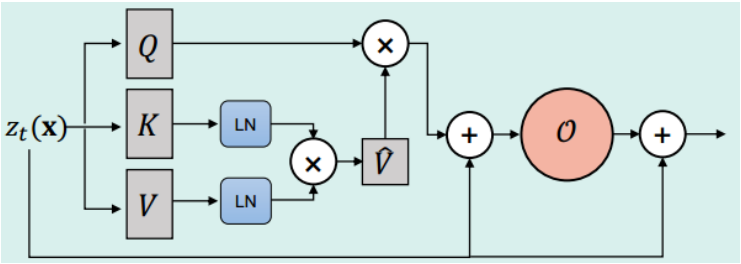

Fig. 3. Structure of the Kernel Integral Operator[4]

The processing logic of the kernel integral operator can be summarized as follows: Initially, the channel count of the input tensor x is expanded to three times its original size, without altering the tensor's height and width, merely the channel count. Subsequently, the tensor is split along its last dimension into three parts: Q, K, and V, which are used to compute self-attention weights. K and V are normalized (Layer Normalization) to accelerate convergence during training. Finally, self-attention weights are calculated using Q, K, and V, depending on the latent representation information of the LR image in Q, K, and V. After computing the attention weights, the output tensor is added to the original input tensor, followed by a point-wise feedforward network O processing, and then added again to the original input tensor. This sequence of operations adds non-linear information to the output, thereby enhancing the module's effectiveness.

The above steps can be represented by the following formula:

$$z_{t+1}(x) = z_t(x) + \mathcal{O}((\mathcal{K}_t(z_t))(x) + z_t(x)), \tag{2}$$

where $z_t(x)$ is the output from the t-th kernel integral operator, $\mathcal{K}_t$ is the kernel integral operator, and $\mathcal{O}$ is the Point-wise FeedForward Network processing.

2.4 Upsampling Module

Upsampling Module

This module corresponds to the projection part P of the neural operator network, learning the mapping of the approximate space $H(\Omega_{hf})$ with grid size hf. The module is responsible for projecting the feature map into the RGB space to generate the SR image. It consists of two fully connected layers, a Gelu activation function, and a bilinear interpolation fusion module. The specific structure is as follows:

Initially, the feature map undergoes a linear transformation and non-linear activation through a fully connected layer and a Gelu activation function. The processed feature map is then passed through another fully connected layer, which serves to project the processed features into the final output space. The bilinear interpolation module processes the feature map and the input grid to generate the output, which is then added to the output from the previous fully connected layer to produce the final SR image.

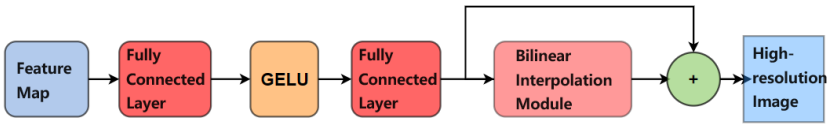

Fig. 4. Network Structure of Upsampling Module

Algorithm 1 Super-Resolution Algorithm Based on Galerkin Attention Mechanism

Input: Low-resolution image LR_Image, Upscaling factor $scale$
Output: High-resolution image reconstructed HR_Image
Get the height and width of the low-resolution image $h, w \leftarrow \text{ImageShape}(LR\_Image)$
Calculate the height and width of the reconstructed HR image $H \leftarrow \text{int}(h \times scale), W \leftarrow \text{int}(w \times scale)$
Generate grid for pixel positioning $coord \leftarrow \text{make\_coord}(H, W)$
Extract features from the LR image $LRfeat \leftarrow \text{feature\_extract}(LR\_Image)$
Magnify the LR image features using the LIIF function $HRfeat \leftarrow \text{LIIF}(LRfeat, scale, coord)$
Enhance the HR feature map using the Galerkin attention module $HRfeat \leftarrow \text{Galerkin\_Attention}(HRfeat)$
Upsample to obtain the super-resolved image $HR\_Image \leftarrow \text{Upsampling}(HRfeat)$

3 Experimentation

The experimental plan for this work is as follows:

- Test models before and after improvements, using the original feature extraction module based on EDSR-Baseline and the improved feature extraction module based on RCAN. Test multiple sets of parameters during the training process according to the order of parameter iteration to explore the performance differences between different feature modules.
- Compare the effects of four different loss functions (L1 loss, VGG loss, Mixed loss, Charbonnier loss) on image super-resolution across multiple public datasets (Set5, Set14, B100, Urban100) to identify the best combination of loss functions.
- Investigate the patterns of model performance trained with three different loss functions (VGG loss, Mixed loss, Charbonnier loss) as training epochs increase, find the general optimal number of training rounds (TR), and provide a data basis for future training planning.

3.1 Training Parameter

The entire development process utilized the Ubuntu LTS operating system. This paper established the coding environment using Python 3.9, completed the coding with PyTorch 1.10.2, and conducted model parameter training on a GeForce RTX 3090Ti.

During network training, each batch randomly sampled 64 HR-LR image pairs with the same scaling factor, with the LR image blocks sized at 128 × 128. Data augmentation techniques such as random horizontal flipping, vertical flipping, and diagonal flipping were applied to the input images. Each sample was reused a maximum of 20 times during training.

In the training of the EDSR-Baseline feature extraction module, validation was performed every 50 epochs, and model parameters were saved every 250 epochs, totaling 1,050 epochs of training. For the RCAN feature extraction module, due to time constraints, the training was limited to 100 epochs, with parameter saving every 10 epochs.

The initial learning rate for training was set at 4×10^{-5}, with a maximum learning rate of 4×10^{-4}, using the Adam optimizer. The learning rate included a warm-up process. Within the first 50 epochs, the learning rate gradually increased. After the warm-up phase of 50 epochs, the learning rate was decayed using a cosine annealing algorithm. The process also recorded the last and best parameters during parameter iteration.

3.2 Quantitative Results and Analysis

This section will first conduct performance tests on the improved super-resolution model based on the RCAN feature extraction module and compare it with the original model. To identify the optimal training epochs and the best loss function combination for the super-resolution network with Galerkin-type attention mechanism, this section will test the super-resolution networks trained under different numbers of training epochs and with various loss functions. All models tested in the experiments were trained using the DIV2K dataset [1].

Comparison of Effects of Different Feature Extraction Modules We compared the single-image super-resolution performance of different super-resolution models based on two feature extraction modules (RCAN and EDSR-Baseline) across multiple public datasets (Set5 [2], Set14 [3], B100 [7], Urban100 [3]). The experiments calculated the Peak Signal-to-Noise Ratio (PSNR) as a performance metric. In the experiments, both models had consistent upsampling and attention modules, the upsampling scale was consistently (×4), and the loss function (LF) was based on VGG19 perceptual loss. Due to time constraints, the RCAN module was only trained for 100 rounds. PSNR was used as the evaluation metric in this section.

From the tables above, it is evident that the PSNR scores of both models improve with the number of training rounds. The super-resolution model based on the RCAN feature extraction module achieved relatively high performance within a shorter training period, achieving better PSNR indices across four public datasets in just 20 rounds compared to the EDSR-Baseline model trained for 250 rounds. However, the original model achieved better PSNR indices on multiple standard super-resolution test sets after a longer training period, with each model having its advantages.

Table 1. PSNR indices of the super-resolution model based on the RCAN feature extraction module

Index \ TR	20	40	60	80	100	best
PSNR (Set5)	31.9609	32.0874	32.1600	32.1913	32.2938	32.2938
PSNR (Set14)	28.4735	28.5468	28.6099	28.6010	28.6636	28.6630
PSNR (Urban100)	25.7483	25.9651	26.0729	26.0975	26.2585	26.2585
PSNR (B100)	27.4839	27.5235	27.5716	27.5572	27.6250	27.6250

Table 2. PSNR indices of the super-resolution model based on the EDSR-Baseline feature extraction module

Index \ TR	250	500	750	1000	best
PSNR (Set5)	31.9343	32.2240	32.3198	32.3905	32.3927
PSNR (Set14)	28.4579	28.5964	28.7078	28.7078	28.7503
PSNR (Urban100)	25.7438	26.0670	26.2668	26.3821	26.3821
PSNR (B100)	27.4377	27.5401	27.6100	27.6415	27.6406

Comparison of Training Effects with Different Loss Functions This experiment compared the effects of four different loss functions (L1 loss, VGG19-based perceptual loss, mixed loss, and Charbonnier loss) on image super-resolution across several public datasets (Set5, Set14, B100, Urban100). The experiments were conducted under different magnification scales (x2, x4, x6, x8, x10, x12), calculating the Peak Signal-to-Noise Ratio (PSNR) and processing time (TIME) as performance indicators. The mixed loss function combines Charbonnier loss, VGG19-based perceptual loss, and L2 loss. In the network, the magnification factors for processing the width and height of the LR images are the same and integers. The model parameters representing each loss function were those that performed best on the validation set during training.

Among the 24 experiments conducted on four datasets at six magnification scales, L1 loss, mixed loss, and Charbonnier loss all achieved good PSNR indices, obtaining the best PSNR indices 8, 9, and 7 times, respectively. In contrast, the VGG19-based perceptual loss did not achieve the best PSNR index in any instance. L1 loss, mixed loss, and Charbonnier loss each excelled in their respective datasets; for example, L1 obtained the best PSNR indices in 5 out of 6 experiments on the Urban100 dataset.

In terms of processing time, there is little difference in the performance of networks trained with different loss functions, with processing times generally distributed within a 0.2s interval. The mixed loss function achieved the fastest processing time the most times. The complexity of the datasets significantly affects both PSNR and processing time.

On the Urban100 and B100 datasets, the PSNRs are generally lower than on the Set5 and Set14 datasets, reflecting the increased difficulty of super-resolution

Table 3. Test results on the Set5 dataset

Mag. \ LF	L1 Loss	VGG Loss	Mixed Loss	Charbonnier Loss
$X2$	38.1355 / 1.772s	38.0772/1.734s	38.1024 / 1.747s	38.1273 / 1.759s
$X4$	32.3870 / 1.794s	32.3927/1.826s	32.3960 / 1.748s	32.4292 / 1.782s
$X6$	29.0256 / 1.723s	28.9771/1.754s	29.0169 / 1.863s	29.0048 / 1.788s
$X8$	26.9355 / 1.752s	26.9450/1.777s	26.9479 / 1.756s	26.8625/1.872s
$X10$	25.6759 / 1.712s	25.6044 / 1.749s	25.6787 / 1.748s	25.5866/1.765s
$X12$	24.4694/1.762s	24.4969/1.883s	24.5153 / 1.772s	24.5027 / 1.749s

Table 4. Test results on the Set14 dataset

Mag. \ LF	L1 Loss	Perceptual Loss	Mixed Loss	Charbonnier Loss
$x2$	33.8727/2.785s	33.8578/2.835s	33.8772/2.795s	33.8708/2.763s
$x4$	28.7874/2.672s	28.7503/2.688s	28.7973/2.706s	32.4292/21.782s
$\times6$	26.5315/2.896s	26.5124/2.729s	26.5815/2.809s	29.0048/1.776s
$\times8$	25.0034/2.699s	24.9665/2.678s	25.0471/2.685s	26.8625/1.872s
$x10$	23.9281/2.681s	23.9165/2.692s	23.9071/2.657s	25.5866/1.765s
$x12$	23.1766/2.645s	23.2035/2.644s	23.1679/2.649s	24.5027/1.749s

Table 5. Test results on the B100 dataset

Mag. \ LF	L1 Loss	Perceptual Loss	Mixed Loss	Charbonnier Loss
$\times2$	32.2489/7.251s	32.2095/7.208s	32.2298/7.143s	32.2466/7.247s
$x4$	27.6696/7.146s	27.6406/7.191s	27.6878/7.159s	27.6640/7.122s
$\times6$	25.9153/7.051s	25.9046/7.058s	25.9534/7.123s	25.9167/7.053s
$\times8$	24.9097/7.059s	24.897/7.124s	24.9346/7.054s	24.9033/7.056s
$\times10$	24.1857/7.092s	24.1673/7.143s	24.2014/7.092s	24.1642/7.075s
$x12$	23.6189/6.975s	23.6185/6.976s	23.6283/6.904s	23.6050/6.898s

Table 6. Test results on the Urban100 dataset

Mag./Dataset	L1 Loss	Perceptual Loss	Mixed Loss	Charbonnier Loss
$X2$	32.6054/30.622s	32.4426/30.593s	32.5433/30.431s	32.5706/30.5019s
$X4$	26.4961/28.207s	26.3821/28.327s	26.4553/28.093s	26.4970/28.027s
$x6$	24.0919/27.604s	24.0190/27.614s	24.0858/27.550s	24.0770/27.5369s
$X8$	22.7083/27.783s	22.6743/27.674s	22.7048/27.523s	22.6945/27.6369s
$x10$	21.7794/27.394s	21.7476/27.447s	21.7722/27.277s	21.7531/27.3229s
$x12$	21.1013/27.068s	21.0677/27.168s	21.0768/26.954s	21.0734/26.9899s

tasks. On the Urban100 dataset, image reconstruction times are also generally longer. As the magnification increases, the PSNRs of all methods decrease, indicating that it is more challenging to recover detailed information in images at higher magnifications.

Overall, L1 loss, mixed loss, and Charbonnier loss all exhibited excellent super-resolution performance, while the VGG19-based perceptual loss function performed poorly.

Comparison of Training Effects with Different Training Epochs We compared the effects of three different loss functions (VGG19-based perceptual loss, mixed loss, and Charbonnier loss) on image super-resolution across several public datasets (Set5, Set14, B100, Urban100). The experiments were conducted at the same upsampling scale (x4) and calculated the Peak Signal-to-Noise Ratio (PSNR) and processing time (TIME) as performance indicators.

Table 7. Test results for the model using VGG19 perceptual loss function in terms of Peak Signal-to-Noise Ratio and processing time

Rounds \ Dataset	Set5	Set14	B100	Urban100
$x250$	31.9343/1.839s	28.4579/2.708s	27.437/7.159s	25.7438/28.278s
$x500$	32.2240/1.767s	28.5964/2.746s	27.5401/7.121s	26.0670/28.330s
$x750$	32.3198/1.723s	28.7078/2.716s	27.6100/7.118s	26.2668/28.2829s
x1000	32.3905/1.814s	28.7500/2.721s	27.6415/7.162s	26.3821/28.1939s
best	32.3927/1.768s	28.7503/2.670s	27.6406/7.113s	26.3821/28.291s
last	32.3927/1.757s	28.7503/32.723s	27.6406/7.102s	26.3821/28.211s

Table 8. Test results for the model using Charbonnier loss function in terms of Peak Signal-to-Noise Ratio and processing time

Rounds \ Dataset	Set5	Set14	B100	Urban100
$x250$	31.1496/1.744s	28.5844/2.709s	27.5566/7.122s	26.0111/28.319s
$x500$	32.2067/1.798s	28.6441/2.672s	27.5940/7.137s	26.1810/28.325s
$x750$	32.3579/1.795s	28.7406/2.756s	27.6367/7.147s	26.3921/28.314s
x1000	32.4301/1.749s	28.7780/2.749s	27.6614/7.214s	26.4941/28.347s
best	32.4292/1.769s	28.7826/2.749s	27.6640/7.812s	26.4970/28.283s
last	32.4292/1.781s	28.7826/2.697s	27.6640/7.086s	26.4970/28.290s

Table 9. Test results for the model using mixed loss function in terms of Peak Signal-to-Noise Ratio and processing time

Rounds \ Dataset	Set5	Set14	B100	Urban100
$x250$	32.1290/1.715s	28.5370/2.702s	27.5171/7.046s	25.9822/28.136s
$x500$	32.2561/1.745s	28.6673/2.707s	27.5931/7.125s	26.1777/28.125s
$x750$	32.3433/1.817s	28.7487/2.709s	27.6491/7.058s	26.3467/28.099s
$x1000$	32.3988/1.744s	28.7981/2.727s	27.6491/7.058s	26.3467/28.099s
best	32.3960/1.781s	28.7973/2.671s	27.6878/7.127s	26.4553/28.097s
last	32.3960/1.761s	28.7973/2.692s	27.6878/7.188s	26.4553/28.112s

After 750 training rounds, approximately three-quarters of the network parameters tend to stabilize in terms of improvement in the PSNR index, staying at a level of 0.01dB. As the number of training rounds increases, most network parameters still slowly improve the PSNR index on datasets. However, for some specific datasets, the network parameters obtained after 1,000 rounds of training performed better than those after 1,050 rounds, even though the fluctuation in the index was only 0.01dB or even 0.001dB, but it does not affect the final model performance.

4 Conclusion

This paper introduces a new feature extraction module based on the RCAN network, building upon the foundation of a neural operator super-resolution network equipped with Galerkin-type attention mechanisms. Experiments demonstrate that, when trained with the same perceptual loss function, the improved model based on RCAN can achieve better super-resolution reconstruction results than the EDSR-Baseline with fewer training rounds. Additionally, this paper explores the optimal combination of loss functions and the best number of iteration rounds. It was found that after 750 training iterations, the improvement in the PSNR indices for networks based on different loss functions slowed to a 0.01 dB level across various datasets. In tests of PSNR indices for super-resolution algorithms at different scales on four different public datasets, networks trained with L1 loss, mixed loss, and Charbonnier loss all exhibited excellent super-resolution performance, while those based on the VGG19 perceptual loss function performed poorly.

References

1. 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 (CVPR) Workshops, pp. 126–135 (2017). URL https://arxiv.org/abs/1705.08958

2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. British Machine Vision Conference (BMVC) pp. 135.1–135.10 (2012). URL https://www.bmva.org/bmvc/2012/BMVC/paper0116/paper0116.pdf
3. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5197–5206 (2015). URL https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Huang_Single_Image_Super-Resolution_2015_CVPR_paper.html
4. Luo, X., Qian, X., Yoon, B.J.: Hierarchical neural operator transformer with learnable frequency-aware loss prior for arbitrary-scale super-resolution. In: Forty-first International Conference on Machine Learning (2024). URL https://openreview.net/forum?id=LhAuVPWq6q
5. Mahapatra, D., Bozorgtabar, B., Garnavi, R.: Image super-resolution using progressive generative adversarial networks for medical image analysis. Computerized Medical Imaging and Graphics **71**, 30–39 (2019)
6. Mario, H.J., Ruben, F.B., Paoletti, M.E., et al.: A new deep generative network for unsupervised remote sensing single-image super-resolution. IEEE Transactions on Geoscience and Remote Sensing **56**(11), 6792–6810 (2018)
7. 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), pp. 416–423 (2001). DOI 10.1109/ICCV.2001.937655. URL https://ieeexplore.ieee.org/document/937655
8. Wang, Z., Chen, J., Hoi, S.C.H.: Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(10), 3365–3387 (2020)
9. Wang, Z., Chen, J., Hoi, S.C.H.: Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(10), 3365–3387 (2021)
10. Wei, M., Zhang, X.: Super-resolution neural operator. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18,247–18,256 (2023)
11. Xu, T., Li, L., Mi, P., Zheng, X., Chao, F., Ji, R., Tian, Y., Shen, Q.: Uncovering the over-smoothing challenge in image super-resolution: Entropy-based quantification and contrastive optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–17 (2024). DOI 10.1109/TPAMI.2024.3378704
12. Yin, Y., Robinson, J., Zhang, Y.L., et al.: Joint super-resolution and alignment of tiny faces. Proceedings of the AAAI Conference on Artificial Intelligence **34**(7), 12,693–12,700 (2020)
13. Yin, Y., Robinson, J., Zhang, Y.L., et al.: Joint super-resolution and alignment of tiny faces. Proceedings of the AAAI Conference on Artificial Intelligence **34**(7), 12,693–12,700 (2020)
14. Zhang, H., Liu, D., Xiong, Z.: Convolutional neural network-based video super-resolution for action recognition. In: Proceedings of the 13th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 746–750. Xi'an (2018)
15. Zhang, N., Wang, Y.C., Zhang, X., Xu, D.D.: A review of single image super-resolution based on deep learning. Acta Automatica Sinica **46**(12), 2479–2499 (2020). DOI 10.16383/j.aas.c190031