# Super-resolution reconstruction based on two-stage residual neural network

Dong, Lin
Department of Communication Design Science, Kyushu University

Inoue, Kohei
Department of Communication Design Science, Kyushu University



出版情報：Machine Learning with Applications. 6 (100162), 2021-12-15. Elsevier
バージョン：

Machine Learning with Applications 6 (2021) 100162

Contents lists available at ScienceDirect

Machine Learning with Applications

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

# Super-resolution reconstruction based on two-stage residual neural network

Lin Dong, Kohei Inoue *

*Department of Communication Design Science, Kyushu University, 4-9-1, Shiobaru, Minami-ku, Fukuoka, 815-8540, Japan*

ARTICLE INFO

*Keywords:*
Super-resolution reconstruction
Deep learning
Two-stage residual network

ABSTRACT

With the constant update of deep learning technology, the super-resolution reconstruction technology based on deep learning has also attained a significant breakthrough. This paper primarily discusses the integration of deep learning and super-resolution reconstruction techniques. Regarding the application of deep learning in super-resolution reconstruction, the improvement is focused on the two dimensions of algorithm efficiency and reconstruction effect. On the basis of the currently available neural network algorithms, this paper puts forward the two-stage residual super-resolution reconstruction network structure. Thereinto, the improvement is mainly embodied in the modification of the image feature extraction network modules and the increase of the residual block into two stages. It is experimentally evidenced by algorithm simulation that the two-stage residual network in this paper shows a certain extent of improvement for the super-resolution reconstruction effect compared with the related methods.

## 1. Introduction

Following the evolution of computer technology, images and videos have not only emerged as the primary information source for the public, but also served critical applications in various fields. For transportation sector, images and videos have acted as the principal legal basis for vehicle violations and traffic accidents; within the medical field, images such as ultrasound and CT have been available as diagnostic aids for physicians to deliver accurate diagnoses; regarding surveillance domain, images and videos have offered round-the-clock environmental information, which guarantees the security of a harmonious society. Beyond that, there are plenty of applications for both images and videos. Yet, in either application, clear quality of images and videos is absolutely a must.

In reality, however, interference from internal hardware and external electromagnetic waves during signal transmission can render the final acquired images to be of low resolution, with some inferior quality issues such as noise, missing details, blurring, and distortion. The reduction of digital image resolution and quality deviates from the objective requirements of a specified scenario and the sensory demands of users. To effectively address such problems, it is usually to consider both hardware and software approaches. However, hardware-based solutions for low image quality often incur significant costs. As a result, there is a growing tendency to exploit software algorithms for image resolution enhancement.

Elad and Feuer originally presented a generalization of restoration theory for the problem of super-resolution reconstruction (SRR) of an image (Elad & Feuer, 1996). Since then, a large number of SRR methods have been proposed by researchers. Recently, Yang et al. have reviewed representative deep learning-based single image super-resolution methods (Yang et al., 2019). Deep learning-based image SRR primarily exploits an extensive pool of high-resolution images $I_{HR}$ as training samples. The characteristics of $I_{HR}$ are studied by convolutional neural network, while the characteristics learned during the training process are utilized to estimate the corresponding super-resolution reconstructed image $I_{SR}$ during the input of low-resolution image $I_{LR}$. A key step for this process lies in a better and faster approach to acquiring the characteristics of $I_{HR}$.

The process of SRR can be viewed as a model for predicting a high-resolution image from it low-resolution counterpart. Such prediction models have a wide variety of applications including forecasting the future price of crude oil (Karasu et al., 2020), detection of solder paste defects (Sezer & Altan, 2021), wind speed forecasting (Altan et al., 2021), prediction of Bitcoin prices (Karasu et al., 2018), exchange rate forecasting (Altan & Karasu, 2019). This paper proposes a novel two-stage neural network model for SRR.

At present, the dominant methods for deep learning-based image super-resolution are all approaches developed on the basis of mean squared error (MES) as the training objective function, including the methods of SRCNN (Dong et al., 2014; Dong et al., 2016), SR-GAN (Ledig et al., 2017), SRResNet (Ledig et al., 2017) and ESPCN (Shi et al., 2016). The above approaches have been consistently optimized regarding super-resolution reconstruction, with significant enhancements achieved in reconstruction effectiveness and efficiency. The SR-CNN model, a classic of deep learning introducing super-resolution

* Corresponding author.
*E-mail addresses:* dolly8060@hotmail.com (L. Dong), k-inoue@design.kyushu-u.ac.jp (K. Inoue).

https://doi.org/10.1016/j.mlwa.2021.100162
Received 16 July 2021; Received in revised form 27 August 2021; Accepted 11 September 2021
Available online 21 September 2021

reconstruction, uses bicubic interpolation as a pre-processing process. The specific method of this model is that for a low-resolution image, it is first scaled up to the target size using interpolation, and then a nonlinear mapping is done by a three-layer convolutional network, and the obtained result is output as a high-resolution image. This model has the advantage of a simple network structure, but the model only uses one convolutional layer to extract features so there is a problem of a relatively small perceptual field, and the extracted features are very local features, which cannot recover image details. FSRCNN improves on SRCNN by using a transposed convolution layer to enlarge the size in the last layer of the SRCNN network, so that the original low-resolution image can be directly input into the network instead of needing to enlarge the size by bicubic interpolation first as in the previous SRCNN. Since FSRCNN does not require the operation of scaling up the image size outside the network, FSRCNN has a large speedup compared with SRCNN. However, FSRCNN still suffers from the same drawback compared to SRCNN, i.e., it cannot recover image details. Because SRCNN needs to interpolate the low-resolution image by upsampling before inputting it into the network, which means to perform convolution operation on higher resolution, thus increasing the computational complexity. In contrast, the ESPCN model is an efficient method that can directly extract features on low-resolution image sizes and compute them to obtain high-resolution images. The paper proposes a method to do super-resolution by CNN directly on low-resolution images, and then complete the super-resolution reconstruction of images by upsampling with a sub-pixel convolutional layer directly at the end of the network. Compared with SRCNN, this method greatly reduces the computational complexity of the model. However, also because of the simplicity of the model, much high-frequency information is lost in the reconstructed SR images by ESPCN. The SRGAN and SRResNet SRR algorithms presented by Ledig et al. (Ledig et al., 2017) better solve this problem by using multiple residual modules and designing new loss functions (adding Perceptual Loss and VGG Loss Function, etc.) to address the problem of high-frequency information loss in the super-resolution reconstruction problem. Compared with the previous model, this model has improved a lot in terms of evaluation metrics. However, due to the complexity of this model, the computational complexity is also increased accordingly.

This paper is devoted to the study of image SRR based on deep learning algorithms, which is expected to enhance the image resolution by utilizing the low-cost software algorithm, and is of great relevance to the demand for high-quality images in the fields of transportation, medicine, and surveillance.

The contributions of this study are summarized as follows:

(1) Within the image feature extraction network block, the residual blocks in SRResNet are substituted with residual dense blocks (RDBs) in Residual Dense Network (RDN). While reducing the computational complexity, the functionality of high-frequency feature extraction can be retained. Simultaneously, Batch Normalization (BN) is removed from the SRResNet network structure for accelerating the algorithm's computational speed.

(2) The residual module in this paper is expanded to two stages. In the first stage of reconstruction, utilizing the local residual learning of the RDBs, the features of the low-resolution image, namely the pseudo-high-frequency information, are extracted. In the second stage of reconstruction, the original low-resolution image is fused with the pseudo-high-frequency features output from the first stage by dilated convolution. The inclusion of the dilated convolutional structure can enhance the conversion quality of pseudo-high frequency information to high-frequency information in a two-order network while retaining a large receptive field. Secondly, the residual information between adjacent RDBs can be made as sparse as possible, which enhances the residual relearning of the network model. The objective of global residual learning is achieved, thereby obtaining realistic high-resolution images with enriched details.

**Table 1**
Report card of the ResNet in 2015 ILSVRC (He et al., 2016).

| Method | Top-5 err. (test) |
|---|---|
| VGG (Simonyan & Zisserman, 2014) (ILSVRC'14) | 7.32 |
| GoogLeNet (Szegedy et al., 2015) (ILSVRC'14) | 6.66 |
| VGG (Simonyan & Zisserman, 2014) (v5) | 6.8 |
| PreLU-net (He et al., 2015) | 4.94 |
| BN-inception (Ioffe & Szegedy, 2015) | 4.82 |
| ResNet (He et al., 2016) (ILSVRC'15) | **3.57** |

As revealed by the algorithm simulation and comparison experiments, comparatively, the algorithm in this paper demonstrates some improvement in the reconstruction effect.

The rest of this paper is organized as follows. Section 2 summarizes two related networks utilized in our network described in the following section. Section 3 proposes a two-stage residual neural network model for image super-resolution. Section 4 shows experimental results of image super-resolution using publicly available image datasets. Section 5 discusses the main points achieved in this work. Finally, Section 6 concludes this paper.

## 2. Related networks

In this section, we briefly summarize two related networks as a preparation for the following discussions.

### 2.1. Residual neural network (ResNet)

Upon the introduction of the SRCNN algorithm (Dong et al., 2014; Dong et al., 2016), concerns have been raised by researchers as to whether "the deeper the layers are, the higher the accuracy will be" in deep learning networks. To wit, is it possible to achieve superior SRR effect than SRCNN algorithm by deepening the number of layers of network structure? Unfortunately, it has been observed through extensive experiments that in case of only deepening the number of layers in the neural network, problems such as gradient dispersion and error increase are often present during the training of the neural network model. Fig. 1 illustrates the simulation experimental results of plain network utilizing CIFAR-10 image dataset (Krizhevsky, 2009) for 20 layers and 56 layers respectively. As can be noticed in the figure, the training error and test error have increased instead as the network layer increases from 20 to 56. Moreover, further experiments have demonstrated that the error can be greater as the network layers expand to 100 or more.

In the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015), the residual module up to 152 layers in depth, which was designed by He et al. (He et al., 2016), won the championship. In the residual neural network, the challenges such as the gradient dispersion and accuracy degradation arising from the increased number of layers are tackled. The report card of ResNet achieved in 2015 ILSVRC is provided in Table 1.

Residual Neural Network (ResNet) (He et al., 2016) refers to a residual structural block that incorporates a "jumping structure" in a typical convolutional neural network structure. As presented in Fig. 2, it is a schematic comparison diagram of a typical neural network structure and one with residual blocks. Owing to the incorporation of the residual network structure, it is also capable to ensure an excellent training accuracy and performance despite the increased number of layers in the deep neural network (over 1000 layers).

In Fig. 2, the residual structure block can be expressed as

$$y_l = H(x_l) + F(x_l, w_l)$$
$$x_{l+1} = f(y_l), \tag{1}$$

where $x_l$ denotes the input of the residual block, $x_{l+1}$ indicates the corresponding output, $l$ indicates the number of layers of the network,

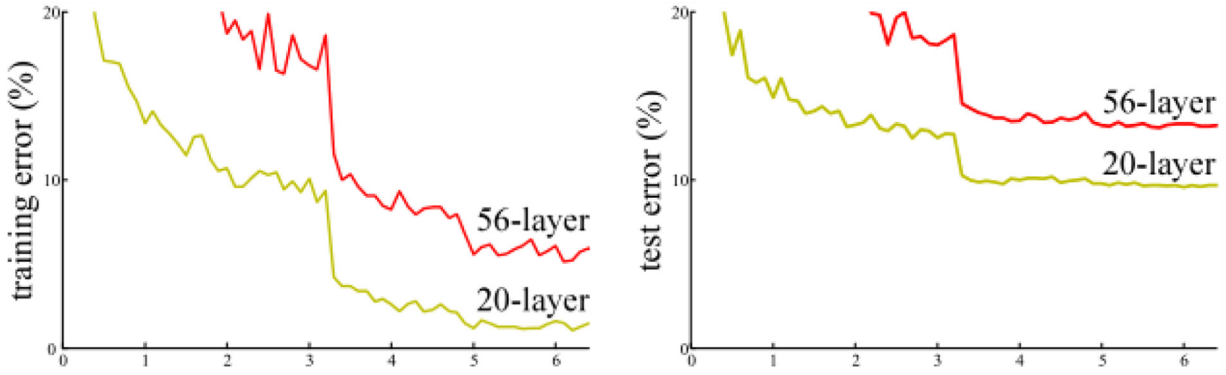
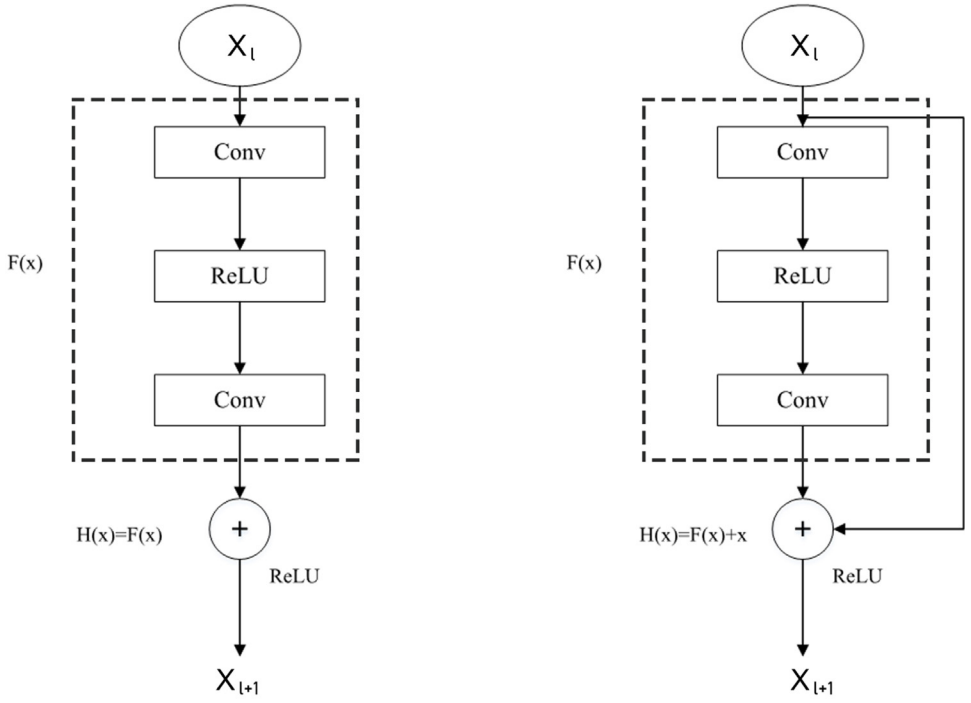



**Fig. 1.** Training error (left) and test error (right) curves on CIFAR-10 with 20-layer and 50-layer plain networks (He et al., 2016).




(a) Regular structure block (b) Residual structure block

**Fig. 2.** Schematic Diagram of the Regular structure block (a) and the one with residuals (b).

$w_l$ indicates the $1 \times 1$ convolution kernel, $F$ represents the residual learning function, as

$$F(x) = H(x) - x \tag{2}$$

for the residual expression, $H(x_l)$ stands for the desired output, which is an identical mapping. When the network structure is increased to a great depth, it is possible to make $F(x) + x \to x$, in other words, only the residuals between the input $x_l$ and the output $x_{l+1}$ are learned, and $f$ represents the ReLU activation function after $H(x)$.

From the residual block structure, the training feature expressions are obtained as

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, w_i). \tag{3}$$

From the perspective of forward propagation, the training results of the features are presented in a cumulative form owing to the design of the "jumping structure". Compared with the serially connected structure, the computational process increases the stability. By analyzing from the perspective of back propagation, the partial derivative of Loss Function $\epsilon$ by using (2) yields

$$\frac{\partial \epsilon}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L}\left[1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, w_i)\right] \tag{4}$$

In the regular neural network structure, with the increasing number of network layers, while training the multilayer feedforward network, since the final result will always be multiplied around 0, it generates gradient dispersion due to the training result being close to 0. However, upon the addition of the residual block design in the neural network structure, when $\frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, w_i) = 0$, the final gradient also converges

to 1. Consequently, the design of the residual network structure can effectively eliminate the gradient dispersion issue.

As an effort to understand the difference between residual block networks and traditional neural network structures at a deeper level, a schematic diagram of a residual block in an unfolded state is demonstrated in Fig. 3. The network input within the residual structure presented in the figure is not only relevant to the output of the previous layer, but also to the input of other layers. In contrast, in traditional networks, the input of the current layer is only associated with the output of the previous layer, with some information loss or depletion. This residual structure can avoid this problem. $F$ in Fig. 3 represents the feature.

### 2.2. Residual Dense Network (RDN)

As one of the approaches for super-resolution reconstruction, the Residual Dense Network (RDN) was proposed by Zhang et al. (Zhang et al., 2018). RDN incorporates dense block and residual learning, whose network structure mainly contains four parts: shallow feature extraction network (SFENet), residual dense blocks (RDBs), dense feature fusion (DFF), and Up-Sampling Net.

The network structure of RDN is shown in Fig. 4. SFENet module comprises two convolutional layers (Conv) with a convolution kernel size of $3 \times 3$, primarily devoted to the shallow feature map region of the input low-resolution image; RDBs incorporate multiple RDBs (RDB), whose main function is to extract the local features of the input image, and the feature extraction of multiple RDBs to acquire the integral image information and features; the Dense Feature Fusion (DFF) is composed of Global Feature Fusion (GFF) and Global Residual Learning (GRL), whose function focuses on fusing the features extracted by the RDB to realize the extraction of global features; the function of Up-Sampling Net module consists of Up-Sampling and Conv, which fulfills the function of amplifying and reconstructing the input low-resolution image. The above process can be expressed by the following equations.

The realization principle of shallow feature extraction network is demonstrated by

$$\begin{aligned} F_{-1} &= H_{SFE1}(I_{LR}) \\ F_0 &= H_{SFE2}(F_{-1}), \end{aligned} \tag{5}$$

where $I_{LR}$ denotes the low-resolution input image, $H_{SFE1}$ and $H_{SFE2}$ indicate the two convolutional layers, and $F_{-1}$ and $F_0$ stand for the result of convolution. The principle of RDB implementation is represented by

$$\begin{aligned} F_d &= H_{RDB,d}(F_{d-1}) \\ &= H_{RDB,d}(H_{RDB,d-1}(F_{d-2})) \\ &= H_{RDB,d}(H_{RDB,d-1}(\cdots(H_{RDB,1}(F_0))\cdots)), \end{aligned} \tag{6}$$

$$F_{d,c} = \sigma(W_{d,c}[F_{d-1}, F_{d,1}F_{d,2}, \ldots, F_{d,c-1}]), \tag{7}$$

where $H_{RDB,d}$ denotes the $d$th RDB, which achieves the results of the $d-1$th local feature fusion and learning. $\sigma$ indicates the activation function between each RDB, where ReLU is generally employed, $W_{d,c}$ is the weight of the convolution operation, $[F_{d-1}, F_{d,1}, F_{d,2}, \ldots, F_{d,c-1}]$ refers to the process of LFF and LRL, and the final output of the d-layered RDB is $F_d$. The realization of the DFF algorithm is illustrated by

$$F_{DF} = H_{DFE}(F_{-1}, F_0, F_1, \ldots, F_D), \tag{8}$$

where $F_{-1}$ is the output feature of the shallow convolution, and $F_0, F_1, \ldots, F_D$ represents the output result of $D+1$ RDBs. The process of RDN to achieve a low-resolution image $I_{LR}$ to a high-resolution image $I_{HR}$ is given by

$$I_{HR} = H_{RDN}(I_{LR}). \tag{9}$$

## 3. Proposed method

In this section, we first describe the structure of a two-stage residual neural network model for image super-resolution based on the above related networks. Then we explain the detailed procedures in each stage. After that, we give an objective function for optimizing the proposed model.

### 3.1. Two-stage residual model

The schematic diagram of the two-stage residual network model proposed in this paper is shown in Fig. 5. During the network input, firstly, the low-resolution images are convolutionally preprocessed with preliminary feature extraction. In the first stage of reconstruction, utilizing the local residual learning of the RDBs, the features of the low-resolution image, namely the pseudo-high-frequency information, are extracted. In the second stage of reconstruction, the original low-resolution image is fused with the pseudo-high-frequency features output from the first stage by dilated convolution. The objective of global residual learning is achieved, thereby obtaining realistic high-resolution images with enriched details. The detailed algorithm will be described in the following sections.

### 3.2. First stage

In the first stage of the two-stage residual learning network structure, it primarily exploits the feature learning performance of the RDBs by inputting preliminary detailed features of low-resolution images for learning.

#### 3.2.1. Residual dense blocks (RDBs)

Residual dense blocks (RDBs) consist of multiple residual dense blocks (RDB). RDB incorporates the advantages of residual learning and dense blocks, where the combination of multiple RDB structures is capable of extracting the complete image information and features (Zhang et al., 2018). The structure diagram of RDB is presented in Fig. 6, where, (a) is the structure schematic diagram of residual block, (b) is the structure schematic diagram of dense block, and (c) is the structure schematic diagram of RDB, from which we can see that RDB is a combination of Residual block and Dense block. Each RDB composes of three modules: Contiguous Memory Structure, Local Feature Fusion, and Local Residual Learning. Contiguous Memory (CM), which is the connection line between multiple Convs and ReLUs in Fig. 6(c), serves mainly to transfer the RDB residual information of the previous layer to the subsequent layer; the role of Local Feature Fusion (LFF) involves the information fusion of the previous RDB layer with the Conv layer information of the current RDB layer. Since in the RDB structure, the output feature map of the previous RDB layer is directly attached to the current layer, there is a large number of features generated under this structure. In LFF, upon information fusion, a Conv of dimension $1 \times 1$ is utilized to reduce the number of feature maps and maintain aspect of the image constant; Local Residual Learning (LRL) functions by accumulating the output of the previous layer of RDB with the output of LFF, with a view to increasing the information representation capability of the network and attaining a more stable output.

#### 3.2.2. Implementation process

In Fig. 5, the input low-resolution image, which passes through a convolutional layer, Conv, and a ReLU densely connected layer, is extracted for features and learned. There are two major functions of the features extracted from this layer, which are firstly to provide input for the next layer, and secondly to offer input for the residual learning module. On each occasion, the output of the current layer is adopted as the input of the next layer. The features obtained from the two layers are mapped nonlinearly after feature extraction. The features obtained

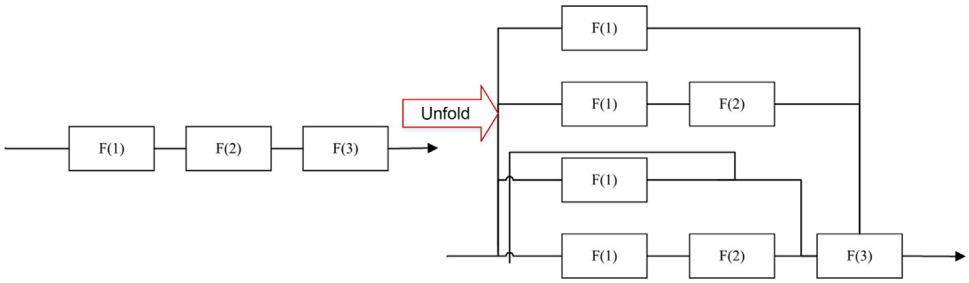



**Fig. 3.** Schematic diagram of expanded residual network.

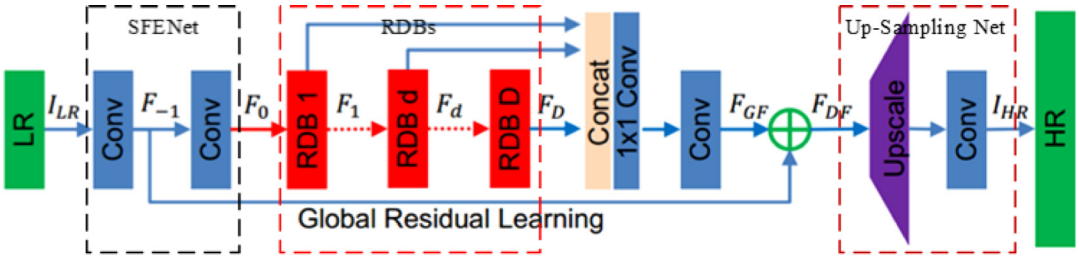



**Fig. 4.** Schematic diagram of RDN network structure (Zhang et al., 2018).

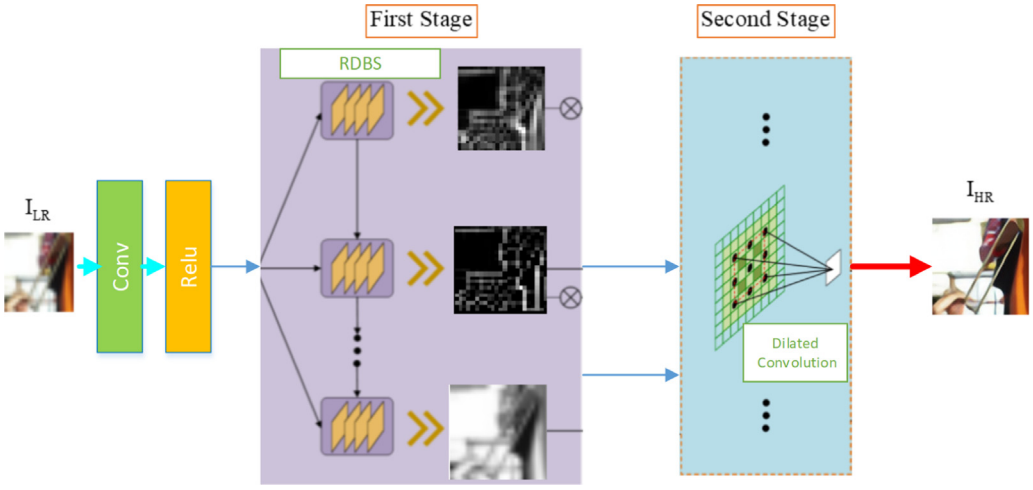



**Fig. 5.** Schematic diagram of the two-stage residual network model in this paper. A low-resolution "violin" image is inputted from left, and its high-resolution version is outputted to the right.

from the mapping are served as inputs to the residual learning module of RDB.

In the network structure proposed in this paper, in accordance with the design idea of MemNet (Tai et al., 2017), the RDBs consists of 16 RDBs (RDB) in series with the Memory Block cyclic cell as the main component. Following the first RDB obtaining the output of the convolutional layer, the features obtained via the processing of the current RDB layer are utilized as the input of the second RDB, and so on, up to the 16th residual layer. The implementation process is presented in

$$Y_d^r = R(Y_d^{r-1}) = \Gamma(Y_d^{r-1}) + Y_d^{r-1} \tag{10}$$

$$Y_d^r = [Y_d^1, Y_d^2, \ldots, Y_d^{16}], \tag{11}$$

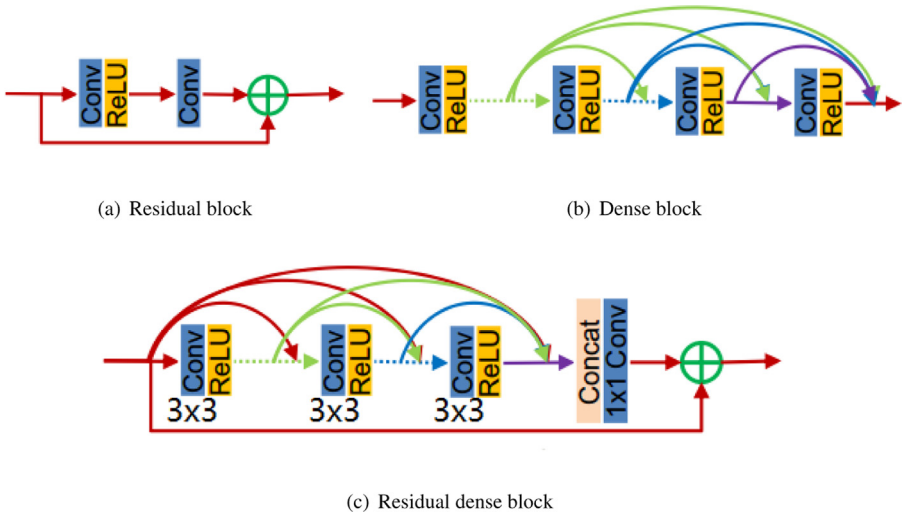
where $R$ is the ReLU activation function and $Y_d^{r-1}$ shows the output of the convolution layer and the input of the first layer RDB concurrently, $Y_d^r$ serves as its corresponding output, in which $Y_d^{r-1}$ and $Y_d^r$ contain the same number of feature maps to guarantee the structural consistency between the residuals and the original sample images during the training process, and d denotes the number of network layers. $\Gamma$ serves as a function of residual learning containing nonlinear mappings. The information filter is designed to effectively filter useless information, with the aim of lowering output dimensionality and enhancing feature fusion. The principle of the information filter will be discussed below.

Fig. 7 shows a schematic diagram of the network structure for the RDBs in this paper, where $F_0$ symbolizes feature.

With the dense connection structure shown in Fig. 7, it can minimize the information loss in each layer feature extraction process, yet increase the computational complexity simultaneously. Accordingly, the information filter designed at the terminal of the residual block is effective in reducing the computational complexity while retaining useful information. The functioning principle of the information filter can be expressed as

$$F_d = \gamma_d[Y_d^1, Y_d^2, \ldots, Y_d^{16}], \tag{12}$$

where $\gamma_d$ is a Conv operation with a kernel of $1 \times 1$ and $F_d$ is the feature output filtered by $\gamma_d$ information. Typically, the convolution kernels used in convolutional operations are in the form of $3 \times 3$, $5 \times 5$ etc. $2n + 1$, whereas $1 \times 1$ convolution kernel is barely used. Nonetheless,




(a) Residual block (b) Dense block

(c) Residual dense block

**Fig. 6.** Schematic diagram of RDB structure: (a) Residual block in MDSR (Lim et al., 2017), (b) Dense block in SRDenseNet (Tong et al., 2017), (c) Zhang's RDB (Zhang et al., 2018).

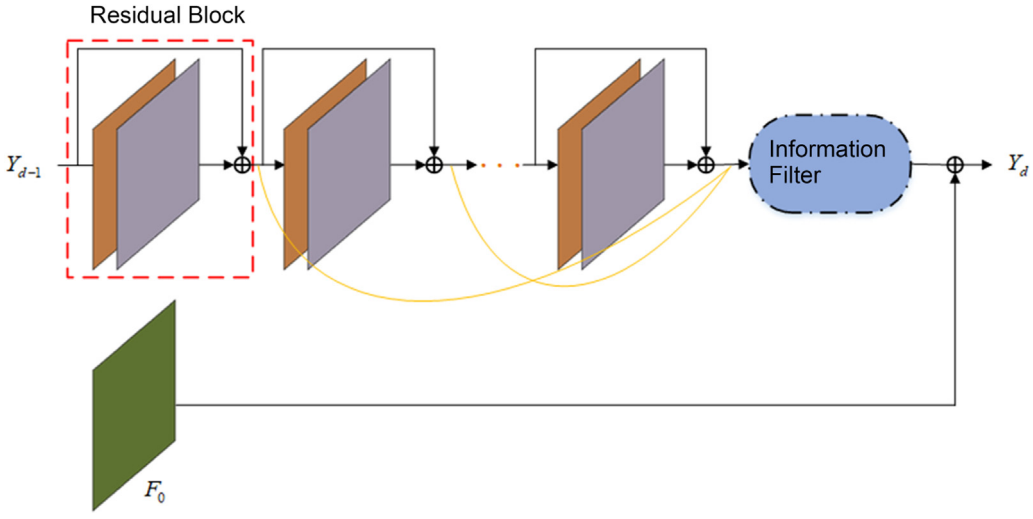



**Fig. 7.** Schematic diagram of the RDBs network structure.

$1 \times 1$ convolution kernels are employed for feature extraction in both the Bottleneck module of the ResNet and the Inception module of the GoogleNet (Szegedy et al., 2015). It can effectively reduce the network computation without compromising the accuracy.

The principle is shown in Fig. 8, where (a) shows the application principle block diagram of the $1 \times 1$ convolution kernel of the residual module, and (b) shows the application principle block diagram of the $1 \times 1$ convolution kernel of the Inception module.

The main function of the convolution kernel with $(2n+1)\times(2n+1)$ size is to reduce the image size while performing feature extraction. Instead, the use of a $1 \times 1$ convolution kernel will not alter the image size, but will increment or decrement the dimensionality of the image features. The use of $1 \times 1$ convolution kernel in this information filter can accomplish the objectives of preserving important original information and eliminating useless information.

### 3.3. Second stage

With respect to the design of the super-resolution reconstruction algorithm in this paper, 16 RDBs will yield 16 image features upon completion of the first stage, otherwise known as pseudo-high-frequency information. In the second stage, the process of residual relearning is to fuse the pseudo-high-frequency information corresponding to each RDB in the first stage with the original high-resolution image element by element, followed by Dilated Convolution (DC) to acquire a high-resolution image with abundant details.

#### 3.3.1. Dilated convolution

Within the classical Convolutional Neural Network (CNN) structure, feature extraction is primarily undertaken in the convolutional and downsampling layers, which also constitute the most vital components of the CNN. In general, regarding image feature extraction, it is possible to obtain superior results by solely relying on stacked convolution and downsampling layers, such as the multiple Visual Geometry Group (VGG) (Simonyan & Zisserman, 2014). By extracting and aggregating features in the image classification task and eventually outputting the results through the fully connected layer, such a structure not only offers the network model with translation and no distortion, but also delivers high computational efficiency of the algorithm. However, for tasks such as image detection and segmentation that need to be performed by extracting features from the last layer of the network, the results of such multi-layer convolution and downsampling will require a large receptive field to fulfill the requirements. On the other hand, downsampling in convolutional networks can lead to the loss

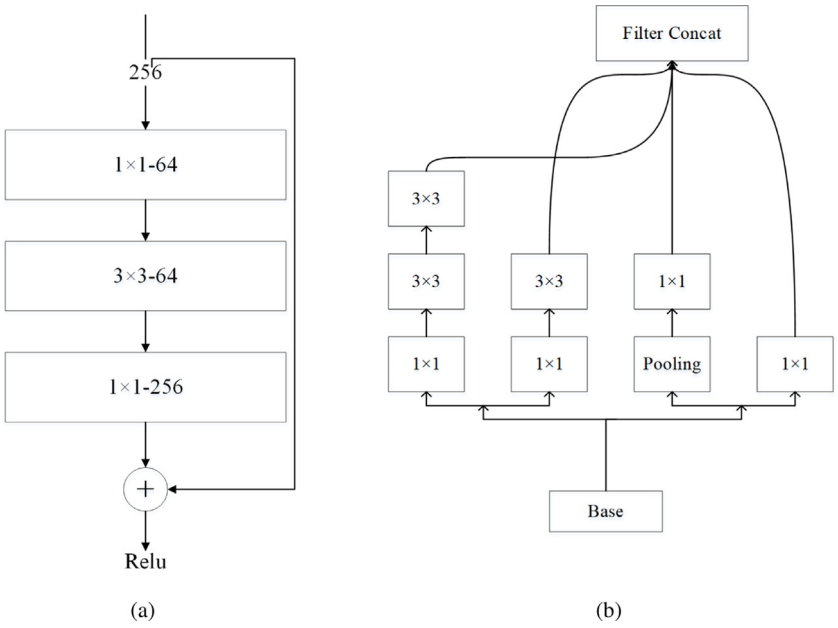



**Fig. 8.** Schematic diagram of 1 × 1 convolution kernel functioning principle.

of small features. Yet, the receptive field cannot be enlarged without downsampling. Increasing the receptive field will significantly increase the data volume of the network, which is computationally intensive and inefficient. Dilated convolution is capable of increasing the size of the receptive field without performing downsampling operations.

Dilated convolution (Yu & Koltun, 2016) is also referred to as *à trous* convolution or inflation convolution. Differing from the typical convolution, the convolution kernel of the dilated convolution is identical, but the receptive field of the convolution is increased. To a certain extent, it addresses the increased computational complexity arising from the large receptive fields in typical convolution. Fig. 9 demonstrates the schematic diagram of dilated convolution.

The dilated convolution receptive field is expressed as

$$r_n = r_{n-1} + d(k_n - 1)\prod_{i=1}^{n-1} s_i, \tag{13}$$

where $r_n$ denotes the size of the $n$th layer receptive field, $r_{n-1}$ denotes the size of the $(n-1)$th layer receptive field, $k_n$ indicates the size of the kernel, d indicates the dilation rate and $\prod_{i=1}^{n-1} s_i$ represents the cumulative offset of kernel's outward dilation.

In conjunction with Fig. 9 and (13), the dilated convolution can be interpreted as follows: Figure (a) shows the basic convolution kernel of size 3 × 3, and Figures (b) and (c) both represent the new convolution kernels created by incorporating the dilation rate into the basic convolution kernel, with dilation rate=2. At this point, despite the convolution kernel still contains 9 parameters, the kernel size is dilated to 7 × 7 with dilation rate=4 in Figure (c), which is equivalent to a convolution kernel of size 15 × 15. In conclusion: Dilated convolution can achieve a larger convolution receptive field without downsampling, enabling the feature map derived from the convolution operation to still possess a larger range of information. With regard to the super-resolution reconstruction of images, the receptive field size is of great significance for the extraction of high-frequency information. Therefore, the second stage in the modified two-stage residual network in this paper primarily concentrates on the extraction of high-frequency information, with the following two advantages of adopting the structure of dilated convolution in the second stage of the network: (1) The inclusion of the dilated convolutional structure can enhance the conversion quality of pseudo-high frequency information to high frequency information in a two-stage network while retaining a large receptive field; (2) Secondly, the residual information between adjacent RDBs can be made as sparse as possible, which enhances the residual relearning of the network model.

### 3.3.2. Implementation process

The second stage of reconstruction can be subdivided into three processes as shown in Figs. 10(a), (b) and (c). Firstly, the pseudo-high-frequency information is aggregated. Utilizing the aggregated pseudo-high frequency information, feature scaling and dilation convolution are conducted to acquire high frequency information. The reconstruction is accomplished by fusing the high frequency information and the input of low-resolution images. The schematic diagram of the second stage implementation process is displayed in Fig. 10. In process (a), a feature scaling structure proposed by Szegedy et al. (Szegedy et al., 2016) is employed for the network. Following each pseudo-high-frequency information extraction, a feature scaling of 0.1 times is performed to assure the stability of the multilayer convolutional layer network. The set of pseudo-high-frequency information $M$ obtained from RDBs in the first stage is expressed as

$$M = \{F_1, F_2, \ldots, F_d\}_{d=1}^{16}, \tag{14}$$

where $F_d$ is the feature output obtained from (12). Moreover, such feature information set $M$ is sequenced, followed by the accumulation of the features with $d$ and $d-1$ serial numbers from $M$ as the input of the dilated convolution in process (c). Upon feature extraction by process (c) dilated convolution, the result is the conversion from pseudo-high frequency information to the genuine high frequency features corresponding to the $d$th RDBs, which can be represented by

$$\tilde{F}_d = \phi(F_d + S(F_{d-1})), \tag{15}$$

where $S$ denotes the size of the scaling factor and $\phi$ denotes the dilated convolution function. The genuine set of high-frequency features $\tilde{M}$ obtained after conversion can be expressed by

$$\tilde{M} = \{\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_d\}_{d=1}^{16}. \tag{16}$$

The processes (a) and (c) are further illustrated by Fig. 11.

In process (b), it primarily integrates the high-frequency feature sets obtained from the above two processes and outputs them as the high-frequency part of the image, with the consolidation process shown as follows:

$$\tilde{I}_l^{HR} = \Sigma\tilde{M} = \tilde{F}_1 + \tilde{F}_2 + \cdots + \tilde{F}_d. \tag{17}$$

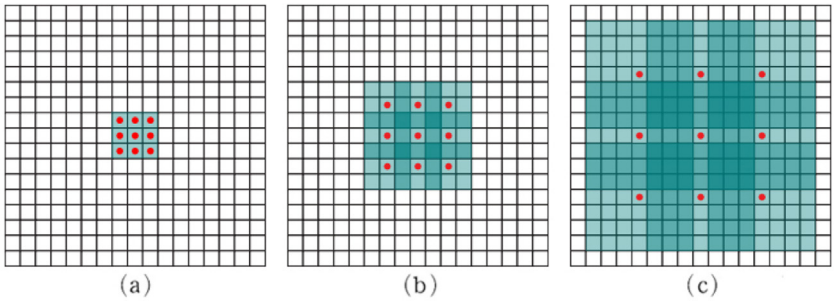



**Fig. 9.** Schematic Diagram of Dilated Convolution Principle (Yu & Koltun, 2016).

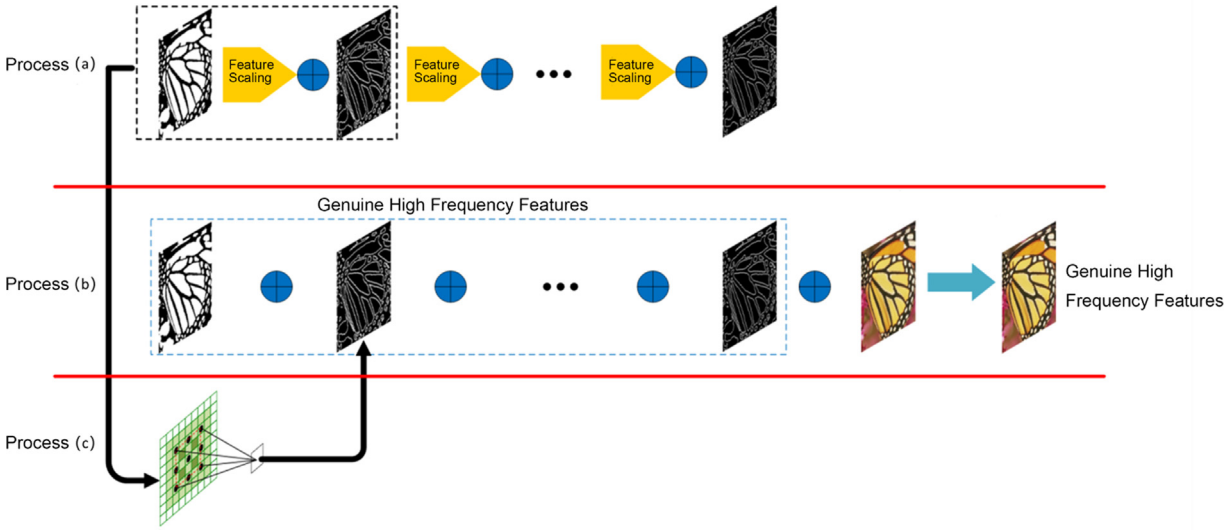



**Fig. 10.** Schematic Diagram of the Second Stage Reconstruction Implementation Process, where the "butterfly" image in the Set5 image set (Bevilacqua et al., 2012) is used.

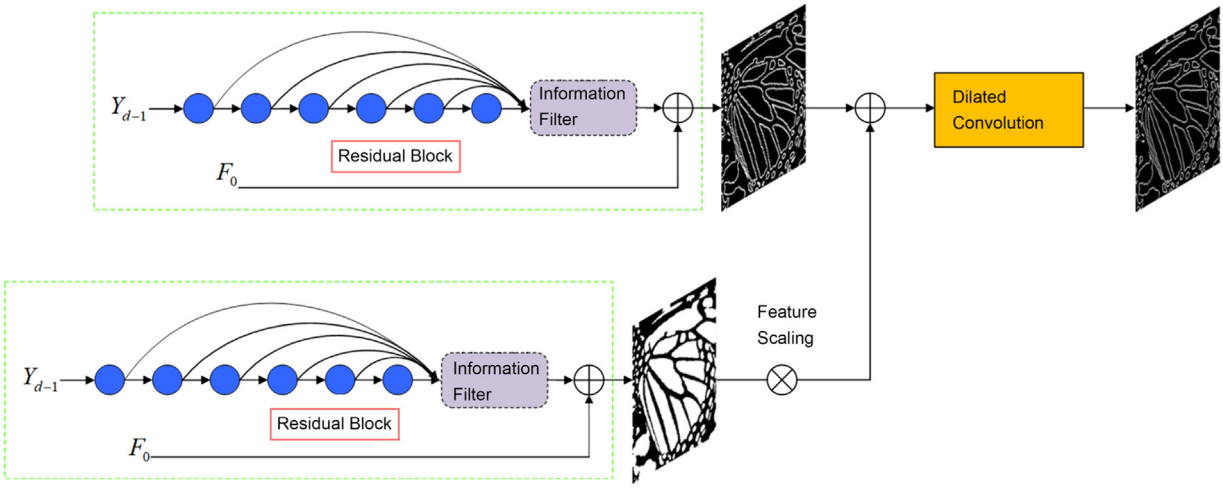



**Fig. 11.** Residual relearning process diagram.

During the final global residual learning process, the genuine high-frequency features are fused with the input image to realize the reconstruction of high-resolution images as follows:

$$I_l^{HR} = I_l^{LR} + \tilde{I}_l^{HR}. \tag{18}$$

### 3.4. Objective function

The objective function refers to a measurement function of the difference between the target output high-resolution image $I_l^{HR}$ and the input low-resolution image $I_l^{LR}$. The super-resolution algorithms such as SRCNN all employ the simple MES as the objective function. The selection of the objective function is of great relevance to the effect of reconstruction. The MSE between a low-resolution image $I^{LR}$ and the corresponding high-resolution image $I^{HR}$ is defined by

$$l_{MSE}^{SR} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}\left(I_{x,y}^{HR} - I_{x,y}^{LR}\right)^2, \tag{19}$$

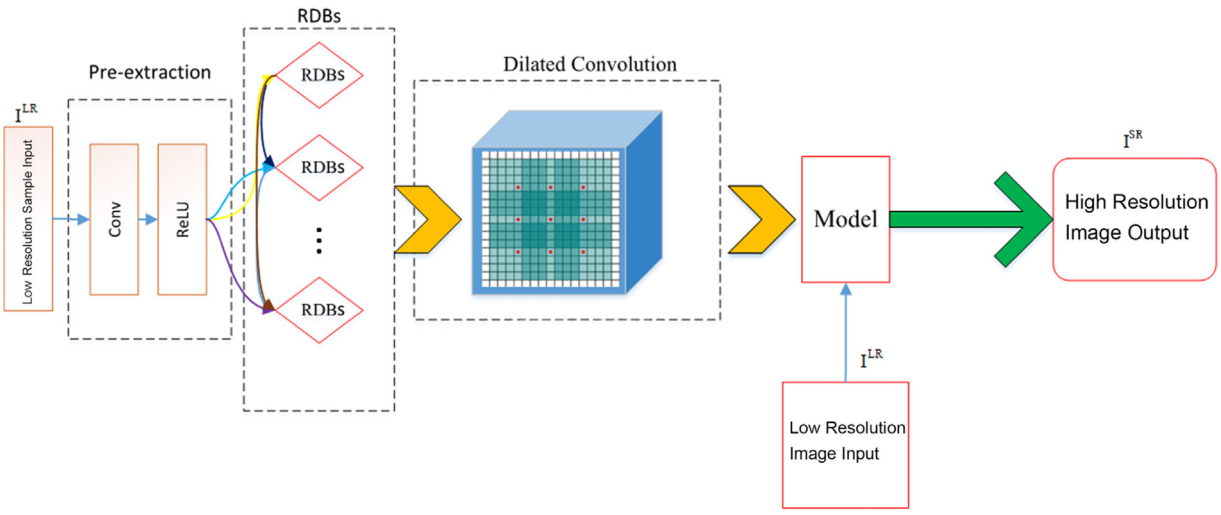
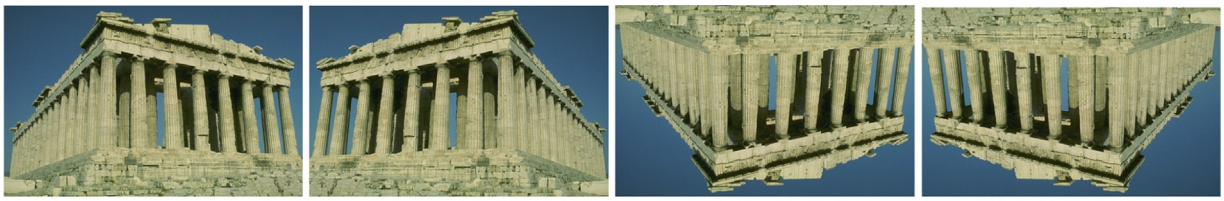



**Fig. 12.** Simulation experiment process diagram.



(a) Original image (b) Mirror image (c) Upside-down of (a) (d) Upside-down of (b)

**Fig. 13.** Sample rotation and mirroring diagram.

where $I_{x,y}^{LR}$ and $I_{x,y}^{HR}$ denote the pixel values at $(x, y)$ in $I^{LR}$ and $I^{HR}$, respectively, for $x = 1, 2, \ldots, rW$ and $y = 1, 2, \ldots, rH$ where $rW$ and $rH$ denote the number of columns and rows in the high-resolution image, and each pixel in the low-resolution image is repeated $r \times r$ times to make both images the same size.

By refining the study of Bruna et al. (Bruna et al., 2016) this paper utilizes a new objective function based on perceptual loss: an objective function consisting of a weighted sum of content loss and adversarial loss. In particular, the content loss is a target function optimized on the basis of the MSE, which is expressed by the Euclidean distance between the reconstructed high-resolution image feature map $G_{\theta_G}(I^{LR})_{x,y}$ and the input low-resolution image feature map $(I^{HR})_{x,y}$ as follows:

$$l_{VGG|i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left[ \phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j} G_{\theta_G}(I^{LR})_{x,y} \right]^2, \tag{20}$$

where $\phi_{i,j}$ denotes the feature map, and $W_{i,j}$ and $H_{i,j}$ represent the aspect of the feature map.

Additionally, the adversarial loss is determined using the probabilistic relationship between the reconstructed high-resolution image $I^{SR}$ and the high-resolution sample image $I^{HR}$ for deep network training as follows:

$$Adv_{loss} = l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}\left( G_{\theta_G}\left( I^{LR} \right) \right), \tag{21}$$

where $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ indicates the probability of $I^{SR}$ with respect to $I^{HR}$, and $-\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$ means the function expression of adversarial loss minimization.

In summary, the loss of the whole network can be expressed as

$$G_{loss} = MSE + VGG_{loss} + Adv_{loss} = l_{MSE}^{SR} + l_{VGG}^{SR} + l_{Gen}^{SR}. \tag{22}$$

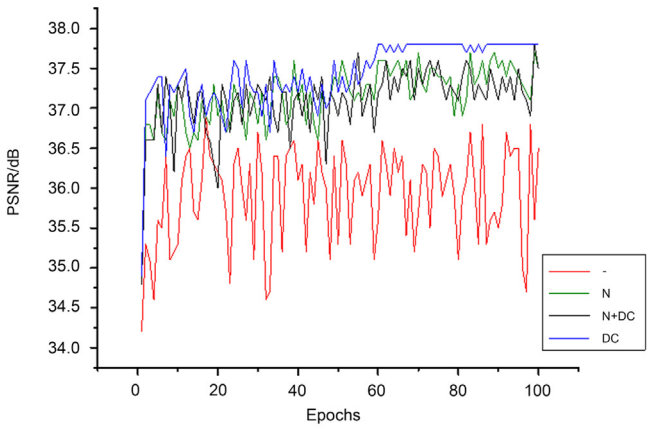
In practical application, for the sake of controlling the network loss within a certain range, the network's target function in this paper is revised on the basis of (21) as

$$G_{loss} = l_{MSE}^{SR} + 2 \times 10^{-6} l_{VGG}^{SR} + 10^{-3} l_{Gen}^{SR}. \tag{23}$$

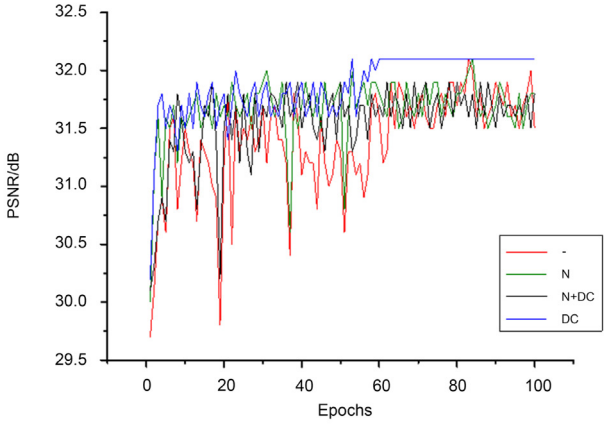
## 4. Experimental results

This section performs experimental validation of the deep learning reconstruction method with the proposed two-stage residuals. In the experiment, the image data downloaded from the GitHub open-source platform is exploited to validate the performance of the proposed super-resolution image reconstruction algorithm. By using low-resolution images for feature training, following the adequate training of the model, the low-resolution images to be reconstructed are imported into the model to accomplish the image resolution reconstruction process, which is illustrated in Figs. 3–11 for the experimental process. It involves comparative experiments containing the necessity of dilated convolutional layers and comparative experimental illustrations with classical reconstruction algorithms. In this paper, the hardware environment in the algorithm simulation experiments remains identical, where the computer processor is configured with Inter(R) Core (TM) i7-8700 3.20 GHz processor, GTX2080Ti graphics card, and the PyTorch deep learning framework is selected for network training. Fig. 12 illustrates the schematic diagram of the simulation experiment process in this paper.

The experiments exploit the BSD100, Set14, Set5, and Urban100 image sets available in the GitHub open-source library (jbhuang0604, 2015) for algorithm validation, of which there are 291 images in total, with 31 being used as sample inputs for model training and 31 as low-resolution image inputs. In particular, within the deep learning algorithm, given a specific number of samples, the more sample images there are, the better the training effect will be. For this reason, the images in the dataset are first preprocessed with Python before the experiments, with the following preprocessing steps:




(a) Set14




(b) BSD100

**Fig. 14.** PSNR comparison schematic on the two datasets.

Step 1: In the case of limited sample size, the rotate algorithm and circular pixel operation in Python are implemented to rotate the sample image by 30°, 60°, 120°, 150°, 190° and 220°, with mirroring and other operations to enlarge the sample data by 8 folds. The schematic diagram of sample rotation and mirroring is presented in Fig. 13.

Step 2: Perform YCbCr format conversion on the RGB format image from Step 1 to obtain three images of $Y, C_b$ and $C_r$ channels. Since $Y = 0.257 \times R + 0.504 \times G + 0.098 \times B + 16$, it can be concluded that the $Y$ channel carries more luminance information which is of more interest to human eyes. Therefore, only the $Y$-channel image is selected as the training input during the algorithm training.

Step 3: By utilizing the *imresize* function of Image Processing Toolbox in MATLAB, the image of Step 2 is degraded to obtain images with 2, 3 and 4 magnifications. Meanwhile, the *imcrop* function is applied to crop the images into 31 × 31 blocks as the input of the training network. To wit, it completes the pre-processing procedures for the tested low-resolution image $I^{LR}$ and the corresponding tagged image.

Following the above preprocessing, in the simulation experiments, only one model needs to be trained to accomplish the reconstruction task of the (x2, x3, x4) scale of the two-stage network. The initial value of bias in each convolutional layer is set to 0. The initialization approaches of the weight values are all adopted from the method developed by Xavier (Bruna et al., 2016) for the purpose of avoiding the exploding gradient problem due to excessive preset learning rate. Moreover, the experiments in this paper employ the VDSR (Bruna et al., 2016) algorithm. In other words, a threshold $g$ is set for the gradient in the model, and when the gradient $g'$ is greater than the set threshold for each model training, the current threshold $g$ is assigned to $g''$, where $g'' \leq g'$.

**Table 2**
PSNR values for Set14 dataset at different epochs.

| Epochs | Models | | | |
|---|---|---|---|---|
| | DC | N+DC | N | – |
| 10 | 37.3 | 37.3 | 37.3 | 35.3 |
| 20 | 37.1 | 36.9 | 36.0 | 36.2 |
| 30 | 37.2 | 36.8 | 37.3 | 36.7 |
| 40 | 37.2 | 37.1 | 37.2 | 36.1 |
| 50 | 37.2 | 37.3 | 36.9 | 35.3 |
| 60 | 37.8 | 37.6 | 37.2 | 35.6 |
| 70 | 37.8 | 37.7 | 37.5 | 35.7 |
| 80 | 37.8 | 37.3 | 37.1 | 35.1 |
| 90 | 37.8 | 37.5 | 37.1 | 35.5 |
| 100 | 37.8 | 37.5 | 37.5 | 36.5 |
| Mean | 37.5 | 37.3 | 37.1 | 35.8 |

For the sake of validating the effect of removing the normalization layer and the addition of the dilated convolutional layer in the improved algorithm, four reconstruction comparison experiments are conducted in a controlled variable manner. Except for the model discrepancies, the four experiments are identical in terms of configuration environment. To be specific, the model in Experiment 1 contains no normalization layer but with a dilated convolution layer symbolized as "DC", the model in Experiment 2 has both a normalization layer and a dilated convolution layer symbolized as "N+DC", the model in Experiment 3 includes a normalization layer but no dilated convolution layer symbolized as "N", and the model in Experiment 4 contains neither a normalization layer nor a dilated convolution layer symbolized as "-". In the experiments, as for the measurement of the reconstruction effect, the peak signal-to-noise ratio (PSNR) of the image quality evaluation index is adopted. The algorithm implementation steps are described as follows:

Step 1: In the pre-processing section, the pre-processed images are acquired as described in the previous section, and the H5 format file is generated.

Step 2: For the training part, firstly, the H5 format file generated in Step 1 is reviewed and extracted to classify the data into training data and test data. Secondly, the convolutional neural network is constructed, including the initial value settings of bias and weights, the creation of activation function ReLU and prediction values, and the parameter settings of loss function and learning rate. Afterwards, the model is stored to local disc after the training is completed.

Step 3: For the testing part, firstly, the model saved in Step 2 is loaded. The result prediction is performed on the test data and the loaded model tagged data through PyTorch function calls, and the reconstructed result map is saved to local disc.

Fig. 14 shows the PSNR comparison graphs for the data sets of Set14 and BSD100 performing the above Experiments 1–4.

As can be observed from Fig. 14 and Tables 2 and 3, following multiple epochs, Experiment 1 (DC) exhibits the smoothest curve with the largest mean value, indicating the best reconstruction effect; Experiment 4 (-) features the greatest curve fluctuation with the least mean value, producing the worst reconstruction effect. The four experimental results described above strongly support the adequacy and necessity of the improvements to the model proposed in this paper.

Furthermore, given the setup of Experiment 1, a reconstruction simulation comparison experiment is carried out in this paper for low-resolution maps by utilizing bicubic interpolation, SRResNet (Ledig et al., 2017) reconstruction algorithm and the algorithm presented in this paper, which is demonstrated in Fig. 15.

In Fig. 15, (a) shows the original high-resolution image, (b) is the low-resolution image, (c) is the effect of bicubic interpolation, and (d)

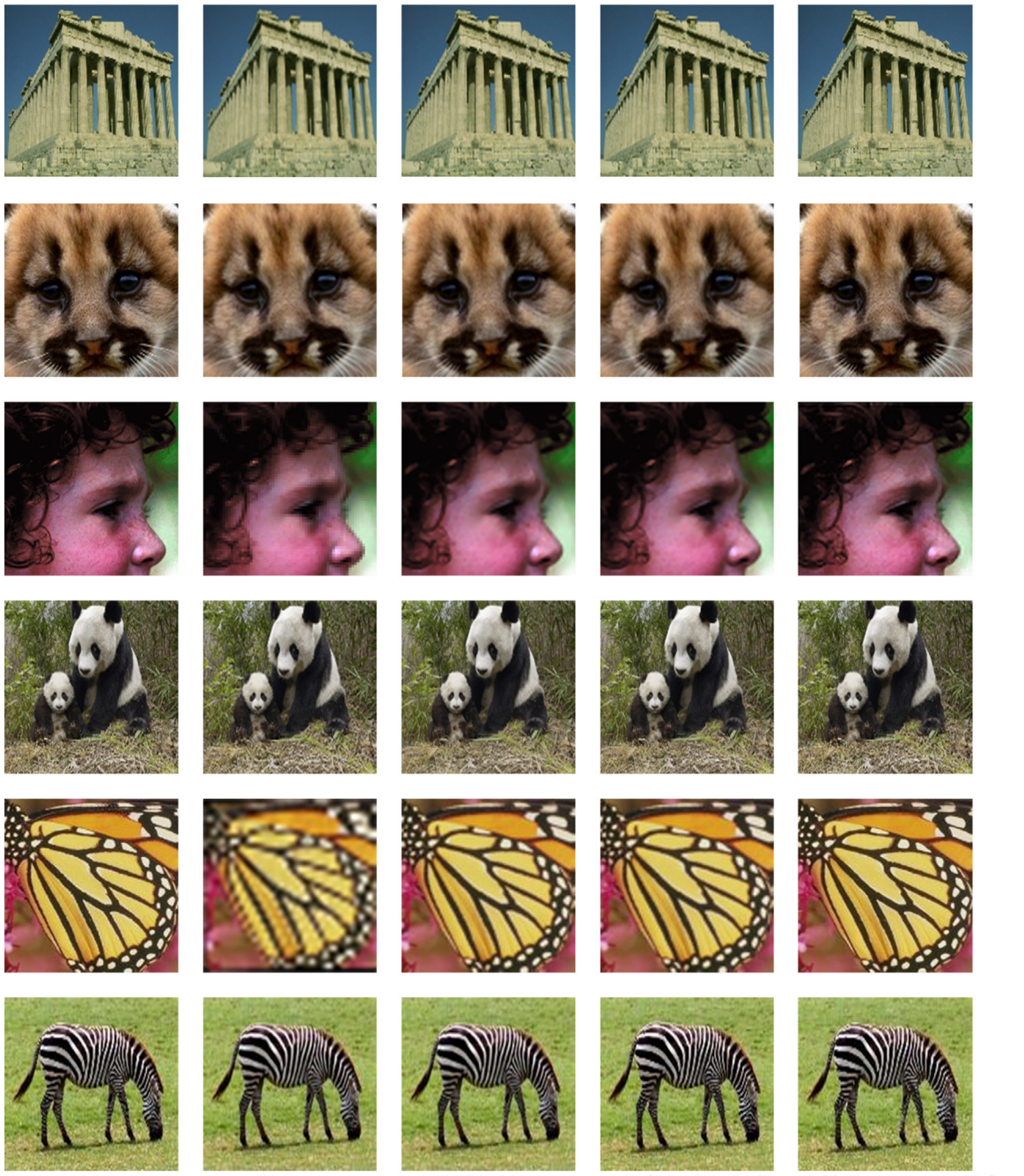


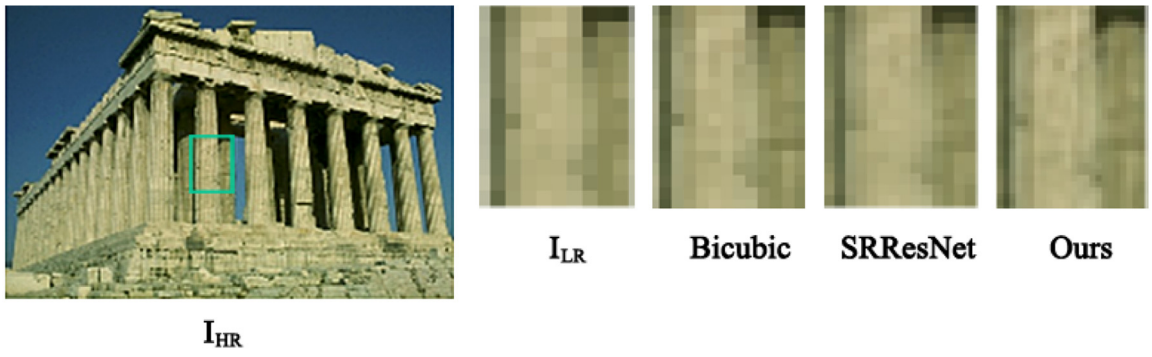

**Fig. 15.** Reconstruction effect comparison.

represents the effect of SRResNet (Ledig et al., 2017) algorithm, which is also the effect of the reconstruction algorithm in this paper. Fig. 16 shows the partial enlargement effect of the first three comparison figures of Fig. 15. From the above two sets of effect comparison figures, it can be seen that, the reconstruction algorithm in this paper tends to be more realistic in effect in comparison with the algorithms in plots (c) and (d), implying that the algorithm presented in this paper achieves the enhancement in the reconstruction effect to some extent.

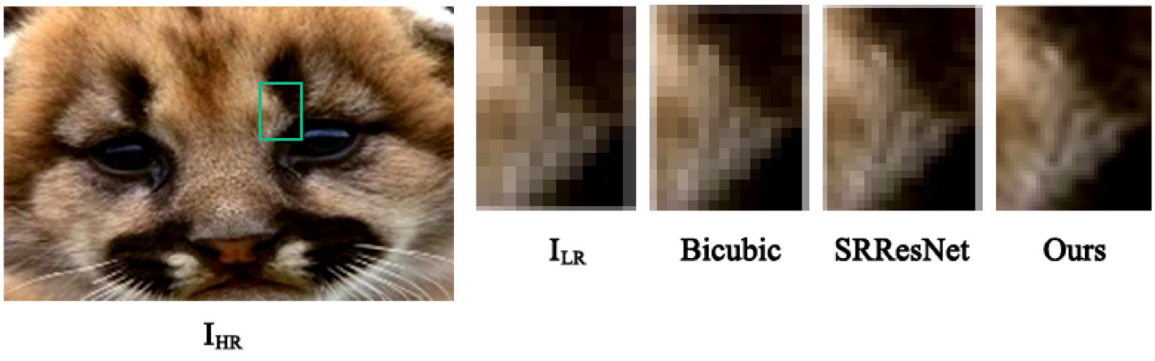
For the purpose of better demonstrating the effectiveness of the algorithm presented in this paper, apart from the reconstruction shown in Fig. 15, this paper also conducts reconstruction tests with up-sampling factors of 4 and 8 on the datasets Set5, Set14, BSDS100, Urban100, and Manga109 through the use of deep learning reconstruction algorithms such as SRGAN (Ledig et al., 2017), EDSR (Lim et al., 2017), ESPCN (Shi et al., 2016), and SRDenseNet (Tong et al., 2017). The results are summarized in Table 4, the algorithm in this paper demonstrates a certain improvement regarding the integrated score in image quality evaluation compared to other deep learning algorithms. As summarized in the rightmost column in Table 4, our method achieved the highest PSNR/SSIM values on average for both scales among the compared methods.

## 5. Discussion

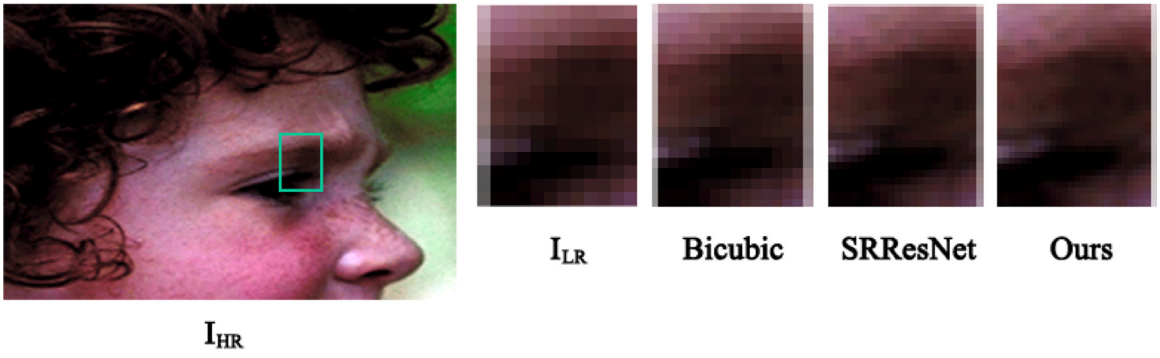
The main improvement point proposed in this paper is to extend the network model into two stages. In the first stage, the high-frequency information is extracted from the low-resolution images using RDBs, thus reducing the loss of high-frequency information, and the batch normalization in the network structure is removed to accelerate the




(a) Parthenon




(b) Dog




(c) Child

**Fig. 16.** Partial enlargement effect.

**Table 3**
PSNR values for BSD100 dataset at different epochs.

| Epochs | Models | | | |
|---|---|---|---|---|
| | DC | N+DC | N | – |
| 10 | 31.5 | 31.3 | 31.5 | 31.5 |
| 20 | 31.8 | 31.7 | 31.5 | 31.2 |
| 30 | 31.8 | 31.9 | 31.7 | 31.7 |
| 40 | 31.9 | 31.7 | 31.5 | 31.1 |
| 50 | 31.7 | 31.9 | 31.9 | 31.3 |
| 60 | 32.1 | 31.8 | 31.7 | 31.6 |
| 70 | 32.1 | 31.8 | 31.6 | 31.7 |
| 80 | 32.1 | 31.8 | 31.9 | 31.7 |
| 90 | 32.1 | 31.7 | 31.5 | 31.9 |
| 100 | 32.1 | 31.8 | 31.8 | 31.5 |
| Mean | 31.9 | 31.7 | 31.7 | 31.5 |

algorithm's computational speed. In the second stage, we first perform residual relearning on the high-frequency information extracted and learned in the first stage. Then, since the size of the receptive field plays an important role in the extraction of high-frequency information, we add dilated convolution in the second stage to increase the receptive field and also improve the quality of the pseudo-high-frequency information converted to high-frequency information in the first stage so as to improve the reconstruction effect of the image.

From the above experimental results, it is known that the two improvements of removing the batch normalization and adding the dilated convolution layer substantially improve the model's reconstruction effect for images under the validation experiments using the control variables method. Meanwhile, the excellent performance of the proposed model on image super-resolution reconstruction is demonstrated in the comparison with Bicubic and SRResNet networks under the same conditions as shown in Table 4. The reliability of the

**Table 4**
Mean comparison of PSNR and SSIM of the algorithm in this paper versus.

| Algorithms | Scale | Set5 PSNR/SSIM | Set14 PSNR/SSIM | BSDS100 PSNR/SSIM | Urban100 PSNR/SSIM | Manga109 PSNR/SSIM | Mean PSNR/SSIM |
|---|---|---|---|---|---|---|---|
| Bicubic | | 27.97/0.791 | 25.83/0.712 | 25.49/0.664 | 23.03/0.612 | 24.15/0.786 | 25.28/0.713 |
| SRResNet (Ledig et al., 2017) | | 32.74/0.897 | 28.52/0.739 | 27.48/0.722 | 26.13/0.727 | 30.50/0.990 | 29.07/0.815 |
| SRGAN (Ledig et al., 2017) | | 29.57/0.829 | 26.92/0.723 | 25.87/0.673 | 23.97/0.719 | 28.05/0.854 | 26.88/0.760 |
| EDSR (Lim et al., 2017) | ×4 | 32.56/0.791 | 28.80/0.788 | 27.71/0.781 | 26.48/0.803 | 31.09/0.974 | 29.33/0.827 |
| ESPCN (Shi et al., 2016) | | 29.21/0.848 | 27.73/0.801 | 27.35/0.728 | 26.39/0.693 | 23.23/0.798 | 26.78/0.774 |
| SRDenseNet (Tong et al., 2017) | | 21.86/0.881 | 28.43/0.776 | **28.55/0.709** | 25.67/0.714 | 29.05/0.899 | 26.71/0.796 |
| Ours | | **33.32/0.925** | **28.99/0.812** | 28.42/0.766 | **28.11/0.809** | **31.78/0.998** | **30.12/0.862** |
| Bicubic | | 24.43/0.632 | 23.12/0.585 | 23.66/0.543 | 20.02/0.511 | 21.11/0.598 | 22.47/0.574 |
| SRResNet (Ledig et al., 2017) | | 26.54/0.757 | 24.68/0.638 | 24.65/0.625 | 23.34/0.531 | 24.68/0.761 | 24.78/0.662 |
| SRGAN (Ledig et al., 2017) | | 25.97/0.791 | 23.99/0.673 | 21.78/0.441 | 22.68/0.506 | 23.52/0.703 | 23.59/0.623 |
| EDSR (Lim et al., 2017) | ×8 | 27.01/0.634 | 24.89/0.658 | 25.12/0.669 | 24.02/0.620 | 22.08/0.616 | 24.62/0.639 |
| ESPCN (Shi et al., 2016) | | 25.32/0.687 | **27.37/0.711** | 25.01/0.602 | 21.09/0.512 | **25.57/0.743** | 24.87/0.652 |
| SRDenseNet (Tong et al., 2017) | | 26.11/0.733 | 23.47/0.598 | 24.52/0.530 | 22.98/0.582 | 24.91/0.764 | 24.40/0.641 |
| Ours | | **27.25/0.712** | 26.94/0.765 | **25.19/0.702** | **24.21/0.617** | 24.97/0.792 | **25.71/0.718** |

proposed two-stage model based on SRResNet is also demonstrated in Tables 2 and 3, and Fig. 14. Finally, in the comparison with other neural networks, the proposed model in this paper also has significant improvement in the evaluation of image quality as shown in Table 4 and Figs. 15 and 16.

Meanwhile, according to the above experimental results, it can be found that the model proposed in this paper also obtains excellent results after being pre-trained on three training sets and validated on other different data sets. This fully illustrates the good generalizability of the model proposed in this paper.

## 6. Conclusions

This paper firstly discusses the advantages and disadvantages of the existing common deep learning super-resolution reconstruction methods, which brings out the idea of the deep learning super-resolution reconstruction method based on two-order residuals in this paper. Secondly, the algorithm structure proposed in this paper is elaborated, including the concept of residuals, the principle of residual dense block (RDB), the implementation process of two reconstruction stages, the selection of objective optimization function and the training process. Eventually, the improved algorithm described in this paper is evaluated in simulation experiments.

Drawing on the SRResNet algorithm, an improved reconstruction network with two-stage residuals is introduced in this paper. Building on the idea of residual learning, this network effectively avoids gradient dispersion and accuracy degradation while deepening the number of network layers for the purpose of deep feature extraction. Additionally, the preliminary feature extraction of the input low-resolution image is performed by the local residual learning of the RDBs in the first stage to acquire the pseudo-high frequency information. In the second stage, dilated convolution fusion is performed by incorporating the input image and pseudo-high frequency information to achieve the objective of global residual learning. Detailed features of the image can be effectively extracted in the two-stage feature learning, and a superior reconstruction effect can be fulfilled.

In a word, this paper is a research on super-resolution reconstruction supported by international literature and web resources, and has gained some achievements in reconstruction effect and algorithm speed. Nonetheless, there are still many shortcomings and areas in need of in-depth study concerning the study of deep learning-based super-resolution reconstruction techniques. In this paper, despite the improved effect of the two-stage residual reconstruction network over the previous algorithms, the network is extended to a two-stage operation, which leads to a large number of network parameters and fails to achieve the real-time reconstruction effect. As a result, future research efforts should be devoted to simplifying the network and realizing the reconstruction process in real time while safeguarding the reconstruction effect.

## CRediT authorship contribution statement

**Lin Dong:** Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Kohei Inoue:** Writing – review & editing, Supervision, Funding acquisition.

## Declaration of competing interest

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

## Acknowledgment

This work was supported by JSPS, Japan KAKENHI Grant Number JP21K11964.

## References

Altan, Aytaç, & Karasu, Seçkin (2019). The effect of kernel values in support vector machine to forecasting performance of financial time series and cognitive decision making. *The Journal of Cognitive Systems*, (1), 17–21.

Altan, Aytaç, Karasu, Seçkin, & Zio, Enrico (2021). A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Applied Soft Computing*, http://dx.doi.org/10.1016/j.asoc.2020.106996.

Bevilacqua, Marco, Roumy, Aline, Guillemot, Christine, & line Alberi Morel, Marie (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the british machine vision conference* (pp. 135.1–135.10). BMVA Press, http://dx.doi.org/10.5244/C.26.135.

Bruna, Joan, Sprechmann, Pablo, & LeCun, Yann (2016). Super-resolution with deep convolutional sufficient statistics. http://arxiv.org/abs/1511.05666.

Dong, Chao, Loy, Chen Change, He, Kaiming, & Tang, Xiaoou (2014). Learning a deep convolutional network for image super-resolution. In David Fleet, Tomas Pajdla, Bernt Schiele, & Tinne Tuytelaars (Eds.), *Computer vision – ECCV 2014* (pp. 184–199). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-10593-2_13.

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307. http://dx.doi.org/10.1109/TPAMI.2015.2439281.

Elad, M., & Feuer, A. (1996). Super-resolution reconstruction of an image. In *Proceedings of 19th convention of electrical and electronics engineers in Israel* (pp. 391–394). http://dx.doi.org/10.1109/EEIS.1996.566997.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *2015 IEEE international conference on computer vision (ICCV)* (pp. 1026–1034). http://dx.doi.org/10.1109/ICCV.2015.123.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). http://dx.doi.org/10.1109/CVPR.2016.90.

Ioffe, Sergey, & Szegedy, Christian (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach, & David Blei (Eds.), *Proceedings of machine learning research*: *Vol. 37*, *Proceedings of the 32nd international conference on machine learning* (pp. 448–456). Lille, France: PMLR, URL http://proceedings.mlr.press/v37/ioffe15.html.

jbhuang0604 (2015). SelfExSR/data at master . jbhuang0604/SelfExSR . GitHub. URL https://github.com/jbhuang0604/SelfExSR/tree/master/data.

Karasu, Seçkin, Altan, Aytaç, Bekiros, Stelios, & Ahmad, Wasim (2020). A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. *Energy*, *212*(C), S0360544220318570, URL https://EconPapers.repec.org/RePEc:eee:energy:v:212:y:2020:i:c:s0360544220318570.

Karasu, Seçkin, Altan, Aytaç, Saraç, Zehra, & Hacioğlu, Rifat (2018). Prediction of bitcoin prices with machine learning methods using time series data. In *2018 26th signal processing and communications applications conference (SIU)*. http://dx.doi.org/10.1109/SIU.2018.8404760.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. (Master's Thesis), University of Tront, URL https://ci.nii.ac.jp/naid/20001706980/.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 105–114). http://dx.doi.org/10.1109/CVPR.2017.19.

Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 1132–1140). http://dx.doi.org/10.1109/CVPRW.2017.151.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., & Fei-Fei, Li (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Sezer, Ali, & Altan, Aytaç (2021). Detection of solder paste defects with an optimization-based deep learning model using image processing techniques. *Soldering & Surface Mount Technology*, http://dx.doi.org/10.1108/SSMT-04-2021-0013.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1874–1883). http://dx.doi.org/10.1109/CVPR.2016.207.

Simonyan, Karen, & Zisserman, Andrew (2014). Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556.

Szegedy, C., Liu, Wei, Jia, Yangqing, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–9). http://dx.doi.org/10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826). http://dx.doi.org/10.1109/CVPR.2016.308.

Tai, Y., Yang, J., Liu, X., & Xu, C. (2017). Memnet: A persistent memory network for image restoration. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 4549–4557). http://dx.doi.org/10.1109/ICCV.2017.486.

Tong, T., Li, G., Liu, X., & Gao, Q. (2017). Image super-resolution using dense skip connections. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 4809–4817). http://dx.doi.org/10.1109/ICCV.2017.514.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J., & Liao, Q. (2019). Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, *21*(12), 3106–3121. http://dx.doi.org/10.1109/TMM.2019.2919431.

Yu, Fisher, & Koltun, Vladlen (2016). Multi-scale context aggregation by dilated convolutions. In *International conference on learning representations (ICLR)*.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 2472–2481). http://dx.doi.org/10.1109/CVPR.2018.00262.