Multimedia Tools and Applications (2024) 83:55921–55962
https://doi.org/10.1007/s11042-023-17660-4

A review of single image super-resolution reconstruction based on deep learning

Ming Yu[1,2] · Jiecong Shi[1] · Cuihong Xue[3] · Xiaoke Hao[2] · Gang Yan[2]

Received: 11 May 2023 / Revised: 14 September 2023 / Accepted: 7 November 2023 /
Published online: 5 December 2023

Abstract
Single image super-resolution (SISR) is an important research field in computer vision, the purpose of which is to recover clear, high-resolution (HR) images from low-resolution (LR) images. With the rapid developments in deep learning theory and technology, deep learning has been introduced into the field of image super-resolution (SR), and has achieved results far beyond traditional methods in many domains. This paper summarizes current image SR algorithms based on deep learning. Firstly, the mainstream frameworks, loss functions, and datasets used for SISR are introduced in detail. Then, the SISR algorithm based on deep learning is explored using three models: a convolutional neural network (CNN), a generative adversarial network (GAN), and a transformer. Next, the evaluation indices used for SR are introduced, and the reconstruction results from various algorithms based on deep learning are compared. Finally, future trends in research on image SR algorithms based on deep learning are summarized.

Keywords Image super-resolution · Deep learning · Convolutional neural networks · Generative adversarial networks · Transformer

1 Introduction

The purpose of image super-resolution (SR) is to convert input low-resolution (LR) images into high-resolution (HR) images [1]. This is currently one of the main technologies associated with image processing and computer vision (CV), and is a very hot research topic in this field. HR images of real scenes are widely used, because they have a high pixel density and can allow more detailed features of the image to be obtained. SR can therefore provide many advantages for practical applications. It can be divided into two types, single LR image generation HR image technology and multiple LR image generation HR image

✉ Cuihong Xue
redxuech@tjut.edu.cn

[1] School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China

[2] School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China

[3] Technical College for the Deaf, Tianjin University of Technology, Tianjin 300384, China

technology, depending on the number of input images. This paper focuses on single image super-resolution (SISR), which has been widely used in image compression, medical imaging [2–4], remote sensing imaging [5], public security, and other fields due to its flexibility, simplicity, and practicability.

Before the advent of deep learning algorithms, SISR used both interpolation-based and reconstruction-based algorithms. Interpolation-based algorithms are simple and run fast, but the resulting image is too smooth, and high-frequency information is lost, resulting in a ringing effect. The outcomes obtained from a reconstruction-based algorithm are better than those of an interpolation-based algorithm, but the execution efficiency is low, and this approach is sensitive to the scaling factor. Thanks to the development of deep learning methods, the algorithm can now learn the mapping relationship between LR and HR images, and then realize image reconstruction through SR image reconstruction algorithms, giving better results than traditional algorithms. Lightweight networks [6] have also been proposed in recent years, which allow SISR models to use fewer parameters while achieving excellent SR results.

The rest of this article is arranged as follows. First, some background knowledge related to image SR is introduced in Sect. 2. In Sect. 3, classic algorithms for image SR are introduced, with a focus on three types: convolutional neural networks (CNNs), generative adversarial networks (GANs), and transformers. Evaluation indicators for SR are discussed in Sect. 4. Finally, prospects for the development of SR are reviewed in Sect. 5.

2 Basics of image super-resolution reconstruction

2.1 Super-resolution framework

The framework of a SISR system consists of two main parts: a nonlinear mapping learning module, and an upsampling module for enlarging images. The functions of the former are to learn the mapping of LR images to HR images, and to use the loss function to guide and supervise the learning process in the learning process, while the latter carries out the enlargement of the reconstructed image. The two modules work together to finalize the SR reconstruction of the input image. Depending on the location of the upsampling module, SISR methods can be divided into four super-scoring frameworks, as shown in Fig. 1 [7] and discussed below.

Front-end upsampling SR framework This framework was the first to be adopted by researchers in the field, and is shown in Fig. 1(a). Front-end upsampling can avoid the need to learn low-dimensional to high-dimensional mapping in low-dimensional space, can reduce the difficulty of learning, and is a simple and easy method. However, the noise and blur are also enhanced, and convolution operations in high-dimensional space will increase the amount of computation done by the model and consume more computing resources.

Back-end upsampling SR framework As shown in Fig. 1(b), in view of the problems with the front-end upsampling SR framework, and to improve the efficiency of computing resource usage, researchers proposed an SR framework in which the upsampling module was located in the back end of the network. In this approach, most of the convolution computations are performed in low-dimensional space. Finally, an end-to-end learnable

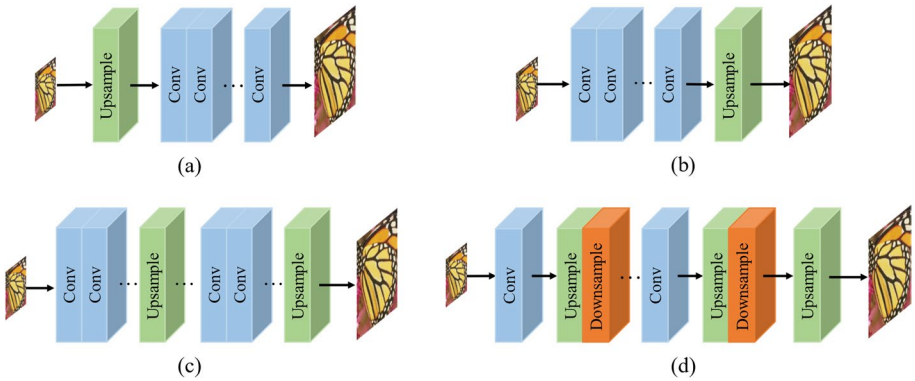


Fig. 1 Four SR frameworks ((**a**) front-end SR framework (**b**) back-end SR framework (**c**) progressive upper sampling SR framework (**d**) ascending and descending sampling iteration SR framework)

upsampling layer is applied, which further releases the computational power of convolution and reduces the complexity of the model.

Progressive upsampling SR framework With the ongoing developments in this area, the scale of SR continues to increase, and conventional upsampling models can no longer meet the requirements. In this context, a progressive upsampling SR framework has been proposed, as shown in Fig. 1(c). In this framework, image upscaling is progressive, with images generated halfway through the process continuing to be fed into subsequent modules until the target resolution is reached. The most common methods involve the use of a convolutional cascade or Laplace pyramid, combined with multi-level supervision and other learning strategies, which allows the SR task to be completed with a large super-multiplication coefficient.

Ascending sampling iterative SR framework Timofte et al. [8] extended the idea of reverse projection [9] by proposing an iterative SR framework for upward and downward sampling, as shown in Fig. 1(d). Subsequently, Haris et al. adopted this SR framework with alternate up-sampling and downsampling in DBPN [10], a and combined all the feature maps obtained in this way to reconstruct LR images. This method can fully learn the mapping relationship between LR and HR by repeatedly performing mapping learning. However, at present, the structure of this super-division framework is complex, and the design standards are not clear, meaning that this approach needs to be further explored further.

2.2 Loss Function

The loss function is one of the essential elements of deep learning models. The role of the loss function in the task of SISR is to quantify the difference between the generated HR image and the ground truth HR image, so that the SISR network model can predict the direction of the true value. The change in the loss function reflects the gap between the training of the current model and the expectation, and can be used to regulate the learning direction of the model. In the field of SR, the pixel loss is most commonly used, although other forms such as the content loss, texture loss, adversarial loss, and perception loss are also used. These loss functions are described below.

Pixel loss This loss function uses pixel values to measure the difference between two images, including the mean square error (MSE, also known as the L2 loss), the mean absolute error (MAE, also known as the L1 loss), and the Charbonnier loss (an improved form of the L1 loss). When deep learning first emerged in the field of SR reconstruction, all of the proposed network models used L2 as the loss function. The expression for the L2 loss function is shown in Eq. (1):

$$L_2 = \frac{1}{n}\sum_{i=1}^{n}(I_{SR}^{i} - I_{HR}^{i})^2 \tag{1}$$

where n represents the number of training samples, I_{SR}^{i} represents the reconstructed image, and I_{HR}^{i} represents the ground truth high-resolution images.

The function curve for MSE is smooth, continuous, and can be guided everywhere, which is convenient when using the gradient descent algorithm: as the error decreases, the gradient also decreases, which is conducive to convergence, even if a fixed learning rate is used, meaning that the algorithm can converge quickly to the minimum. Due to the characteristics of the squaring operation, when the difference is less than one, the error will be reduced, although when the difference is greater than one, the MSE will be very sensitive to this error and will give it a higher weight. This means the MSE is extremely sensitive to outliers, causing the prediction effect to be sacrificed within the normal error range and making the final reconstruction image smoother, blurred, and lacking in high-frequency texture detail. To improve the reconstruction effect of the model, the L1 loss is typically used, and the expression for this is given in Eq. (2):

$$L_1 = \frac{1}{n}\sum_{i=1}^{n}\left|I_{SR}^{i} - I_{HR}^{i}\right| \tag{2}$$

compared with MSE, MAE has the advantage that it is less sensitive to outliers, so it can maintain a stable gradient for various input values. This avoids the problem of gradient explosion, and the training process is therefore relatively stable. However, it has non-derivable points, which are not conducive to the convergence of functions and learning by models. In practice, the actual effect of the L1 loss function is better than that of MSE, as it can improve the performance of the model and obtain higher indicators.

To overcome the problem of non-reducible points with the L1 loss, Lai et al. [11] used the Charbonnier loss in LapSRN, the expression for which is shown in Eq. (3). This approach addresses the drawbacks of the L1 loss by introducing a constant ε.

$$L_{char} = \frac{1}{n}\sum_{i=1}^{n}\rho(I_{SR}^{i} - I_{HR}^{i}) \tag{3}$$

where $\rho = \sqrt{x^2 + \varepsilon^2}$,$\varepsilon$ is a very small constant, generally take 10^{-3}.

Content loss To improve the perceived quality of images, a content loss function is introduced. Unlike the pixel loss, the content loss does not require precision at the pixel level, but instead focuses on the similarity at the sensory level of the human eye. A pre-trained image classification network is typically used to evaluate the semantic differences between two images. The expression for this loss is shown in Eq. (4):

$$L_{content} = \frac{1}{n_l}\sqrt{\sum_{i,j}(\Phi_{i,j}^{(l)}(I) - \Phi_{i,j}^{(l)}(\hat{I}))^2} \tag{4}$$

where n_l represents the number of pixels corresponding to the feature map of the l layer, and $\Phi_{i,j}^{(l)}(I)$ and $\Phi_{i,j}^{(l)}(\hat{I})$ respectively represent the feature maps obtained by the j-th convolution before the i-th largest pooling layer in the i-th largest pooling layer in the I layer.

Texture loss Since the reconstructed image should have the same style as the target image (e.g., in terms of color, texture, and contrast), the texture of the image can be considered as a correlation between different feature channels, which can be represented by matrix point multiplication. The texture loss utilizes the Gram matrix $G^{(l)} \in \boldsymbol{R}^{C_l \times C_l}$, which is expressed as shown in Eq. (5):

$$G_{i,j}^{(l)}(I) = F_i^{(l)}(I) \cdot F_j^{(l)}(I) \tag{5}$$

where $F_i^{(l)}(I)$ is the feature map of the i-th channel of the l-th layer of image I; $F_j^{(l)}(I)$ is the feature map of the j-th channel of the l-th layer of image I. The expression for texture loss is shown in Eq. (6):

$$L_{texture} = \frac{1}{C_l^2}\sqrt{\sum_{i,j}(G_{i,j}^{(l)}(I) - G_{i,j}^{(l)}(\hat{I}))^2} \tag{6}$$

Adversarial loss The concept of the adversarial loss is drawn from GANs. In 2017, Ledig et al. [12] introduced GANs into the SR field for the first time, proposing the SRGAN model and using adversarial losses in their work. The expressions for the adversarial loss in SRGAN are shown in Eqs. (7) and (8):

$$L_{Gen}(I_{SR}) = -\log(D(I_{SR})) \tag{7}$$

$$L_D(I_{SR}, I_{HR}) = -\log(D(I_{HR})) - \log(1 - D(I_{SR})) \tag{8}$$

where $L_{Gen}(I_{SR})$ is the loss function of the generator, $L_D(I_{SR}, I_{HR})$ is the loss function of the discriminator, and $D(I_{SR})$ represents the probability that the image generated by the generator I_{SR} is a natural image.

Perceptual loss Since the pixel-based loss function always uses an average, the perceived textural quality of the image is smoother and visually unsatisfactory. When GANs started to be applied to the field of image SR, perception-based loss functions were widely used, because they could recover richer high-frequency (HF) details. In this approach, the perceptual loss is optimized by measuring the distance between extracted features, in a process that can effectively improve the perceived quality of the image. In SRGAN, the perception function is defined as the weighted sum of the content loss and adversarial loss, as shown in Eq. (9):

$$L^{SR} = L_{content} + 10^{-3}L_{Gen}(I_{SR}) \tag{9}$$

where L^{SR} is the perceived loss, $L_{content}$ is the content loss, and $L_{Gen}(I_{SR})$ is the adversarial loss.

2.3 Datasets

The dataset is an important aspect of an SISR network, and the selection of an appropriate training dataset can greatly improve the performance of a network. In the SISR field, there are now many datasets that are used for model training and testing, which vary in terms of the number, size, type, and mode of degradation of the images. Most datasets contain only HR images, and do not contain LR-HR image pairs under different magnifications. Hence, to construct appropriate image pairs, the bicubic interpolation algorithm is required.

In this section, we introduce some commonly used SISR datasets, and these are summarized in Table 1. The table shows the name, size, download address, source, and purpose of each dataset. From the experimental results obtained from various SR models, it can be seen that the most commonly used test sets are Set5 [13], Set14 [9],Urban100 [14], BSD100 [15], and Manga109 [16]. These five datasets usually have a collection called the benchmark, which contains pictures of people, animals and plants, buildings, food, natural landscape, and the environment. DIV2K [17] is also a popular SISR dataset that contains 1,000 images of different scenes, including people, handicrafts, environments, landscapes, etc. Of these, 800 are used for training, 100 are used for validation, and 100 are used for testing. This dataset was developed to enable SR to be studied based on more realistic degraded images, and has been adopted in several works [18–20]. Flickr2K [21] is another large extended dataset containing 2,650 2 K images (mostly of people, flora and fauna, architecture, and scenery), and is used for training. In recent years, researchers in the SR field have often merged DIV2K and Flickr2K to form the DF2K training dataset, to improve the network performance of SR. The authors of [22–24]and other works adopted this dataset to train their models. RealSR [25] was the first truly collected SISR dataset to contain paired LR and HR images, and includes 595 pairs of LR-HR images, taken with two different digital cameras (Canon 5D3 and Nikon D810), of various indoor and outdoor scenes. The ImageNet [26] dataset, which has been used for image classification, has also been introduced into the SR field in recent years. This dataset contains more than 14 million.

full-size tagged images of animals, plants, transportation tools, furniture, musical instruments, tectonics, tools, etc., and the authors of [31] uses this dataset. In addition to the more commonly used datasets described above, there are several others, such as L20 [8] (including people, flora and fauna, architecture, landscape, etc., with sizes ranging from 3 to 29 million pixels), OutdoorScene [28] (including seven types of textured images of animals, architecture, grass, mountains, plants, sky, and water), PIRM [29] (including images of people, objects, environment, plants, natural landscapes, etc.), MSCOCO [32] (including 91 object categories commonly used for object detection, segmentation, etc.), PIPAL [33] (a perceptual image quality evaluation dataset that includes 250 high-quality reference images with 40 types of distortion), City100 [34] (a real dataset that simulates camera lenses and includes 100 pairs of HR-LR image pairs taken using a Nikon camera and an iPhone, respectively), DPED [27] (composed of real photos taken with three different mobile phones and a high-end reflective camera, including various images of buildings, plants, roads, etc.), and T91 [30] (including local texture images of animals, plants, people, cars, etc.). These have also been used for image SR reconstruction, and have greatly expanded the numbers and types of datasets available, which is conducive to testing the generalization ability of various models. Figure 2 is a partial image display of the SR dataset.

Table 1 Introduction of SR Dataset

Dataset	Quantity (piece)	Download address	Source	Use
Set5 [13]	5	https://www.kaggle.com/msahebi/super-resolution	ACCV	train
Set14 [9]	14	https://www.kaggle.com/msahebi/super-resolution	ACCV	train
BSD100 [15]	100	https://www.kaggle.com/msahebi/super-resolution	IEEE	train
Urban100 [14]	100	https://drive.google.com/drive/folders/1pRmhEmmY-tPF7uH8DuVthfHoApZWJ1QU?usp=sharing	IEEE	train
Manga109 [16]	109	http://www.manga109.org/en/index.html	ICML	train
DIV2K [17]	1000	https://data.vision.ee.ethz.ch/cvl/DIV2K/	IEEE	train/validate
Flickr2K [21]	2650	https://drive.google.com/drive/folders/1B-uaxvV9qeuQ-t7MFiN1oEdA6dKnj2vW	CVPR	train
RealSR [25]	595 pairs	https://drive.google.com/open?id=17ZMjo-zwFouxnm_aFM6CUHBwgRrLZqIM	IEEE	train/validate
DPED [27]	5827	https://drive.google.com/file/d/0BwOLOmqkYj-jeUJwQjRNUFkzOTA/view	ICCV	train
OutdoorScene [28]	10624	https://drive.google.com/drive/u/1/folders/1iZfzAxAwOpeutz27HC56_y5RNqnsPPKr	CVPR	train/test
PIRM [29]	200	https://drive.google.com/drive/folders/17FmdXu5t8wlKwt8extb_nQAdjxUOrb1O?usp=sharing	ECCV	validate/test
T91 [30]	91	https://drive.google.com/drive/folders/1pRmhEmmY-tPF7uH8DuVthfHoApZWJ1QU?usp=sharing	IEEE	train

Fig. 2 Image display of the dataset section

3 Image super-resolution reconstruction based on deep learning

Based on the network model used, current algorithms in the field of deep learning-based SISR can be divided into three types: CNN-based models, GAN-based models, and transformer-based models. Figure 3 shows some classic algorithms based on these three models. Algorithms based on CNNs are the most widely used for SISR, and many researchers have achieved excellent results using models of this type. However, over time, the improvements in CNN networks have tended to be gradual. In recent years, researchers have therefore done a lot of work on transformer models, and these now produce better results than CNN models. In the following, we introduce the classic SISR algorithms of the three types mentioned above.

3.1 Super-resolution models based on convolutional neural networks

The CNN evolved from the multi-layer perceptron (MLP). Due to its structural characteristics, such as local area connections, weight sharing, and downsampling, CNNs perform well on the task of image processing. Many scholars have applied CNNs to SISR, and have achieved results that have exceeded those of traditional methods. Through the use of CNNs, the images obtained from SISR have shown significant improvements in terms of both the peak signal-to-noise (PSNR) and structural similarity index (SSIM) metrics. Over

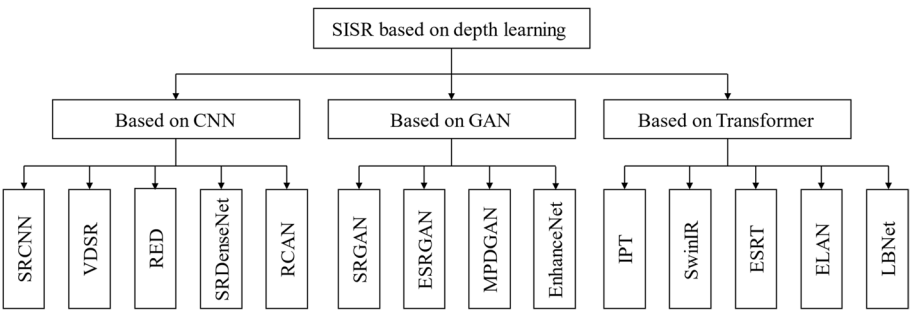


Fig. 3 Classification of SISR algorithm based on deep learning part

time, CNN networks have also been continuously developed, resulting in many variants based on basic CNN networks and driving progress in the field of SISR. At present, based on the different CNN networks used, models based on CNNs can be generally divided into convolutional direct connection models, residual network models, recursive network models, dense convolution network models, attention mechanism models and lightweight convolutional network models. These will be described separately in the following.

3.1.1 Convolutional direct connection models

Dong et al. [35] were the first to apply CNNs to SR in 2014, and proposed a super-resolution convolution neural network(SRCNN). Their approach was to downsample the collected HR images with a certain sampling factor to obtain LR images, and then to use the bicubic interpolation method to reconstruct the LR image into an image of the same size as the original input HR image; a three-layer CNN was then applied to obtain the output in the form of an HR image. The network structure of SRCNN is shown in Fig. 4. This technique represented a breakthrough in the field of image SR. Compared to traditional methods, it has the advantages of a simple model, high accuracy, and fast speeds, and at the time, the reconstruction quality was better than the alternatives. However, its front-end upsampling model framework suffered from problems such as computational complexity and slow training convergence. In addition, SRCNN also have certain shortcomings such as a simple structure and difficulty in fully utilizing the contextual information of the image.

In the same year, Dong et al. [36] proposed a fast super-resolution reconstruction convolution neural network (FSRCNN) to solve the problems inherent in SRCNN. As shown in Fig. 5, FSRCNN achieved some improvements compared to SRCNN. SRCNN enlarges an LR image into an HR image size after bicubic interpolation, meaning that the subsequent convolution operation is calculated based on the size of the HR image, thereby increasing the consumption of computing resources, causing low efficiency and an inability to meet real-time requirements. The FSRCNN uses a deconvolution layer at the end to enlarge the size; this means that the original LR image can be directly input into the network, and the convolution can be calculated based on the size of the LR image, thus greatly reducing time consumption. To reduce the computational complexity of the mapping layer in SRCNN, the number of parameters in FSRCNN is reduced by adding a contraction layer

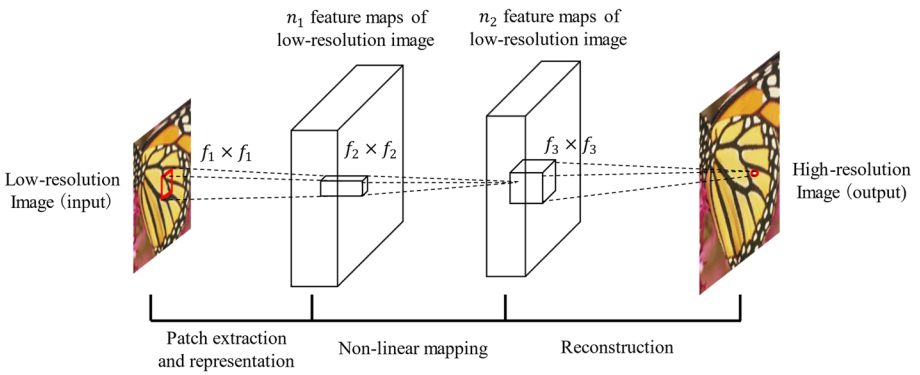


Fig. 4 Architecture of SRCNN

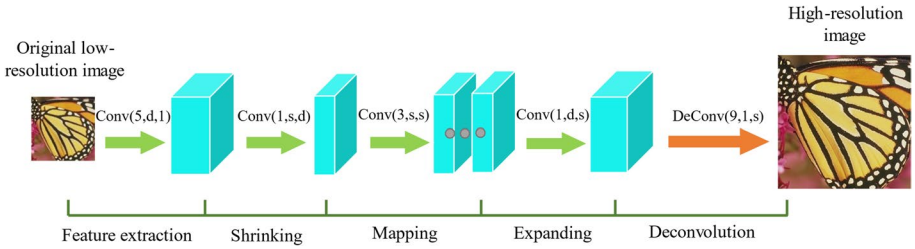


Fig. 5 Architecture of FSRCNN

before the mapping layer, giving fewer filters but more layers, with a narrower but longer network structure, thereby accelerating the reconstruction speed of the SR network.

Due to the need for SRCNN to upsample LR images to obtain HR images before inputting them into the network, convolution operations are performed at higher resolution, which increases the computational complexity. For this reason, Shi et al. [37] proposed the efficient subpixel convolutional neural network (ESPCN) model, as shown in Fig. 6, in 2016. The core idea of this approach is to use subpixel convolutional layers rather than deconvolution layers to achieve upsampling operations. The input to the network is a LR image, from which features are extracted in low-dimensional space. Then, through three convolutional layers, a feature map with a channel number of r^2 is obtained, where r is the upsampling magnification. The r^2 channels for each pixel in the feature image are then rearranged into a region of size $r \times r$, corresponding to a sub-block in the HR image. The feature map with size $H \times W \times r^2$ is therefore rearranged into a HR image of $rH \times rW \times 1$. In this network, the interpolation function used for the image size enlargement process is implicitly included in the previous convolutional layer, and can be automatically learned. Convolutional operations are performed on LR images with high efficiency.

Although early CNN-based models such as SRCNN, FSRCNN, and ESPCN only used stacked convolutional layers, and had a relatively simple network structure without complex network strategies, their reconstruction performance was improved compared to traditional image SR methods; they have therefore played a pioneering role in the development of deep learning-based image SR.

3.1.2 Residual network models

Since a few simple layers of convolutional networks were initially found to achieve good results in terms of extracting more image features and giving high performance, researchers have since focused on increasing the depth of the network, which has indeed

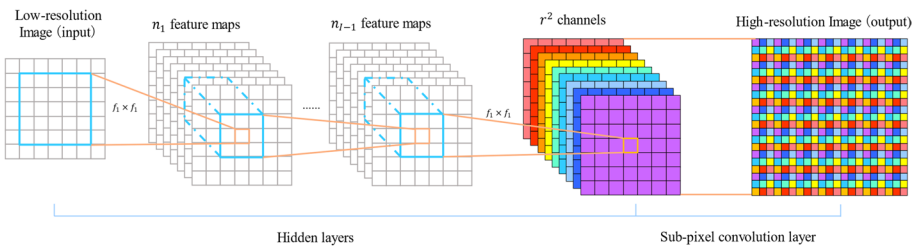


Fig. 6 Architecture of ESPCN

improved the performance of these models. However, simply deepening and widening the original network leads to problems such as gradient vanishing, gradient explosion, and network degradation. The solution to the gradient problem is batch normalization (BN) or regularization, but the issue of degradation persists. In response, He et al. [38] proposed the residual network (ResNet) in 2015 to solve the problems of gradient and degradation caused by deep networks. Its structure is shown in Fig. 9(a), and involves the addition of quick connections or skip connections to the network, thus allowing an ordinary network to become a corresponding residual network through residual learning.

Kim et al. [39] were inspired by VGG-Net [40] to apply residual networks to image SR, and proposed the very deep convolutional network for super-resolution (VDSR) model in 2016. The structure of this network is shown in Fig. 7. VDSR uses 20 convolutional layers, and effectively exploits contextual information on large image regions by cascading small filters multiple times in a deep network structure. VDSR performs a padding operation on the image before each convolution, thus ensuring that all feature maps and the final output image remain consistent in size, and thereby solving the problem in which the image becomes smaller and smaller through gradual convolution. Compared with SRCNN, VDSR has a deeper network structure, an expanded receptive field, makes full use of context information, and avoids the loss of image information. VDSR also applies adaptive gradient cropping, using adjustable gradients to maximize speed while suppressing gradient explosion. Images at different scales can be used for training, so that the trained model can solve the problem of SR at different scales.

In the same year, Mao et al. [41] proposed the residual encoder-decoder network (RED). The structure of this network is symmetric, with each convolutional layer having a corresponding deconvolution layer, as shown in Fig. 8. The convolutional layer is used to obtain the abstract content of the image, while the deconvolution layer is used to enlarge the feature size and restore the details of the image. The convolutional layer reduces the size of the input image, and upsampling is then carried out through the deconvolution layer to make the input and output sizes the same. Each set of images corresponds to a jumper connection structure between the convolutional layer and the deconvolution layer, where features of the same size are added before being input to the next deconvolution layer. This structure allows the backpropagation signal to be directly transmitted to the bottom layer, thereby solving the problem of gradient vanishing. At the same time, it can transmit the details of the convolutional layer to the deconvolution layer, which can restore a cleaner image.

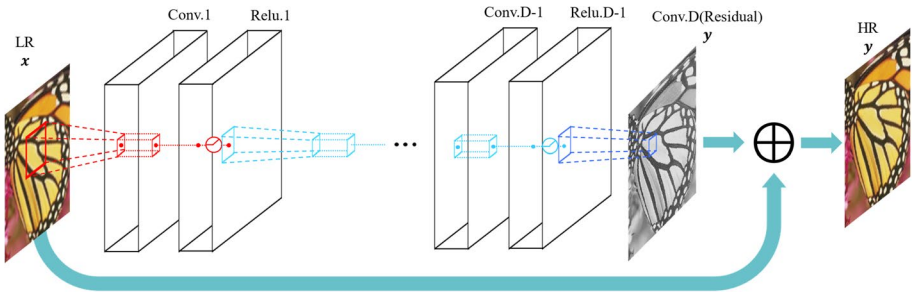


Fig. 7 Architecture of VDSR

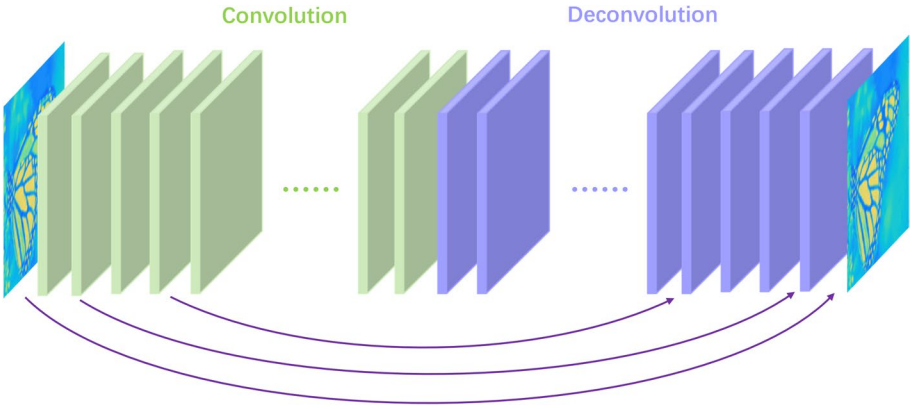


Fig. 8 Architecture of RED

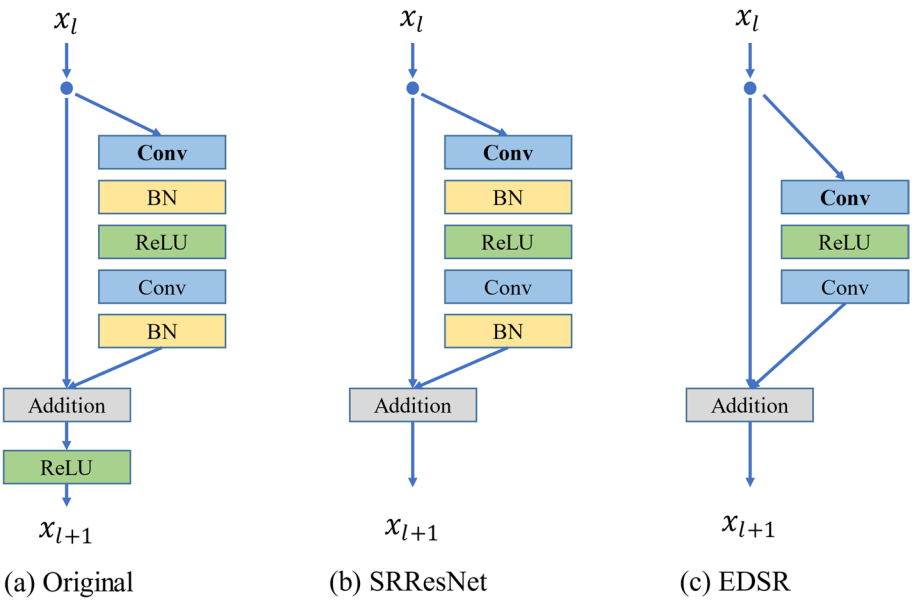


Fig. 9 Architecture of EDSR ((**a**) Architecture of ResNet (**b**) Architecture of SRResNet (**c**) Architecture of EDSR)

In 2017, Lim et al. [42] proposed the enhanced deep super-resolution network (EDSR), which won first prize in the NTIRE2017 Super-Resolution Challenge. The structure is shown in Fig. 9 (c). Based on work by Nah et al.'s work in [43], EDSR innovatively removed the batch normalization (BN) layer from the SRResNet [12]. Since the BN layer consumes the same amount of memory as the convolutional layer before it, removing this step means that EDSR can stack more network layers or extract more features from each layer for better performance while using the same computational resources. To stabilize the training process, the authors adopted residual scaling [44], which involved placing a

constant scaling layer after the last convolutional layer in each residual block. When a large number of filters are used, these modules greatly stabilize the training process. In [42], the authors simultaneously proposed a multi-scale super-resolution network (MDSR), which takes advantage of the inter-scale correlation made by VDSR [40] and can effectively handle SR at various scales within a unified framework, with performance comparable to that of single scale models.

Residual networks can not only effectively extract feature information from LR input images in SR network models through local or global residual learning, but can also solve many of the training and gradient problems that arise in deep networks. Hence, many SR network models apply the idea of residual learning to network structures to improve the performance.

3.1.3 Recurrent neural network models

The recurrent neural network (RNN) was proposed for natural language processing (NLP). The input to the recursive module is the current input and the output of the last recursion. After continuous iteration, the final output can be obtained. An RNN does not require multiple overlaps and hierarchical processing, as it mainly relies on memory data, and the information from each iteration is synchronized and shared with each layer. In the task of image SR, the purpose is to generate higher-quality images, and the details of the image are crucial, meaning that most CNNs do not use pooling layers, as these cause a loss of pixel information. However, with the deepening of the network, more parameters will be added, overfitting may occur, and the model may be too large to store and reproduce.

As a solution to the above problems, internal parameter sharing is used in RNNs, which allows the model to learn more features without adding too many parameters. In 2016, Kim et al. [45] applied an RNN to image SR and proposed the deep recursive convolutional network (DRCN). This network consists of embedding, inference, and reconstruction modules, which correspond to the feature extraction, nonlinear mapping, and reconstruction modules in SRCNN. A recursive layer is used in the inference network, as shown in Fig. 10, and each recursion uses the same convolutional kernel and ReLU activation. By using cores with a size larger than one, the range of the receptive field becomes increasingly large as recursion deepens. To prevent gradient explosion and vanishing, the author proposed the use of supervised recursive layers, each of which is supervised by the network. At the

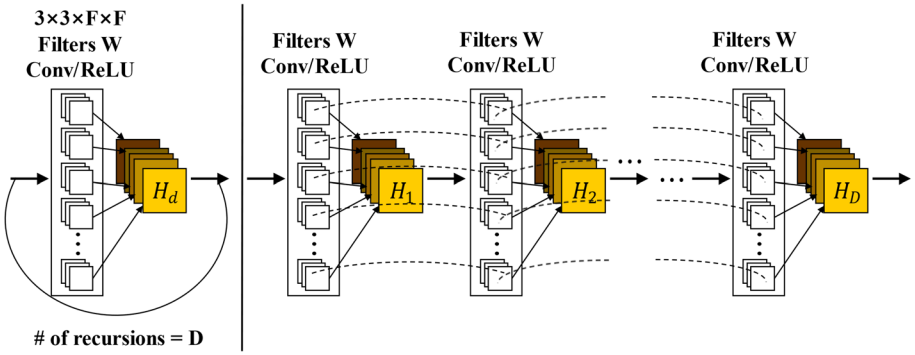


Fig. 10 Inference Network in DRCN (Left: Recursive Layer, Right: Expanded Structure)

same time, a skip connection structure is adopted, which not only reduces the computational power needed for long-distance information transmission, but also allows for more accurate reproduction of low-frequency (LF) structures. The use of skip-layer connections significantly improves the learning process, and the author therefore introduced the idea of using a single-layer network with 16 layers of recursion, showing that increasing the depth of recursion could improve the network performance. A simplified structure of a DRCN model is shown in Fig. 11(a).

In 2017, Tai et al. [46] proposed a deep recursive residual network (DRRN) based on DRCN, which adopted global and local residual structures. Local residual learning was introduced to solve the problem of image degradation caused by detail loss, as it was performed between stacked layers. A simplified structure of this type of network is shown in Fig. 11(b). DRRN has a recursive block composed of several residual units, and the weight set is shared between these residual units, unlike in DRCN, where weights are shared between convolutional layers. DRRN also addresses the problem of gradient vanishing or exploding by including recursive blocks with multipath structures, while DRCN supervises each recursive block to address this issue. Hence, even if the number of network layers is increased to 52, the DRRN model can still be successfully trained and achieve good performance. However, although the results are better compared to the VDSR model with a depth of 20 layers, the improvement is not significant. Compared to LapSRN [11], developed in the same period, the performance of the network needed to be optimized, and the use of only two convolutional layers per recursive unit was somewhat insufficient.

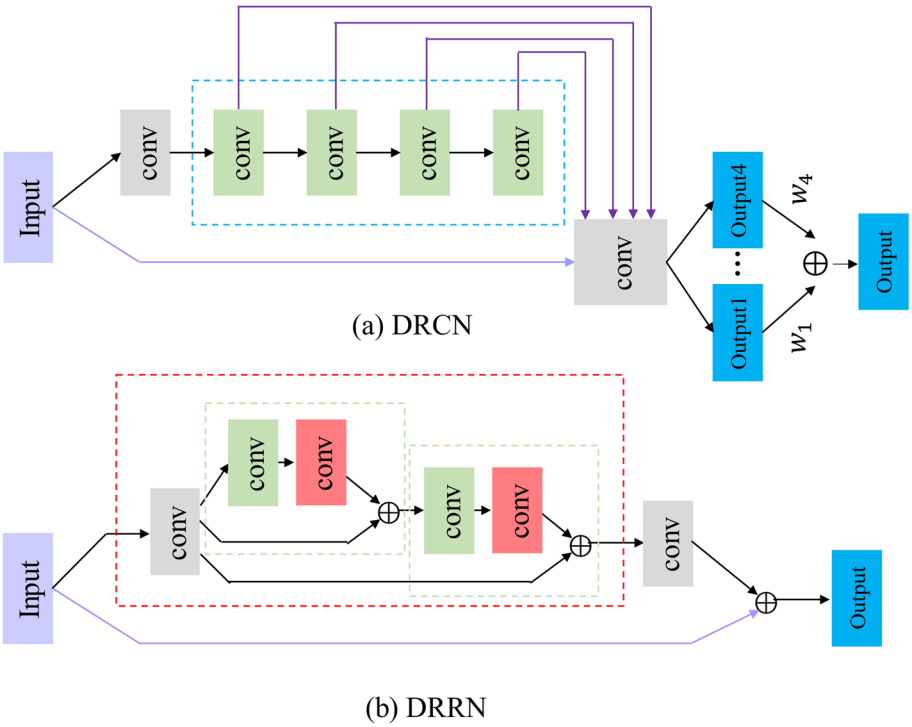


Fig. 11 Network Structure of DRCN (**a**) and DRRN (**b**)

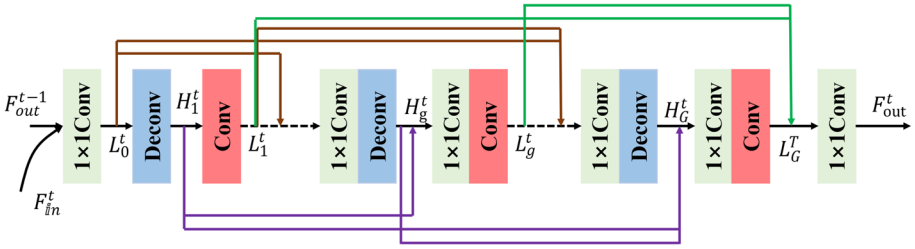
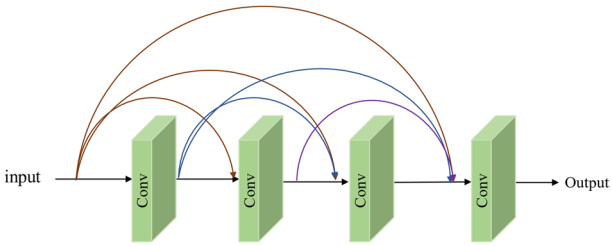


Fig. 12 Feedback block structure in SRFBN

Fig. 13 Architecture of DenseNet



The two RNN networks described above are similar to most traditional deep learning-based methods and can be extrapolated to a single-state RNN [47]. These network structures can share information in a feedforward manner, but in this approach, the upper layers cannot obtain useful information from the lower layers. In some well-known theories, feedback connections connecting the visual regions of the cortex can transmit response signals from higher-order regions to lower-order ones [48, 49]. Inspired by this idea, Li et al. [50] introduced a feedback mechanism into an RNN to create the image super-resolution feedback network (SRFBN) in 2019. Through the use of feedback connections, higher-order information is used to refine lower-order information and to obtain clearer reconstructed images. The core of this network is the feedback block (FB), as shown in Fig. 12, which includes multiple sets of up- and downsampling layer constructions with dense skip connections. In addition, to better adapt the network to complex tasks, the author also proposed a new training strategy that took HR images with increased reconstruction difficulty as the target input to the network for continuous iteration. This strategy enabled the network to further learn complex degraded models.

3.1.4 Dense convolutional models

In deep learning networks, as the number of layers increases, the gradient signal becomes more likely to disappear during the training process after being passed through many layers. Many papers have proposed solutions to this problem, such as ResNet [38], highway networks [51], stochastic depth [52], and FractalNets [53], among others. The solution common to these networks is to create connections between the previous and subsequent layers. Based on this idea, in 2017, Huang et al. [51] designed a new connection mode and developed DenseNet, in which all layers are directly connected, while maximum information transmission is ensured between layers in the network. To create feedforward characteristics, each layer concatenates the inputs of all previous layers and then transmits the output feature maps to all subsequent layers. The network structure of DenseNet is shown

in Fig. 13. The input to each layer in the network consists of the feature maps from all previous layers, and the feature maps from that layer are included as input to all subsequent layers.

In 2017, Tong et al. [52] first applied DenseNet to SR and proposed the super- resolution dense network (SRDenseNet), whose structure is shown in Fig. 14. This network includes a convolutional layer for extracting low-level features, a DenseNet block for learning high-level features, a deconvolution layer, and a reconstruction module. The low- and high-level features are effectively fused through a dense jump connection, and a deconvolution layer is then used to further enhance the details of the reconstructed image, improve the information flow, and alleviate the problem of gradient disappearance.

In the same year, Tai et al. [53] proposed the deep persistent memory network (MemNet). Its structure is shown in Fig. 15, and is different from the one-way propagation structure of traditional neural networks, as this is a long-term memory model. The network is composed of a feature extraction network, multiple stacked memory blocks, and a reconstruction network. The most important module is the memory block, which consists of recursive units for simulating nonlinear functions and gate units for adaptive learning of different memory weights. It excavates persistent memory through an adaptive learning process, thereby constructing long-term dependencies of deep networks.

3.1.5 Attention mechanism models

The constructed image is likely to have blurred artifacts at the edges, and attention mechanisms have therefore been introduced into the field of image SR. These assign different

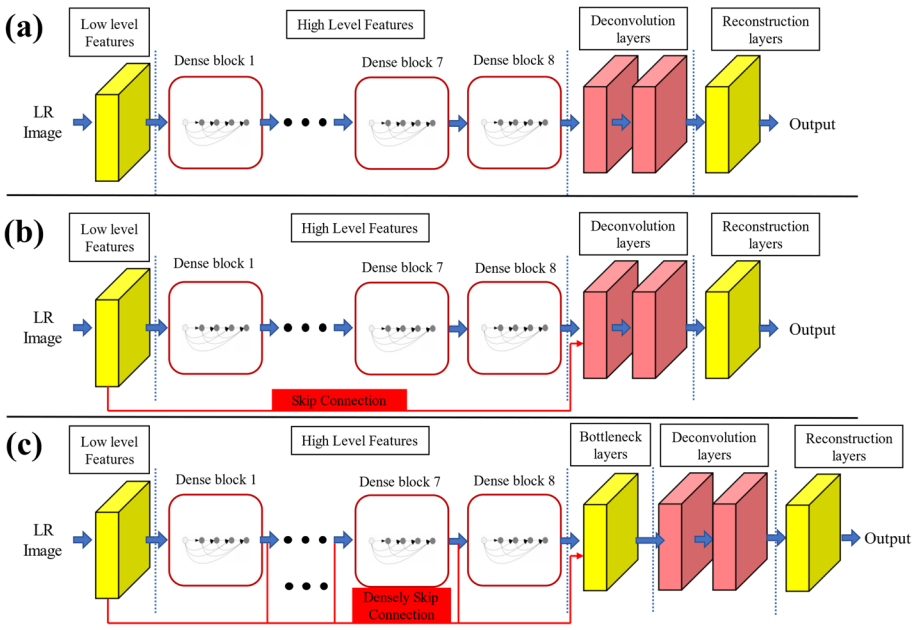


Fig. 14 Network Structure of SRDenseNet ((**a**) SRDenseNet_H: Use only advanced feature maps as input. (**b**) SRDenseNet_HL: Combining low-level and high-level features as input. (**c**) SRDenseNet_All: Combine all levels of features by skipping connections as input)

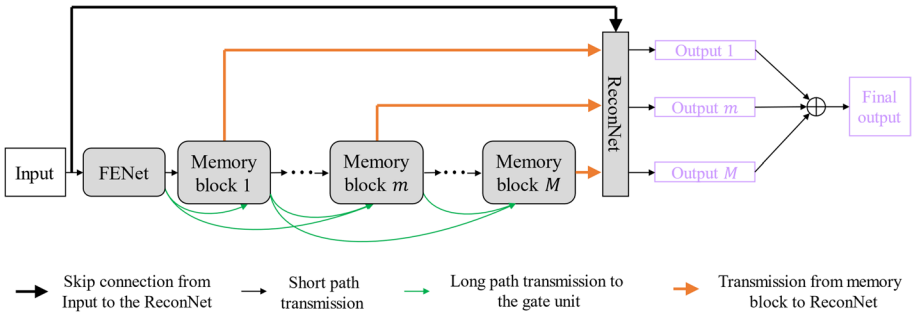


Fig. 15 Architecture of MemNet

weights to each part of the image features based on their importance, meaning that the network focuses more on learning important, highly weighted information, and ignores irrelevant information with lower weight, thereby improving the details of the image and improving the reconstruction quality. These models have good flexibility and robustness [54]. Current mainstream attention mechanisms include channel attention, spatial attention, and self-attention.

In 2018, Hu et al. [55] proposed the squeeze and extraction network (SENet) to introduce a channel attention mechanism into deep neural networks. A channel corresponds to a feature of the image. SENet is divided into two steps: squeezing and excitation. Firstly, the features of each channel are squeezed as descriptors for that channel. Then, the relationship between channels is captured through excitation, and the interdependence between channels is explicitly modeled to improve the network's feature learning ability. The structure of this network is shown in Fig. 16.

The CNN-based network model described above treats the rich LF information contained in LR input images equally in the channels, which affects the network's representation ability. Zhang et al. [56] were the first to apply an attention mechanism to SR in 2018. They developed the residual channel attention network (RCAN), which consists of four main parts: shallow feature extraction, residual-in-residual (RIR) deep feature extraction, an upsampling module, and a reconstruction module. The RIR module includes multiple residual groups and long-hop connections, and each residual group also includes multiple residual channel attention blocks with short-hop connections. Jump connections help to transmit LF information, and enable the main network to learn more effective information. A network with this structure may have a depth of over 400 layers. The channel attention mechanism can adaptively adjust the features of each channel based on the dependency relationship between channels, thereby allowing the model to learn more useful channel features and improving the network's representation

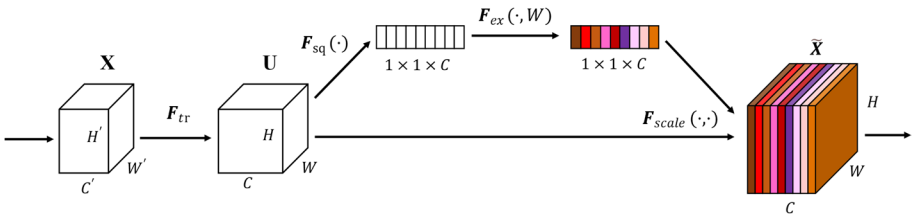


Fig. 16 Architecture of SENet

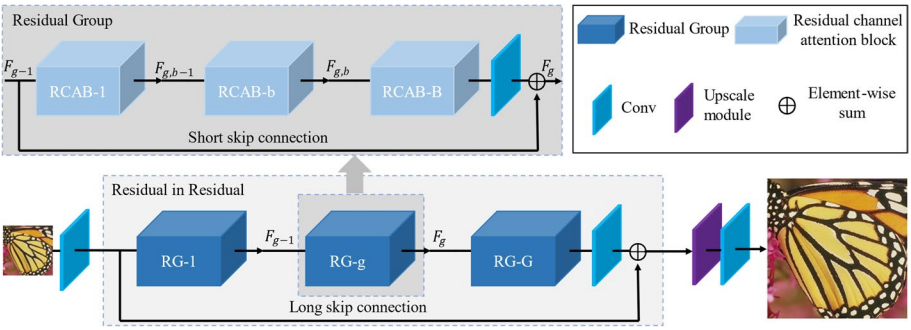


Fig. 17 Architecture of RCAN

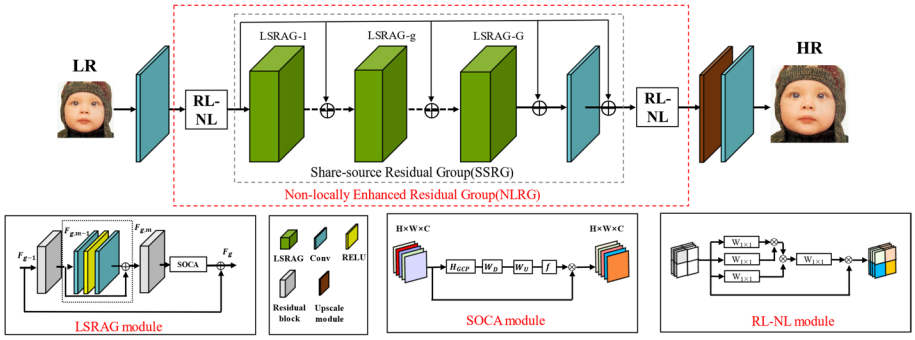


Fig. 18 Architecture of SAN

ability. Overall, RCAN was able to achieve the best reconstruction results in terms of both PSNR and SSIM at the time. The network structure can be seen in Fig. 17.

In 2019, Dai et al. noted that most CNN-based network models did not fully utilize the information of the original LR images and focused on instead on designing deeper network models to learn advanced features; they rarely utilized the feature correlation of the middle layer, which hindered the representation ability of a CNN. The attention mechanism proposed for SENet uses the first-order statistics of features through global average pooling but ignores the higher-order statistics, which reduces the network's discriminative ability. However, in the field of image SR, the more HF information, the better the reconstructed image. Hence, Dai et al. [57] studied the second-order statistics of features and proposed a deep second-order attention network (SAN). The most noteworthy aspect of the network was that it was based on non-local enhanced residual group (NLRG) and second-order channel attention (SOCA) mechanisms. The structure of this network is shown in Fig. 18. NLRG combines non-local operations to obtain potential contextual information, as well as shared residual group structures to learn deep features. Unlike SENet, which uses first-order statistics, SOCA learns the correlations between features through global covariance pooling to obtain the second-order statistics of features. This enables the network to focus on more specific features and improves the discriminative learning ability, giving rise to stronger feature-related learning and feature expression capabilities. Due to the use of the SOCA mechanism,

the SAN network model performs better on images with higher-order information, such as textures.

Since the convolutional layer of a CNN is designed to extract local features, most CNN-based SR networks lack the ability to model global contextual information. Although an SAN can recover several local textures, the directions of these textures may differ from those of the real images, as it combines global contextual information with local features. In 2021, Zhang et al. [58] therefore proposed a context reasoning attention network (CRAN), which can adaptively adjust the convolutional kernel based on the global context, enhanced by semantic inference. This model adopts the network structure of RCAN, except that the original RCAB in the network is replaced with CRAB, which includes the context reasoning attention convolution (CRAC) proposed by the author. As shown in Fig. 19, CRAC carries out context information extraction, inference of the context descriptor relationships, channel interaction, and spatial interaction. More specifically, the contextual information on the input features is first extracted through the fully connected layer, followed by CDRR, which is obtained by applying graph convolution. The output is decomposed into two tensors, which undergo channel interaction and spatial interaction, respectively. Finally, the final annotation mask is formed through contextual inference, and the convolutional kernel is adaptively modified using this mask. This network has achieved excellent results under different degradation models, and a good balance can be achieved between performance and model complexity.

After the above work, the attention mechanism has been proven to be effective in SR networks. The PAN model [59] has achieved good performance through the use of pixel attention, with a significant reduction in the number of parameters. Pixel attention is a more common form of operation than channel or spatial attention. Inspired by this, Zhou et al. [22] proposed an efficient image SR method called VapSR (VAst receptive field Pixel attention network) in 2022, which introduces the large receptive field design into the attention mechanism. Multiple sets of controlled experiments have proven that the introduction of a large core convolution can improve the network performance, but this involves numerous additional parameters, and depth separable convolution is therefore used to segment dense large convolution cores. This operation uses small normal convolution cores and small hole convolutions to achieve the receptive field of large core convolution, to reduce the number of network parameters. However, due to the use of element-by-element multiplication in the attention mechanism, training of the network becomes unstable. Hence, the author proposed a pixel normalization method to normalize the shifted layer distribution

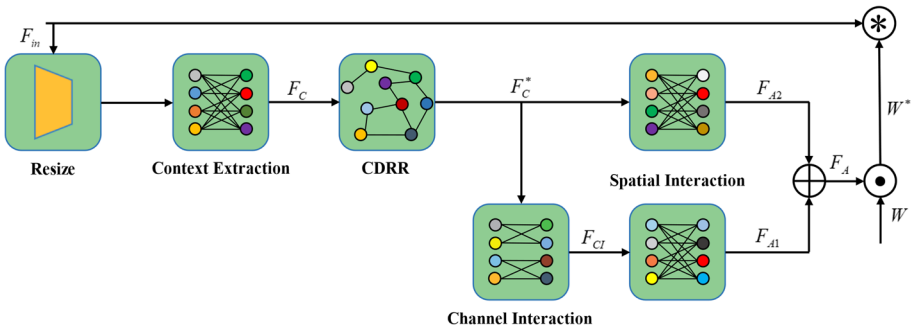


Fig. 19 Structural schematic diagram of CRAC

to give a standard normal distribution. After a series of operations, this network model can achieve good performance with a minimal number parameters compared to using large kernel convolution alone. The structure of this network can be seen in Fig. 20.

3.1.6 Lightweight convolutional network models

Following recent developments in this area, researchers are paying more attention to the practical application and performance of deep neural networks, in the hope that neural network models can run in real time in real scenarios (such as on mobile devices, embedded devices, etc.). However, due to the limitations on memory resources, low processor performance, and limited power consumption of these platforms, it is difficult to deploy the highest precision models and to achieve real-time operation on these platforms. To enable neural network models to run on mobile and embedded devices, the development of lightweight network models has become a research hotspot in the field of artificial intelligence in recent years.

The core idea of lightweight networks is to design a more compact network structure or to use lightweight strategies with the original network, to reduce the number of network parameters, improve the network speed, and achieve a lightweight transformation from both volume and speed while maintaining the accuracy as far as possible. At present, the most commonly used strategies for creating lightweight networks are to replace traditional convolutions with lightweight convolution methods (such as deep separable convolution, group convolution, or extended convolution), and to use global pooling to replace fully connected layers to reduce the number of network parameters. Other strategies such as knowledge distillation, network pruning, quantization, low-rank decomposition, and adaptive reasoning have also been widely applied to construct lightweight networks.

To address the challenges of computational complexity and memory consumption faced by CNN-based SR methods in practice, Zheng et al. [60] proposed an information distillation network (IDN) with lightweight parameters and low computational complexity in 2018, as shown in Fig. 21. The network consists of three parts: feature extraction blocks (FBlocks), multiple stacked information distillation blocks (DBlocks), and reconstruction modules (RBlocks). The information distillation block, composed of an enhancement unit and a compression unit, is the core structure of the IDN network, and can gradually extract rich and effective image features. Its structure can be seen in Fig. 22(a). The enhancement unit applies a channel separation strategy to retain local information and to process subsequent information, and is mainly used to enhance the contour areas of LR input images, while the compression unit consists of a 1×1 convolutional layer, and is mainly used for dimensionality reduction and the extraction of relevant image information. This network has the advantages of fast execution speeds and significant improvements in time performance.

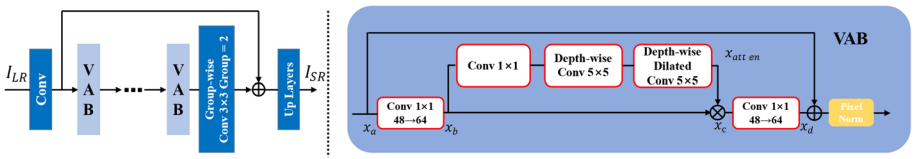


Fig. 20 Architecture of VapSR

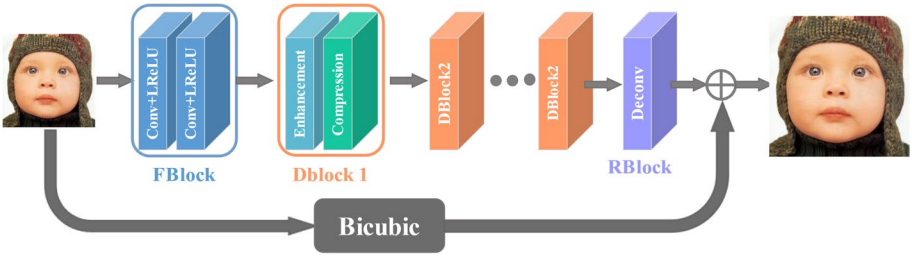


Fig. 21 Network Structure of IDN

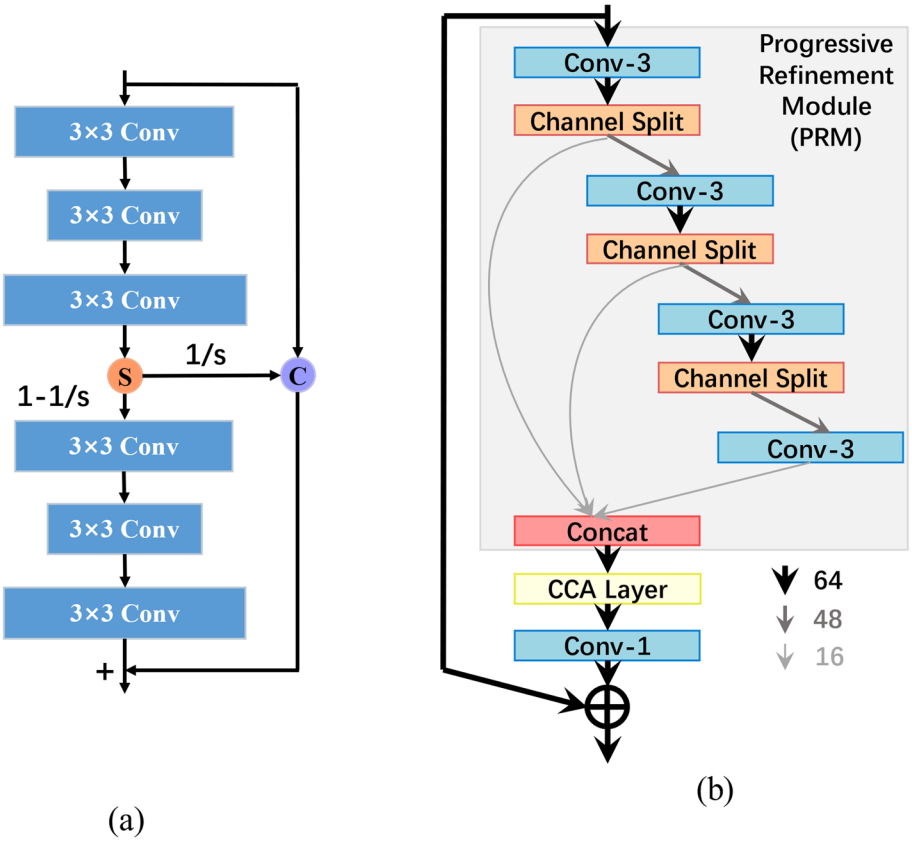


Fig. 22 (**a**) IDN enhancement unit (**b**) IMDB structure diagram in IMDN

Inspired by IDN, Hui et al. [61] improved the information distillation block in IDN in 2019, and proposed an information multi-distillation block (IMDB) for the construction of a lightweight information multi-distillation network (IMDN). As shown in Fig. 23, the network architecture also involves shallow feature extraction, deep feature extraction through the use of multiple stacked IMDBs, and upsampling modules. Figure 22(b) shows the IMDB, the core structure of this network, which can gradually extract more delicate and

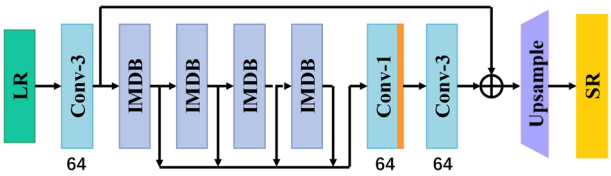


Fig. 23 Network Structure of IMDN

realistic image features. It consists of a progressive refinement module (PRM), a contrast perception channel attention layer (CCA), and a 1×1 convolution to reduce the number of feature channels. The PRM performs channel separation through convolutional layers, specifically preserving 75% for concat and separating 25% for backward propagation. The CCA calculates the mean and standard deviation of the feature map at each layer, and uses the sum of the two as contrast information. Through the use of these strategies, the IMDN performs well in terms of performance and inference time. In addition, the author devised an adaptive cropping strategy to solve the problem of SR at any scaling factor, which is crucial for the application of the SR algorithm in practical scenarios.

In 2020, Luo et al. [62] improved the residual block (RB) commonly used in SR and proposed a lattice block (LB), whose structure can be seen in Fig. 24. LB has the advantage of various linear combinations of two RBs (among which the combination coefficients are also called connection weights, calculated through attention mechanisms). LB can reduce the number of parameters by about half while maintaining similar SR performance. On this basis, Luo et al. proposed a lightweight network model called LatticeNet. As shown in Fig. 25, the network consists of four parts: shallow feature extraction, multiple cascaded LBs, a reverse fusion module, and an upsampling module. The core operation of the reverse fusion module is 1×1 convolutional and ReLU activation, the purpose of this which is to fuse hierarchical information that is very important for SR. The author of this scheme fused the output of each LB into BFM and adopted a reverse sequential cascade

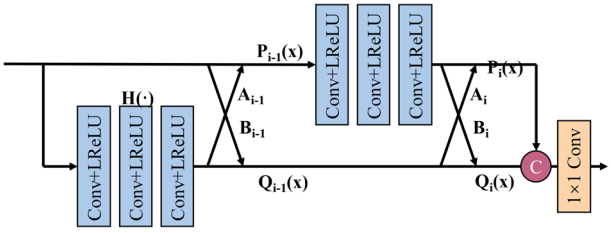


Fig. 24 Schematic diagram of LB structure

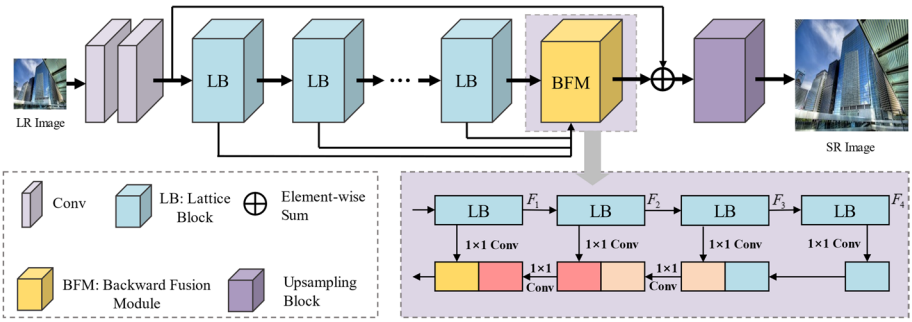


Fig. 25 Network Structure of LatticeNet

strategy to fuse different receptive field features. The network achieved a good balance between performance and parameters, and was best-performing lightweight network at the time. The proposed LB can also be applied in other SR networks using residual blocks.

Although lightweight networks have achieved good results, they are still not sufficiently lightweight. In addition, knowledge distillation typically requires a large amount of additional resources, whereas network pruning is a cheap and effective model compression method. Hence, in 2021, Zhang et al. [63] proposed a structural regularization pruning network (SRPN) for image SR. The author used L2 regularization to achieve structural regularization pruning (SRP), since L1 regularization makes it difficult to achieve an ideal balance between sparsity and performance by adjusting coefficients. For the constrained Conv layer, unimportant filters were randomly selected and their expressive power transferred to the rest of the network through a large number of training iterations. During the training process, the penalty strength was gradually increased, and pruning ended when the regularization coefficients of all unimportant filters reached a preset maximum value. The SRP model can be applied to high-performance SR algorithms in a plug-and-play manner. By applying SRP to the EDSR baseline network, the number of parameters can be reduced from 19.5 M to 609 K, thereby achieving a balance between performance and model size.

In the same year, Zhang et al. [64] proposed aligned structured sparse learning (ASSL), which is essentially a regularization-based filter pruning method. The authors introduced a weight normalization layer after each convolutional layer, and applied sparse-induced L2 regularization to the scale parameters at the weight normalization stage. At the same time, to solve the problem of a sparse structure after alignment across different layers, they proposed a new sparse structure alignment regularization term to encourage the pruning filter positions between different layers to be the same, thereby effectively pruning a large number of residual blocks. Finally, an efficient aligned structured sparse learning network (ASSLN) was trained using ASSL, which outperformed an SOTA lightweight image SR method in terms of both quantitative and visual results. Subsequently, Wang et al. [65] proposed global aligned structured sparse learning (GASSL), based on the two important components of Hessian-aided regulation (HAIR) and ASSL, as an upgrade and improvement to the ASSL method. The problem of hierarchical sparse allocation was solved through HAIR, as this approach can automatically allocate appropriate sparse ratios to different layers. A well-known proposition was presented in the article to prove the rationality of sparse allocation in HAIR. An illustration of the use of ASSL/GASSL to perform filter pruning on residual blocks can be seen in Fig. 26.

Tremendous progress has been made in the field of lightweight networks in recent years; these can reduce the numbers of network parameters, improve network speeds, and achieve more efficient SR networks while maintaining or improving network performance. This enables SR algorithms to be deployed in real-world applications. Although several

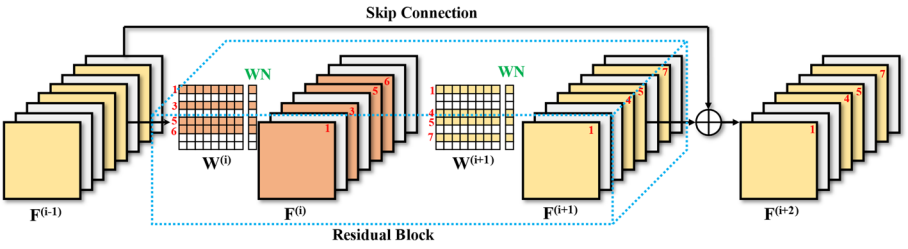


Fig. 26 Application of ASSL/GASSL method for pruning filters in typical residual blocks

excellent lightweight networks are emerging, most existing networks focus mainly on the numbers of parameters and floating-point operations (FLOPs). However, fewer FLOPs does not necessarily mean better network efficiency, and the number of network activations is in fact a more accurate measure of network efficiency. Hence, measurements of the efficiency of lightweight networks should not focus solely on parameter quantities and FLOPs, and they should instead be comprehensively analyzed from various perspectives.

3.2 Super-resolution models based on generative adversarial networks

Initially, the goal of the SISR task was to minimize the MSE between the generated and original images. This approach can maximize the PSNR, which is the standard measure of SISR. However, due to the average effect of MSE, models that use MSE losses are difficult to train on complex texture regions. The MSE loss is not always proportional to the quality perceived by a human, and may not be sufficient to accurately measure the quality. To pay more attention to the visual quality of generated images, a perceptual loss closer to perceptual quality was proposed. Although images measured in this way score lower on standard quantitative indicators such as PSNR and SSIM, they are visually more convincing.

In 2014, Goodfellow et al. [66] proposed the GAN, in which the two important components are generators and discriminators. By learning data, the generator can generate new images based on the learned data, with the aim of deceiving the discriminator; in turn, the discriminator determines whether the data provided by the generator are real or machine-generated. Through continuous gaming between the generator and discriminator, the abilities of both are continuously improved. When the generator and discriminator are balanced, the GAN model training is complete. A GAN-based method adopts the structure of a GAN, and the network is trained in a perceptually driven manner, with the aim of achieving better perceptual quality and more realistic visual effects in the reconstructed image. However, this method performs poorly based on objective evaluation indicators, and gives errors when restoring the details of an image.

In 2017, Ledig et al. [12] first applied a GAN to the task of image SR and proposed SRGAN, based on a GAN approach. The network structure is shown in Fig. 27. To obtain textural details that were more in line with human perceptions, the network used a GAN to train SRResNet, which formed the generator part of the network. The reason why GAN networks are used for training is that, as mentioned earlier, the MSE loss is commonly used in the current SR field to maximize PSNR; however, the recovered images often lose HF details, so the highest PSNR cannot represent the best visual results. In view of this, the author used a content loss based on VGG [40], defined the loss function with the ReLU activation layer of the 19-layer VGG network based on pre-training, and combined the content and confrontation losses to give a new perception loss function to improve the authenticity of the image. The training of this algorithm involves inputting LR image samples into the generator network to generate HR images, and then using the discriminator network to distinguish whether the input HR images are original, real, or generated HR images. When the discriminator cannot determine the authenticity of the images, this indicates that the generator network has generated high-quality HR images. Although the PSNR values obtained by SRGAN were not very high, it generated visually clear images; the authors therefore also used mean opinion scores (MOSs) to support their results.

Although the visual effects produced by SRGAN are impressive, as the network deepens, BN may lead to the appearance of artifacts. Hence, in 2018, Wang et al. [67] improved SRGAN and proposed the Enhanced SRGAN(ESRGAN). As shown in Fig. 28, ESRGAN

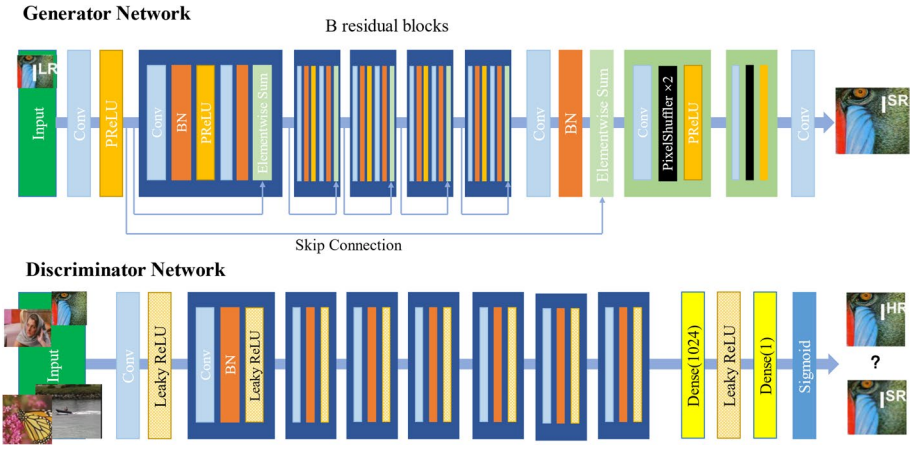


Fig. 27 Architecture of SRGAN

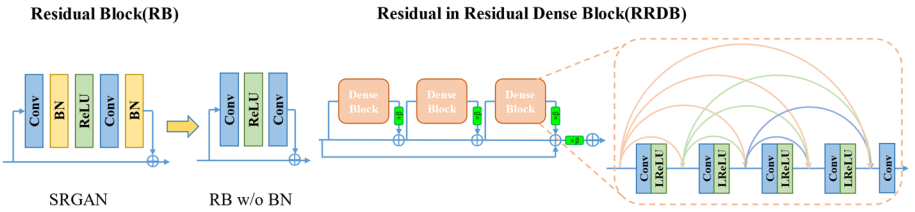


Fig. 28 RRDB structure used in ESRGAN

does not have the BN layer of SRGAN, thus reducing the computational complexity and saving storage resources to a certain extent. The residual-in-residual dense block (RRDB) is used as the basic unit of the network, which combines the ideas of multi-level residual networks and dense connections. For the discriminator, the author adopted the concept of RaGAN [68], but changed the absolute discriminator to a relative one; that is, when the discriminator receives an image from the generator, it learns to evaluate the probability based on the idea that a real image is more real than a fake image, rather than deciding whether an image is real or a fake one created by the generator. ESRGAN uses the features before the VGG activation stage to calculate the perceptual loss, while SRGAN uses the features before activation to calculate the perceptual loss. This approach overcomes two drawbacks: (i) the activated features are sparser, whereas using pre-activated features for the perceptual loss will result in more information; (ii) if the activated features are used as input, the reconstructed image may have inconsistent levels of brightness compared to the real image.

Although GAN-based image SR has successfully improved image quality based on subjective perceptions, it can also lead to checkerboard artifacts and unpleasant HF components. In 2019, Lee et al. [69] proposed a multi-perspective discriminator-based generative adversarial network (MPDGAN), which used various perspective discriminators to distinguish between real and fake images, in order to reduce SR artifacts and noise. The framework can be seen in Fig. 29. MPDGAN uses three discriminators to reduce the occurrence of checkerboard artifacts and unclear high-frequency components. The first discriminator

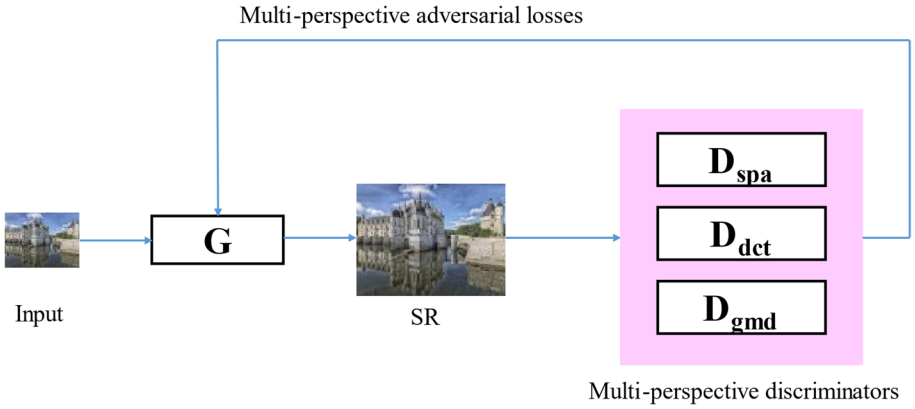


Fig. 29 Architecture of MPDGAN

is D_{spa}, which is the same as that used in the existing SRGAN and can distinguish between real and fake images in the spatial domain. The second discriminator, D_{dct}, distinguishes between real and fake images by checking the DCT coefficients. The purpose of this discriminator is to remove artifacts from the chessboard. Finally, the third discriminator D_{gmd} checks the gradient amplitude distribution of the image, and attempts to recover the natural HF components by comparing the gradient amplitude distribution (GMD) output by the generator with the real GMD. MPDGAN achieves a higher visual quality by reducing the number of artifacts compared to existing GAN-based SR methods, through the addition of frequency and gradient discriminators.

Although the images generated by ESRGAN have high visual quality, there is still room for improvement. In 2020, Nathanaël et al. [70] proposed ESRGAN+, based on ESRGAN, with the aim of further improving the perceptual quality of images generated by ESRGAN. ESRGAN+used a new block called a residual nested dense residual block (RRDRB), as shown in Fig. 30. The main improvement was the addition of additional residual learning to dense blocks, to increase the capacity of the network without increasing its complexity. As mentioned in reference [71], ResNet can reuse features, while DenseNet can discover new features. The visual quality of images generated using RRDRB is significantly better than that from simple dense blocks. Secondly, as described in [72], to add random details, noise is input into the architecture of the generator. Gaussian noise is added to the output

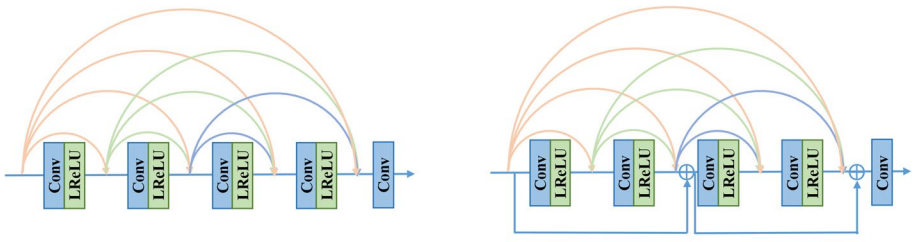


Fig. 30 Comparison between residual dense blocks and original dense blocks in ESRGAN+

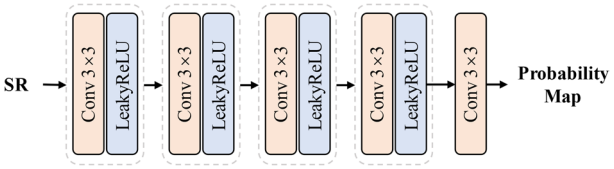


Fig. 31 Network of pixel-level adversarial discriminator

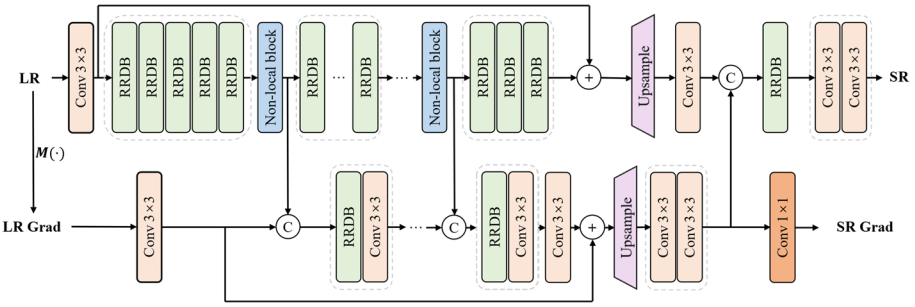


Fig. 32 Structure-aware deep network

of each remaining dense block and to each feature scaling factor learned, to enable the network to benefit from random changes. These improvements all contribute to generating images with more natural texture, clarity, and detail.

Although numerous GAN-based SR methods have emerged in recent years, and have achieved good visual results, recent studies have shown that GAN-based SR methods can cause structural distortion. Since it still uses image-level adversarial training, this approach cannot fundamentally solve the problem of structural loss caused by this type of training. Shi et al. [73], inspired by [74], proposed a pixel-level generative adversarial training method to solve this problem in 2023. This training strategy constrained the structure of the generated image by determining whether each generated pixel and each real pixel came from the same distribution. Adversarial training for pixels uses the neighborhood information, meaning that each generated pixel undergoes adversarial training. The SR results from this strategy were significantly better than those from PixelGANs [74]. The network structure of the pixel-level adversarial discriminator is shown in Fig. 31. In addition, inspired by [75, 76], the author proposed a gradient-guided structure-aware depth network in which the structure generation was enhanced through gradient guidance, and also effectively integrated nonlocal self-similarity modules at multiple levels. The structure of this network is shown in Fig. 32, and consists of two parts: the structure-aware gradient generation branch, and the structure-aware SR branch. The author named the proposed network PGAN, and showed that it yielded state-of-the-art performance on all five benchmark datasets and effectively alleviated the structural distortion problem of the GAN network.

Image SR algorithms based on GANs give improved visual perception quality, with some of these algorithms offering improvements in the image quality perceived with the human eye with almost no loss in terms of the PSNR and SSIM indicators, and this approach has driven developments in this field. However, a GAN involves a constant confrontation between a generator and a discriminator, resulting in unstable training processes, and this dual structure greatly increases computational costs and memory consumption. Hence, when GAN-based methods are used, strategies should be adopted to enhance the

ability of generators and discriminators to extract significant features, in order to further stabilize the training process, and lightweight GANs for SISR should be designed.

3.3 Super-resolution models based on transformer

A transformer is a network architecture for natural language processing (NLP),, and was proposed by Google's Vaswani et al. [77] in 2017. The authors abandoned the use of an RNN or CNN, and developed a transformation model entirely based on a self-attention mechanism to obtain the global dependency relationship between the input and output. Compared with a CNN, which expands the receptive field by stacking convolutional layers to obtain full-text information, RNN captures global connections by recursion; however, it is difficult to capture long-distance dependence using this approach. Self-attention mechanisms are effective in capturing global connections, solving the problem of long-distance dependence, supporting parallel computing, speeding up training, and improving network efficiency.

With the introduction of the Vision Transformer (ViT) [78], transformers gradually began to be applied in the field of computer vision, and achieved better results than CNNs, non-maximum suppression, and 3D convolution. In 2021, Chen et al. [79] proposed a pre-trained network model called an image processing transformer (IPT) for various low-level computer vision tasks, such as SR and denoising. The network mainly consisted of three parts: the heads for feature extraction, the encoder-decoder transformer for reconstructing lost information, and the tails for outputting the reconstructed images. A detailed diagram of this structure can be seen in Fig. 33. To adjust to different image processing tasks, the head and tail use a multi-head structure to handle each task separately. The encoder part of the encoder-decoder transformer is similar in structure to the original transformer, but the decoder is not quite the same: the embeddings of specific tasks are passed as additional inputs to the decoder, which is trained to decode the features of different tasks. To maximize the performance of the transformer architecture on various tasks, the author degraded each pair of original images from the ImageNet dataset into a series of corresponding images. When this data were used to train the IPT model, it was found that the model had a strong ability to capture inherent features for low-level image processing. After fine-tuning, the IPT can outperform existing techniques by using only one pre-trained model.

In the same year, the emergence of the swin transformer [80] created a new wave of research in the field of computer vision, as it solved the computational complexity problem of the transformer in a very elegant way. Inspired by this approach, Liang et al. [81] proposed a network called image restoration using the swin transformer (SwinIR) for image restoration. This network combined a CNN and a transformer, and its structure can be seen in Fig. 34. SwinIR consists of three main modules: shallow feature extraction, deep feature extraction, and image reconstruction. The shallow feature extraction module applies convolution operations to extract shallow features, and uses residual connections to directly transfer these shallow features to the reconstruction module to preserve the LF information of the image. The deep feature extraction module mainly consisted of multiple residual swin transformer blocks (RSTBs) and a convolutional layer for feature enhancement. Each RSTB used a swin transformer layer (STL) for local attention and cross-window interaction, whereas the image reconstruction module achieved high-quality image reconstruction by fusing the shallow and deep features. The SwinIR network model has the advantages of both a CNN and a transformer. Compared with mainstream CNN-based image restoration models, it not only exploits the ability of

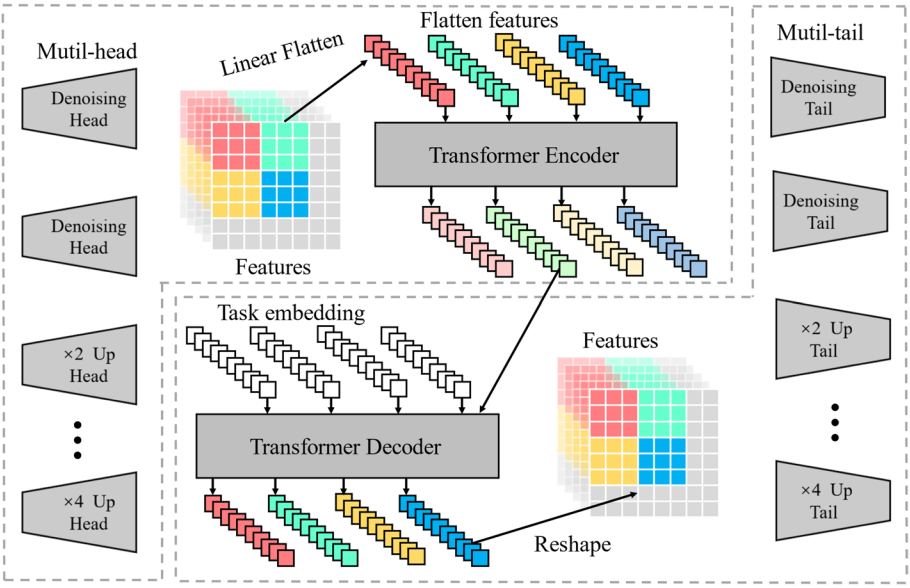


Fig. 33 Architecture of IPT

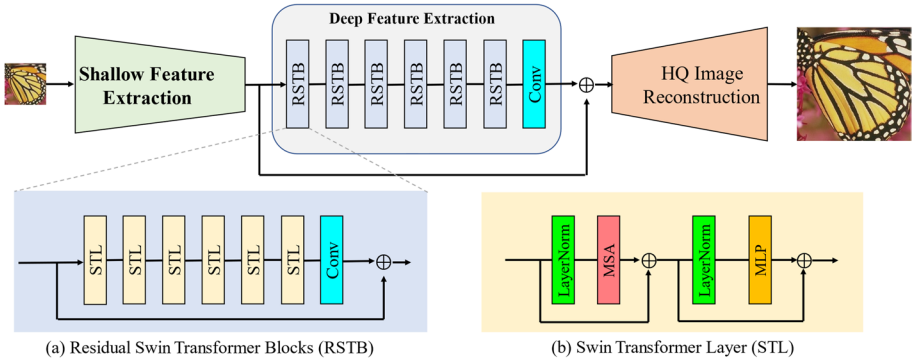


Fig. 34 Architecture of SwinIR

a CNN to focus on local features, but also utilizes the transformer to achieve long-term dependency modeling, while achieving better performance with fewer parameters. The visualization results indicate that SwinIR can remove severe noise interference, restore HF details, and reduce blurred artifacts, resulting in clearer edges and more natural textures. Numerous experiments have shown that SwinIR achieves state-of-the-art performance on classic image SR, lightweight image SR, real image SR, grayscale image denoising, color image denoising, and JPEG compression artifact reduction, thereby demonstrating its effectiveness and scalability.

Although the introduction of the transformer made significant breakthroughs in the field of SISR, large amounts of training data and GPU memory are required to train

models, meaning that they are not suitable for practical applications. Hence, Lu et al. [82] proposed a lightweight transformer model (ESRT) for SISR tasks in 2022. This model consisted of a lightweight CNN backbone (LCB) and a lightweight transformer backbone (LTB). The LCB could extract deep image features at a lower computational cost by dynamically adjusting the size of the mapping map, while the LTB was mainly used to obtain the long-term dependencies of similar blocks in the image. The efficient transformer (ET) module in the LTB was an improvement on the traditional multi-head attention (MHA), and the authors also proposed an efficient multi-head attention (EMHA) algorithm, as shown in Fig. 35. In the original MHA, Q, K, and V are directly used to calculate self-attention through large-scale matrix multiplications, which require huge amounts of memory. EMHA uses a feature segmentation (FS) module to segment Q, K, and V into s equal segments using a segmentation factor s, as the predicted pixels in images SR typically only depend on the local adjacent regions in LR. Assuming Q and K calculate the self-attention matrix with a shape of $B \times m \times N \times N$. The third and fourth dimensions of the last self-matrix are $N/s \times N/s$, meaning that this approach can significantly reduce the computation and GPU memory costs. Through these improvements, ESRT can effectively enhance the feature expression ability and long-term dependencies of similar blocks in images, thereby achieving better performance and verifying the feasibility of the transformer for lightweight SR tasks.

Although SwinIR has achieved impressive results on image SR, its network structure relies mainly on the swin transformer, which is designed specifically for advanced visual tasks. The network design of SwinIR is therefore redundant for SR problems, and self-attention is calculated based on a fixed-size window. Zhang et al. [83] proposed an effective long-range attention network (ELAN) in 2022, which consists of three parts: shallow feature extraction, deep feature extraction, and HR image reconstruction. The shallow feature extraction module is composed of a 3×3 convolution, while

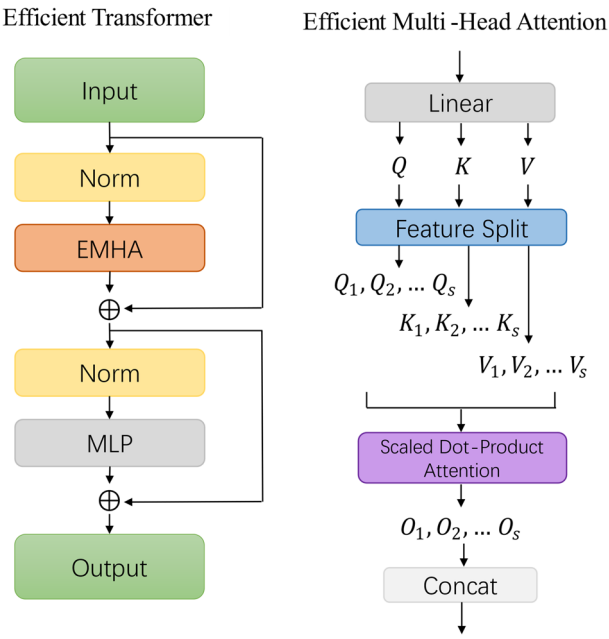


Fig. 35 Architecture of EF

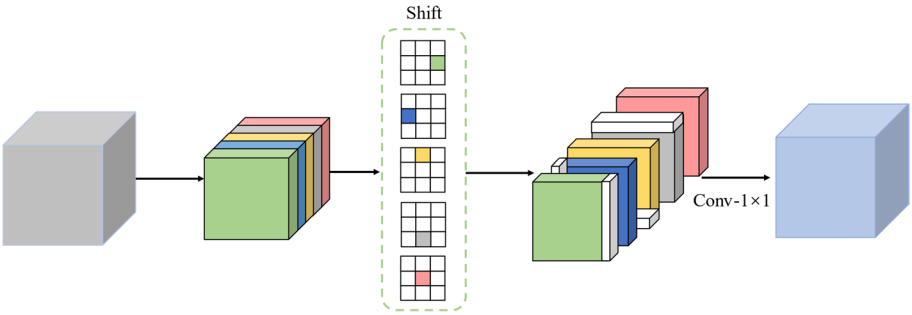


Fig. 36 Shift-Convolution

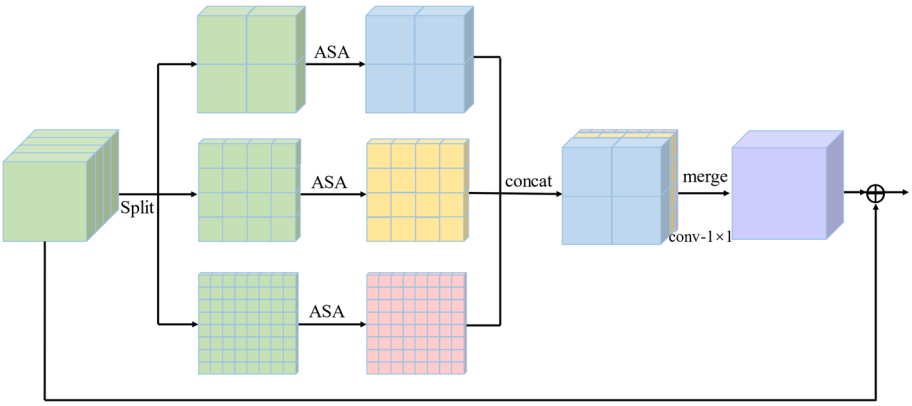


Fig. 37 Group-wise multi-scale self-attention

the depth feature extraction module is composed of stacked effective remote attention blocks (ELAB) and a residual connection, and the reconstruction module is composed of a 3×3 convolution and a PixelShuffle operation. Each ELAB consists of local feature extraction and group-wise multi-scale self-attention (GMSA) modules, both of which are equipped with residual connections. The local feature extraction module adopts the shift convolution operation, which does not introduce additional learnable parameters and has a complexity similar to that of a 1×1 convolution while causing the receptive field to change from one to three. GMSA divides the input features into K groups, each of which calculates the self-attention using different window sizes, and the SA outputs from the different groups are then aggregated using a 1×1 convolution. Diagrams showing the shift convolution and GMAS structure are shown in Figs. 36 and 37. The authors also abandoned layer normalization and adopted BN to stabilize the training process, while sharing attention scores between adjacent modules, thereby saving considerable resources. The researchers improved the shift window of SwinIR and proposed a cyclic shift mechanism in which the masking strategy and relative position encoding used in SwinIR were removed, making the network cleaner and more efficient. Thanks to these improvements, the ELAN-light model not only achieved good metrics on all five datasets, but also was 4.5 times faster than SwinIR-light, with fewer parameters and lower computational complexity.

In summary, it can be seen that transformers have played an important role in the field of CV. Both the simple transformer structure and the CNN + transformer structure have achieved better results than pure CNN networks. However, the widespread application of transformers in the CV field is still hindered, as transformer models are usually huge and have high computational costs. Although the works described above have made significant contributions to the development of lightweight transformers, there is still a long way to go to create practical lightweight transformer network models.

4 Quality evaluation of image super-resolution

4.1 Evaluation indicators

To evaluate the results of image SR, many indicators for evaluating the quality of the image have been introduced. Based on the different measurement methods used, they can be divided into two types: objective and subjective evaluation. In an objective evaluation, also known as quantitative evaluation, a specific result is obtained by calculating numerical values, which directly reflect the quality of the image and allow the quality of the results to be judged directly from the data. In a subjective evaluation, also known as qualitative evaluation, the reconstructed image is presented to invited evaluators, who are asked to rate the quality of the image. The latter approach can reflect the actual results more accurately than objective methods, and is more in line with practical applications; however, it requires a great deal of manpower and material resources, and the quality of the results is strongly dependent on human self-perception. Hence, objective evaluation is still the most important evaluation indicator in the field of image SR reconstruction. An overview of these two types of evaluation indicators is given below.

4.1.1 Objective evaluation indicators

(1) Peak signal-to-noise ratio

PSNR is the most widely used evaluation metric in the field of image SR, and is defined by the maximum pixel value and the mean square error between two images. The MSE is calculated as shown in Eq. (10). In the task of SR reconstruction, PSNR is used to detect the similarity between the reconstructed image and the real image, using the expression in Eq. (11):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(I_{SR}^{i} - I_{HR}^{i})^2 \tag{10}$$

$$PSNR = 10 \cdot \lg(\frac{MAX_I}{MSE}) = 20 \cdot \lg(\frac{MAX_I}{\sqrt{MSE}}) \tag{11}$$

where, n is the number of pixels in the image; I_{HR}^{i} is the real image; I_{SR}^{i} represents the reconstructed image of the model; and MAX_I is the maximum pixel value of the image, which is usually 255. The larger the value of PSNR, the better the image quality. In general, the PSNR is in the range 20–40 dB: below 20 dB, the image quality is unacceptable; a

value of 20–30 dB indicates poor image quality; a value of 30–40 dB indicates good image quality and acceptable distortion; and a value of above 40 dB indicates excellent image quality, meaning that it is very close to the original image.

(2) Structural similarity index

The structural similarity is a quality assessment framework based on structural information degradation, and was proposed by Wang et al. [84] based on the human visual system (HVS). SSIM is an indicator used to measure the similarity between two images, and is generally used to compare the similarity of the brightness, contrast, and structural details of the images. It uses the mean as the estimate of brightness, the standard deviation as the estimate of contrast, and the covariance as a measure of structural similarity. Given two images x and y, the structural similarity can be calculated using Eq. (12):

$$SSIM(x, y) = l^{\alpha} \cdot c^{\beta} \cdot s^{\gamma} \tag{12}$$

where, l represents brightness, c represents contrast, and s represents structure, and their expressions are shown in Eqs. (13), (14), and (15) respectively:

$$l = \frac{(2\mu_x\mu_y + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)} \tag{13}$$

$$c = \frac{(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{14}$$

$$s = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{15}$$

where, μ_x is the average value of x, μ_y is the average value of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y,C_1、 C_2 and C_3 is a constant used to maintain stability. Specifically, when $\alpha = \beta = \gamma = 1$, and $C_3 = \frac{1}{2}C_2$, SSIM can be expressed as Eq. (16), which is the most commonly used form for SR image quality evaluation:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{16}$$

the range of structural similarity is 0 to 1. When two images are identical, the SSIM value is equal to 1.

(3) Mean structural similarity (MSSIM) [85]

MSSIM refers to a measurement method that obtains c and s for images of different resolution by performing low-pass filtering on the same image while keeping its l constant, and then comprehensively evaluating the images. The expression for MSSIM is shown in Eq. (17):

$$MSSIM(x, y) = l^{\alpha} \cdot \prod_{i=1}^{m} (c_i)^{\beta} (s_i)^{\gamma} \tag{17}$$

(4) Feature similarity index measure (FSIM)

FSIM [86] is a measurement method that utilizes the phase consistency (PC) to extract highly correlated image feature information, and the gradient amplitude (GM) to extract contrast information that affects human visual perception, to comprehensively measure the local similarity of images. This is possible because HVS perceives images based on low-level features, whereas PC can effectively depict local structures. Due to the relative invariance of PC to image changes, it is beneficial in terms of extracting stable features in the image. However, changes in the image do sometimes affect its appearance, and it is then necessary to use the GM to compensate. In FSIM, the PC and GM complement each other. The expression for calculating FSIM is shown in Eq. (18):

$$FSIM = \frac{\sum_{x \in \Omega} S_L(x) \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \tag{18}$$

where, $S_L(x) = [S_{PC}(x)]^{\alpha} \cdot [S_G(x)]^{\beta}$ similarity for PC and GM fusion, $S_{PC}(x)$ and $S_G(x)$ represents the PC similarity and GM similarity of two images, respectively.

(5) Learned Perceptual Image Patch Similarity (LPIPS)

LPIPS [87] originates from a paper from CVPR2018. It learns the reverse mapping of generated images to real images on the ground, forces the generator to learn the reverse mapping of reconstructed real images from fake images, and prioritizes the perceptual similarity between them. LPIPS is more in line with human perception than traditional methods (SSIM, FSIM). The lower the value of LPIPS, the more similar a pair of images are. It is generally used to evaluate the performance of GAN models. The expression used to calculate this metric is shown in Eq. (19):

$$LPIPS = d(x, x_0) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot (\hat{y}_{hw} - \hat{y}_{0hw}) \right\|_2^2 \tag{19}$$

where, l represents the l-th layer of deep convolutional neural networks; $\hat{y}_{hw}$ And $\hat{y}_{0hw}$ representing the features extracted from layer l for x and x_0, and performing unit normalization on the channel dimension, w_l is a vector used to scale the active channel.

(6) Perceptual index (PI)

PI is a perception indicator proposed by Blau et al. [29] in combination with two non-reference objective evaluation indicators, NIQE and Ma. The PI value represents the subjective perception quality of an image: the lower the PI value, the better the perceptual quality of the image, which is opposite to the PSNR value. In general, a lower PI value means that the PSNR value will also be lower. The expression used to calculate the PI is shown in Eq. (20):

$$PI = \frac{1}{2}((10 - Ma) + NIQE) \tag{20}$$

4.1.2 Subjective evaluation indicators

(1) Average Opinion Score (MOS)

As set out in international standards, the MOS was initially used as a criterion to evaluate the quality of compressed speech, and was later applied to evaluate the quality of images. As shown in Table 2, the MOS value is in the range of one to five points, with one point representing the worst image quality and five representing the best. Unlike PSNR, which is an objective evaluation method, MOS is a subjective evaluation method.

4.2 Comparison of reconstruction effects

In this section, we compare the reconstruction results from the classic SISR algorithms based on deep learning. Table 3 shows the results, with the best values highlighted in bold. As mentioned earlier, the reconstruction results on the benchmark dataset are compared and analyzed from three models: CNNs, GANs, and transformers. They are also considered at three different scales: $\times 2, \times 3$, and $\times 4$. Firstly, it can be seen from the CNN-based methods, from the earliest SRCNN to the latest method, that the PSNR shows an improvement of at least 1 dB at all three scales. The RCAN with a scale of $\times 2$ even achieved a 3.84 dB improvement on the Urban100 dataset, demonstrating the excellent results that have been achieved by CNN-based methods through continuous development. However, as the depth increases, it is difficult for CNN-based models to make significant breakthroughs in indicators.

From a comparison of CNN-based and transformer-based methods, it can be seen that the PSNR value for the first IPT model to introduce a transformer into the SISR domain considerably surpasses the best PSNR value for the CNN-based models at all sthree scales. It is evident that the introduction of the transformer is very important for the SISR field, and has promoted its development. It can also be seen that SwinIR yields a relatively high level of PSNR. The ELAN model proposed in the future still has some shortcomings compared to SwinIR, but the number of parameters in ELAN is much lower than for SwinIR, making it a lightweight transformer model.

Table 2 MOS Evaluation Criteria

grade	Absolute evaluation	Relative evaluation
1	Very poor image quality	Worst in this group
2	Poor image quality	Below the average level in the group
3	The image quality is average	The average level in this group
4	Good image quality	Better than the average level in this group
5	very good image quality	The best in this group

Table 3 Performance Comparison of SISR Models Based on Deep Learning

Method	classification	scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
SRCNN [35]	CNN	×2	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
FSRCNN [36]			37.05/0.9560	32.66/0.9090	31.53/0.8920	29.88/0.9020	36.67/0.9710
VDSR [39]			37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140	37.22/0.9750
EDSR [42]			38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773
DRCN [45]			37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	—/—
DBPN [10]			38.09/0.9600	33.85/0.9190	32.27/0.9000	32.55/0.9324	38.89/0.9775
RDN [88]			38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	39.18/0.9780
RCAN [56]			38.27/0.9614	34.12/0.9216	32.41/0.9027	33.34/0.9384	39.44/0.9786
VapSR [22]			38.08/0.9612	33.77/0.9195	32.27/0.9011	32.45/0.9316	—/—
SRGAN [12]	GAN	×2	—/—	32.14/0.8860	31.89/0.8760	—/—	—/—
ESRGAN [67]			—/—	33.62/0.9150	31.99/0.8870	—/—	—/—
ESR-GAN + [70]			—/—	—/—	—/—	—/—	—/—
DGAN [89]			—/—	—/—	—/—	—/—	—/—
IPT [79]	Transformer	×2	38.37/—	34.43/—	32.48/—	33.76/—	—/—
SwinIR [81]			38.42/0.9623	34.46/0.9250	32.53/0.9041	33.81/0.9427	39.92/0.9797
ESRT [82]			—/—	—/—	—/—	—/—	—/—
ELAN [83]			38.36/0.9620	34.20/0.9228	32.45/0.9030	33.44/0.9391	39.62/0.9793
LBNet [90]			—/—	—/—	—/—	—/—	—/—
ACT [31]			38.46/0.9626	34.60/0.9256	32.560.9048	34.07/0.9443	39.95/0.9804
SRCNN [35]	CNN	×3	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
FSRCNN [36]			33.18/0.9140	29.370.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
VDSR [39]			33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290	32.01/0.9340
EDSR [42]			34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
DRCN [45]			33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276	—/—
DBPN [10]			—/—	—/—	—/—	—/—	—/—
RDN [88]			34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484
RCAN [56]			34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702	34.44/0.9499
VapSR [22]			34.52/0.9284	30.53/0.8452	29.19/0.8077	28.43/0.8583	—/—
SRGAN [12]	GAN	×3	—/—	—/—	—/—	—/—	—/—
ESRGAN [67]			—/—	—/—	—/—	—/—	—/—
ESR-GAN + [70]			—/—	—/—	—/—	—/—	—/—
DGAN [89]			—/—	—/—	—/—	—/—	—/—
IPT [79]	Transformer	×3	34.81/—	30.85/—	29.38/—	29.49/—	—/—
SwinIR [81]			34.97/0.9318	30.93/0.8534	29.46/0.8145	29.75/0.8826	35.12/0.9537
ESRT [82]			34.42/0.9268	30.43/0.8433	29.15/0.8063	28.46/0.8574	33.95/0.9455
ELAN [83]			34.90/0.9313	30.80/0.8504	29.38/0.8124	29.32/0.8745	34.73/0.9517
LBNet [90]			34.47/0.9277	30.38/0.8417	29.13/0.8061	28.42/0.8559	33.82/0.9460
ACT [31]			35.03/0.9321	31.08/0.8541	29.51/0.8164	30.08/0.8858	35.27/0.9540

Table 3 (continued)

Method	classification	scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
SRCNN [35]	CNN	×4	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCNN [36]			30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [39]			31.35/0.8830	28.02/0.7680	27.29/0.0726	25.18/0.7540	28.83/0.8870
EDSR [42]			32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
DRCN [45]			31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	—/—
DBPN [10]			32.47/0.8980	28.82/0.7860	27.72/0.7400	26.38/0.7946	30.91/0.9137
RDN [88]			32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151
RCAN [56]			32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	31.22/0.9173
VapSR [22]			32.38/0.8978	28.77/0.7852	27.68/0.7398	26.35/0.7941	—/—
SRGAN [12]	GAN	×4	29.40/0.8472	26.02/0.7397	25.16/0.6688	—/—	—/—
ESRGAN [67]			—/—	—/—	—/—	—/—	—/—
ESR-GAN + [70]			—/—	19.79/—	—/—	—/—	—/—
DGAN [89]			—/—	31.62/0.9166	31.53/0.9105	—/—	—/—
IPT [79]	Transformer	×4	32.64/—	29.01/—	27.82/—	27.26/—	—/—
SwinIR [81]			32.92/0.9044	29.09/0.7950	27.92/0.7489	27.45/0.8254	32.03/0.9260
ESRT [82]			32.19/0.8947	28.69/0.7833	27.69/0.7379	26.39/0.7962	30.75/0.9100
ELAN [83]			32.75/0.9022	28.96/0.7914	27.83/0.7459	27.13/0.8167	31.68/0.9226
LBNet [90]			32.29/0.8960	28.68/0.7832	27.62/0.7382	26.27/0.7906	30.76/0.9111
ACT [31]			32.97/0.9031	29.18/0.7954	27.95/0.7507	27.74/0.8305	32.20/0.9267

Finally, from the GAN models, we can see that the PSNR values obtained by the GAN method are not as good as those from the CNN and transformer; this is because a GAN network generally uses the perception loss to optimize the network, and the generated indicators are not as good as the pixel loss function method, but the subjective perception effect of the images generated by the GAN method is due to the other two types of methods, so we need to find more appropriate evaluation indicators to consider both performance and sense.

5 Development trends

This article has presented a review of single-image SR techniques based on deep learning, and has summarized the current research in this field. In this section, the future prospects for research in the field of image segmentation are presented, and several development trends are described below.

(1) Building a lightweight SISR model

The existing SISR network generally has two limitations: first, the network is very deep, which not only weakens the bottom-up information flow, but also causes a large model capacity and computing burden; and secondly, the network architecture is often feedforward, making it difficult for the first few layers to capture useful information from the later ones, which limits the ability of the network to learn features. It is therefore necessary to

devise lightweight SISR models. The main aims of designing a lightweight neural network are to obtain a more efficient network, to optimize the network structure and convolution calculations, to reduce the number of network parameters without loss of performance, to strengthen the understanding of the internal network, and to alleviate the problems with implementing SR on mobile devices.

(2) Unsupervised SR reconstruction

Supervised image SR reconstruction requires LR-HR image pairs as datasets, and a degradation module needs to be designed for the model to degrade the HR images. This degradation method is fixed and single, which is inconsistent with the complex and variable types of degradation seen in reality, which can seriously affect the development and application value of image SR. An unsupervised image segmentation process does not require paired training samples, which reduces the requirement for training samples and is more in line with the actual needs of segmentation. However, this also places higher demands on the learning ability of the model. The issue of how to achieve unsupervised SR reconstruction (that is, SR reconstruction without the need to construct LR-HR image pairs) is therefore a promising direction for future development.

(3) Designing a more scientific and reasonable loss function and evaluation index

Following the introduction of the perception loss function, more HF texture details can now be recovered in the SISR reconstruction task, and the effects as seen by the human eye are better, but the values of the RSNR evaluation index are lower. There is therefore a contradiction between good visual perception and high performance indicators. Although the MOS has been proposed as an evaluation indicator, this evaluation process involves a significant amount of time and labor costs. The future SISR reconstruction task requires that the human visual perception system is considered, in order to propose a loss function that is more consistent with human visual perception and an evaluation method that takes both human senses and the model performance into account to meet the actual needs of the user.

(4) Improving upsampling methods

Current upsampling methods suffer from problems such as a lack of end-to-end learning, uneven distribution of the receptive field, and the chessboard effect, which will lead to inefficiency of the SISR algorithm and unstable reconstruction results. At present, most upsampling methods are based on integer multiples, and developing an efficient and suitable upsampling method for any amplification factor is therefore a direction that is worthy of further research in the future.

Acknowledgements This work was supported by the following grants: National Natural Science Foundation of China: 62276088, 62102129, Natural Science Foundation of Hebei Province: F2021202030.

Data availability The datasets generated during and/or analyzed during the current study are available in the following repositories: Set5(https://www.kaggle.com/msahebi/super-resolution), Set14(https://www.kaggle.com/msahebi/super-resolution), BSD100(https://www.kaggle.com/msahebi/super-resolution), Urban100(https://drive.google.com/drive/folders/1pRmhEmmY-tPF7uH8DuVthfHoApZWJ1QU?usp=sharing), Manga109(http://www.manga109.org/en/index.html), DIV2K(https://data.vision.ee.ethz.ch/cvl/DIV2K/), Flickr2K(https://drive.google.com/drive/folders/1B-uaxvV9qeuQ-t7MFiN1oEdA6dKnj2vW), RealSR(https://drive.google.com/open?id=17ZMjo-zwFouxnm_aFM6CUHBwgRrLZqIM), DPED(https://

drive.google.com/file/d/0BwOLOmqkYj-jeUJwQjRNUFkzOTA/view), OutdoorScene(https://drive.google.com/drive/u/1/folders/ 1iZfzAxAwOpeutz27HC56_y5RNqnsPPKr), PIRM(https://drive.google.com/drive/folders/ 17FmdXu5t8wlKwt8extb_nQAdjxUOrb1O?usp=sharing), T91(https://drive.google.com/drive/folders/1pRmhEmmY-tPF7uH8DuVthfHoApZWJ1QU?usp=sharing),ImageNet(https://image-net.org/challenges/LSVRC/), City100(https://github.com/ngchc/CameraSR), MSCOCO(https://cocodataset.org/#download), PIPAL(https://github.com/HaomingCai/PIPAL-dataset).

Declarations

Conflicts of interests All authors declared that we have no conflicts of interests to this work.

References

1. Huang TJCV, Processing I (1984) Multi-frame image restoration and registration. Multiframe Image Restor Registration 1:317–339
2. Greenspan HJTCJ (2009) Super-resolution in medical imaging. Comput J 52:43–63
3. Isaac JS, Kulkarni R (2015) Super resolution techniques for medical image processing. 2015 International Conference on Technologies for Sustainable Development (ICTSD). IEEE, pp 1–6
4. Huang Y, Shao L, Frangi AF (2017) Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 5787–5796
5. Thornton MW, Atkinson PM, Holland DJIJORS (2006) Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping. Int J Remote Sens 27:473–491
6. Barzegar S, Sharifi A, Manthouri MJMT et al (2020) Super-resolution using lightweight detailnet network. Multimed Tools Appl 79:1119–1136
7. Yang W, Zhou F, Zhu R et al (2019) Deep learning for image super-resolution. Neurocomputing 398:291–292
8. Timofte R, Rothe R, Van Gool L (2016) Seven ways to improve example-based single image super resolution. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1865–1873
9. Zeyde R, Elad M, Protter M (2012) On single image scale-up using sparse-representations. Curves and Surfaces: 7th International Conference. Springer, pp 711–730
10. Haris M, Shakhnarovich G, Ukita N (2018) Deep back-projection networks for super-resolution. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1664–1673
11. Lai W-S, Huang J-B, Ahuja N et al (2017) Deep laplacian pyramid networks for fast and accurate super-resolution. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 5835–5843
12. Ledig C, Theis L, Huszár F et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 105–114
13. Bevilacqua M, Roumy A, Guillemot C et al (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 23rd British Machine Vision Conference, pp 1–10.
14. Huang J-B, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 5197–5206
15. Martin D, Fowlkes C, Tal D et al (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. IEEE, pp 416–423
16. Fujimoto A, Ogawa T, Yamamoto K et al (2016) Manga109 dataset and creation of metadata. 1st international workshop on comics analysis, processing and understanding (MANPU), pp 1–5
17. Agustsson E, Timofte R (2017) Ntire 2017 challenge on single image super-resolution: Dataset and study. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, pp 1122–1131
18. Xia B, Hang Y, Tian Y et al (2022) Efficient non-local contrastive attention for image super-resolution. 36th AAAI Conference on Artificial Intelligence 36(3): 2759–2767
19. Lee J, Jin KH (2022) Local texture estimator for implicit representation function. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1929–1938

20. Ma C, Zhang J, Zhou J et al (2022) Learning Series-Parallel Lookup Tables for Efficient Image Super-Resolution. 17th European Conference on Computer Vision (ECCV). Springer, 13677: 305–321
21. Timofte R, Agustsson E, Van Gool L et al (2017) Ntire 2017 challenge on single image super-resolution: Methods and results. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, pp 1110–1121
22. Zhou L, Cai H, Gu J et al (2022) Efficient image super-resolution using vast-receptive-field attention. European Conference on Computer Vision (ECCV). Springer, pp 256–272
23. Ji X, Cao Y, Tai Y et al (2020) Real-world super-resolution via kernel estimation and noise injection. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops(CVPRW). IEEE, pp 1914–1923
24. Liang J, Zeng H, Zhang L (2022) Efficient and degradation-adaptive network for real-world image super-resolution. 17th European Conference on Computer Vision (ECCV). Springer, 13867: 574–591
25. Cai J, Zeng H, Yong H et al (2019) Toward real-world single image super-resolution: A new benchmark and a new model. IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, pp 3086–3095
26. Deng J, Dong W, Socher R et al (2009) Imagenet: A large-scale hierarchical image database. IEEE-Computer-Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp 248–255
27. Ignatov A, Kobyshev N, Timofte R et al (2017) Dslr-quality photos on mobile devices with deep convolutional networks. 16th IEEE International Conference on Computer Vision (ICCV). IEEE, pp 3297–3305
28. Wang X, Yu K, Dong C et al (2018) Recovering realistic texture in image super-resolution by deep spatial feature transform. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 606–615
29. Blau Y, Mechrez R, Timofte R et al (2018) The 2018 PIRM challenge on perceptual image super-resolution. 15th European Conference on Computer Vision (ECCV), vol 11133. Springer, pp 334–355
30. Yang J, Wright J, Huang TS et al (2010) Image super-resolution via sparse representation. IEEE Trans Image Process 19:2861–2873
31. Yoo J, Kim T, Lee S et al (2022) Enrich CNN-transformer feature aggregation networks for super-resolution. 23rd IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 4945–4954
32. Lin T-Y, Maire M, Belongie S et al (2014) Microsoft coco: common objects in context. 13th European Conference on Computer Vision (ECCV), vol 8693. Springer, pp 740–755
33. Jinjin G, Haoming C, Haoyu C et al (2020) Pipal: a large-scale image quality assessment dataset for perceptual image restoration. European Conference on Computer Vision (ECCV). Springer, pp 633–651
34. Chen C, Xiong Z, Tian X et al (2019) Camera lens super-resolution. 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1652–1660
35. Dong C, Loy CC, He K et al (2015) Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell 38:295–307
36. Dong C, Loy CC, Tang X (2016) Accelerating the super-resolution convolutional neural network. 14th European Conference on Computer Vision (ECCV), vol 9906. Springer, pp 391–407
37. Shi W, Caballero J, Huszár F et al (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1874–1883
38. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 770–778
39. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1646–1654
40. Simonyan K, Zisserman AJaPA (2015) Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR), pp 1–14
41. Mao X-J, Shen C, Yang Y-BJaPA (2016) Image restoration using convolutional auto-encoders with symmetric skip connections. Neural Information Processing Systems (NIPS) 29
42. Lim B, Son S, Kim H et al (2017) Enhanced deep residual networks for single image super-resolution. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, pp 1132–1140
43. Nah S, Hyun Kim T, Mu Lee K (2017) Deep multi-scale convolutional neural network for dynamic scene deblurring. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 257–265

44. Szegedy C, Ioffe S, Vanhoucke V et al (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. 31st AAAI Conference on Artificial Intelligence, pp 4278–4284
45. Kim J, Lee JK, Lee KM (2016) Deeply-recursive convolutional network for image super-resolution. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1637–1645
46. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2790–2798
47. Han W, Chang S, Liu D et al (2018) Image super-resolution via dual-state recurrent networks. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1654–1663
48. Gilbert CD, Sigman MJN (2007) Brain states: top-down influences in sensory processing. Neuron 54:677–696
49. Hupé J, James A, Payne B et al (1998) Cortical feedback improves discrimination between figure and background by V1, V2 and V3 neurons. Nature 394:784–787
50. Li Z, Yang J, Liu Z et al (2019) Feedback network for image super-resolution. 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 3867–3871
51. Huang G, Liu Z, Van Der Maaten L et al (2017) Densely connected convolutional networks. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2261–2269
52. Tong T, Li G, Liu X et al (2017) Image super-resolution using dense skip connections. 16th IEEE International Conference on Computer Vision (ICCV). IEEE, pp 4809–4817
53. Tai Y, Yang J, Liu X et al (2017) Memnet: A persistent memory network for image restoration. 16th IEEE International Conference on Computer Vision (ICCV). IEEE, pp 4549–4557
54. Chaudhari S, Mithal V, Polatkan G et al (2021) An attentive survey of attention models. ACM Trans Intell Syst Technol (TIST) 12(5):1–32
55. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 7132–7141
56. Zhang Y, Li K, Li K et al (2018) Image super-resolution using very deep residual channel attention networks. 15th European Conference on Computer Vision (ECCV), vol 11211. Springer, pp 294–310
57. Dai T, Cai J, Zhang Y et al (2019) Second-order attention network for single image super-resolution. 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 11057–11066
58. Zhang Y, Wei D, Qin C et al (2021) Context reasoning attention network for image super-resolution. 18th IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, pp 4258–4267
59. Zhao H, Kong X, He J et al (2020) Efficient image super-resolution using pixel attention. European Conference on Computer Vision (ECCV) Workshops. Springer, pp 56–72
60. Hui Z, Wang X, Gao X (2018) Fast and accurate single image super-resolution via information distillation network. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 723–731
61. Hui Z, Gao X, Yang Y et al (2019) Lightweight image super-resolution with information multi-distillation network. 27th ACM International Conference on Multimedia (MM), pp 2024–2032
62. Luo X, Xie Y, Zhang Y et al (2020) Latticenet: towards lightweight image super-resolution with lattice block. European Conference on Computer Vision (ECCV). Springer, pp 272–289
63. Zhang Y, Wang H, Qin C et al (2021) Learning efficient image super-resolution networks via structure-regularized pruning. International conference on learning representations 1–12
64. Zhang Y, Wang H, Qin C et al (2021) Aligned structured sparsity learning for efficient image super-resolution. Adv Neural Inf Process Syst 34:2695–2706
65. Wang H, Zhang Y, Qin C et al (2023) Global aligned structured sparsity learning for efficient Image super-resolution. IEEE Transactions on pattern analysis and machine intelligence. IEEE 45:10974–10989
66. Goodfellow I, Pouget-Abadie J, Mirza M et al (2020) Generative adversarial networks. IEEE Signal Process Mag 63:139–144
67. Wang X, Yu K, Wu S et al (2018) Esrgan: Enhanced super-resolution generative adversarial networks. 15th European Conference on Computer Vision (ECCV) Workshops, vol 11133. Springer, pp 63–79
68. Jolicoeur-Martineau AJaPA (2018) The relativistic discriminator: a key element missing from standard GAN. International Conference on Learning Representations (ICLR 2019)
69. Lee O-Y, Shin Y-H, Kim J-OJIA (2019) Multi-perspective discriminators-based generative adversarial network for image super resolution. IEEE Access 7:136496–136510
70. Rakotonirina NC, Rasoanaivo A (2020) ESRGAN+: Further improving enhanced super-resolution generative adversarial network. ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp 3637–3641
71. Chen Y, Li J, Xiao H et al (2017) Dual path networks. 31st Annual Conference on Neural Information Processing Systems (NIPS) 30

72. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, pp 4401–4410
73. Shi W, Tao F, Wen YJITOI et al (2023) Structure-aware deep networks and pixel-level generative adversarial training for single image super-resolution. IEEE Trans Instrum Meas 72:1–14
74. Isola P, Zhu J-Y, Zhou T et al (2017) Image-to-image translation with conditional adversarial networks. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 5967–5976
75. Ma C, Rao Y, Lu J et al (2021) Structure-preserving image super-resolution. IEEE Trans Pattern Anal Mach Intell 44:7898–7911
76. Ma C, Rao Y, Cheng Y et al (2020) Structure-preserving super resolution with gradient guidance. IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, pp 7769–7778
77. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. 31st Annual Conference on Neural Information Processing Systems (NIPS) 30
78. Arnab A, Dehghani M, Heigold G et al (2021) Vivit: A video vision transformer. 18th IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, pp 6816–6826
79. Chen H, Wang Y, Guo T et al (2021) Pre-trained image processing transformer. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 12294–12305
80. Liu Z, Lin Y, Cao Y et al (2021) Swin transformer: Hierarchical vision transformer using shifted windows. 18th IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, pp 9992–10002
81. Liang J, Cao J, Sun G et al (2021) Swinir: Image restoration using swin transformer. 18th IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, pp 1833–1844
82. Lu Z, Li J, Liu H et al (2022) Transformer for single image super-resolution. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops(CVPRW). IEEE, pp 456–465
83. Zhang X, Zeng H, Guo S et al (2022) Efficient long-range attention network for image super-resolution. 17th European Conference on Computer Vision (ECCV), vol 13677. Springer, pp 649–667
84. Wang Z, Bovik AC, Sheikh HR et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Proc 13:600–612
85. Ma C, Yang C-Y, Yang X et al (2017) Learning a no-reference quality metric for single-image super-resolution. Comput Vis Image Underst 158:1–16
86. Zhang L, Zhang L, Mou X et al (2011) FSIM: A feature similarity index for image quality assessment. IEEE Trans Image Proc 20:2378–2386
87. Zhang R, Isola P, Efros AA et al (2018) The unreasonable effectiveness of deep features as a perceptual metric. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 586–595
88. Zhang Y, Tian Y, Kong Y et al (2018) Residual dense network for image super-resolution. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2472–2481
89. Zareapoor M, Celebi ME, Yang JJSPIC (2019) Diverse adversarial network for image super-resolution. Signal Proc: Image Commun 74:191–200
90. Gao G, Wang Z, Li J et al (2022) Lightweight bimodal network for single-image super-resolution via symmetric cnn and recursive transformer. Thirty-First International Joint Conference on Artificial Intelligence (IJCAI), pp 913–919