RBDN: Residual Bottleneck Dense Network for Image Super-Resolution

Zeyu An

Southwest University

Junyuan Zhang

Southwest University

Ziyu Sheng

Southwest University

Xuanhe Er

Chengdu University of Information Technology

Junjie Lv (✉ messi10@email.swu.edu.cn)

Southwest University

Research

Keywords: Single-image Super-resolution, Features Utilization, Scalability, Network Parameters

Posted Date: March 25th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-309485/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

RBDN: Residual Bottleneck Dense Network for Image Super-Resolution

Zeyu An · Junyuan Zhang · Ziyu Sheng · Xuanhe Er ·
Junjie Lv[✉]

Received: date / Accepted: date

Abstract Recent studies have shown that Super-Resolution Generative Adversarial Network (SRGAN) can significantly improve the quality of single-image super-resolution. However, the existing SRGAN approaches also have drawbacks, such as inadequate of features utilization, huge number of parameters and poor scalability. To further enhance the visual quality, we thoroughly study three key components of SRGAN: network architecture, adversarial loss and perceptual loss, and propose a DenseNet with Residual-in-Residual Bottleneck Block (RRBB) named as Residual Bottleneck Dense Network (RBDN) for single-image super-resolution. In particular, RBDN combines ResNet and DenseNet with different roles, in which ResNet refines feature values by addition and DenseNet memorizes feature values by concatenation. Specifically, the DenseNet adopts the Bottleneck structure to reduce the network parameters and improve the convergence rate. In addition, the proposed RRBB, as the basic network building unit, removes the batch normalization (BN) layer and employs the ELU function to reduce the opposite effects in the absence of BN. In this way, RBDN can enjoy the merits of the sufficient feature value refined by residual groups and the refined feature value memorized by dense connections, thus achieving better performance compared with most current residual blocks.

1 Introduction

Image super-resolution (SR) techniques reconstruct a higher-resolution (HR) image or sequence from the observed lower-resolution (LR) images. Usually the benchmarks are single-image super-resolution (SISR) tasks. It's a well-known ambiguous problem, since there are always multiple HR images corresponding to a single LR image. With the development of deep learning in recent years, many methods have been proposed to solve this problem. Dong et al. [1, 2] made the first successful attempt, they proposed a three-layer Super-Resolution Convolution Neural Network (SRCNN) and which achieved better performance than the traditional methods. After that, Kim et al. [3, 4] improved the network depth of Super-Resolution using Very Deep Convolutional Networks (VDSR) and Deeply-Recursive Convolution Network for image Super-Resolution (DRCN) to 20 levels, and eased the training difficulty by introducing residual learning. After these pioneering work, many Convolutional Neural Network (CNN)-based methods [10, 12, 22, 23, 24, 25, 26, 27, 28] were proposed, and these methods achieved high performance, especially in peak signal-to-noise ratio (PSNR) and Structural SIMilarity (SSIM) metrics. Among the many solutions proposed in the past, most optimization schemes calculate the pixel distance between LR and HR images through root mean square error, that is, MSE. However, PSNR tends to output smooth results without enough details, because the PSNR metric is fundamentally different from the subjective evaluation of human vision. Therefore,

Z. An, J. Zhang, and J. Lv
College of Electronics and Information Engineering, Southwest University, Chongqing 400715, P. R. China.
E-mail: messi10@email.swu.edu.cn

X. Er
College of Software Engineering, Chengdu University of Information Technology, Chengdu 610200, P. R. China

Z. Sheng
The Key Laboratory of Networks and Cloud Computing Security of Universities in Chongqing, Chongqing 400715, P. R. China.

Z. An and J. Zhang contributed to the article equally, and the order of authorship was determined by coin toss.

in order to restore realistic images, several perceptual driven methods are proposed to improve the visual quality of the final result, such as the introduction of perceptual loss [13, 14]. Later, a variety of networks based on GAN networks that use residual blocks and pursue visual quality are proposed such as Super-Resolution Using a Generative Adversarial(SRGAN) and further Enhanced Super-Resolution Generative Adversarial Networks(ESRGAN), Recurrent generative adversarial Networks(RGAN), etc. Although GAN-based methods can produce high fidelity of the output we want, but due to the more the number of parameters of the network, the remaining characteristics of the low utilization rate and some inherent problems existing in the network layer structure, the SR reconstruction task is still a long way to go.

SRGAN method carried SR reconstruction task to a new level, SRGAN ensures that the PSNR and other indicators remain at a high level while the image texture details of a higher standard can be recovered. However, there is still a difference in perception between the image recovered by SRGAN and the image in the real world. Nevertheless, the appearance of SRGAN still opens a new and valuable way for super-resolution reconstruction. In recent years, a variety of different methods based on SRGAN network structure have emerged, especially ESRGAN has been able to restore the reconstructed image to a very high level [15]. The feature re-use in this network plays a great role, which also provides inspiration for our work. However, the introduction of feature multiplexing (the idea of DenseNet) [19] also brings the disadvantages of increasing computation and network parameters.

In recent years, more and more models have used Residual Network(ResNet) as their own network architecture, and these models have shown good generalization performance, for example, Ledig et al. [5] came up with Super-Resolution Residual Network(SRResNet) using the idea of ResNet. One of the reasons that make ResNet exceptionally popular is the simple design strategy which introduces only one identity shortcut. The identity shortcut skips the residual blocks to preserve features, as a result limiting the representation power of the network [39, 40]. The drawback of ResNets is that it causes the collapsing domain problem which reduces the learning capacity of the network [41] and [40] proposed to mitigate it with non-linear shortcuts. Another simple but effective technique, dense connectivity, is proposed in DenseNet [19] to facilitate the training of deep networks. DenseNet uses dense cascading for all subsequent layers to avoid directly adding and preserve the characteristics of the previous layer, and has been proven to be more functionally efficient to use [42]. Nevertheless, DenseNet requires a lot of GPU memory, which is more complex from an engineering point of view, and it further increases DenseNet’s training time [42]. The main reason DenseNet needs more training time is that DenseNet uses dense cascading in the network, which increases the computational load of the whole network, and in short, ResNet and DenseNet’s choice of performance and GPU resources is a dilemma for a wide range of applications.

In this work, motivated by the recent advances in deep learning and overcoming the shortcomings in these network architectures, we proposed a dense network with Residual-in-residual Bottleneck Block (RRBB) named as RBDN for single-image super-resolution. In RBDN, we not only introduced the dense connections in the original network structure, but also introduced the bottleneck structure in order to reduce the huge amount of computing in the network and accelerate the convergence speed. At the same time, inspired by Lim [12] and Wang [15], we delete the BN layer in the original network structure. Although BN can speed up the training and convergence speed, and prevent problems such as gradient disappearance and overfitting, the BN layer will lead to our final output results accompanied by artifacts. In order to obtain the gain brought by removing the BN layer to the whole network and reduce the negative effect brought by removing the BN layer, we will replace the original ReLU function as ELU function, ELU function can not only relieve the gradient disappeared, can also make the convergence speed is faster, ELU function like BN layer average data can be more close to 0, and the computation complexity is lower than mass regularization. The ELU function has a negative component compared to the ReLU function, and it’s robust to noise compared to LReLU and PReLU. Therefore, our work further improves the GAN-based image super-resolution models through the above improved method.

In order to verify the effectiveness of our method, we applied the above ideas to the network structure we designed and combined them with the antagonistic generation network in [19] to form the RBDN, so as to adapt to the real world and reach a higher level of network structure, as show in Fig. 1. In summary, the main contributions of this paper are as follows:

- In order to maximize the information flow throughout the network, we adopt a dense connection in RBDN, similar to DenseNet structure, which not only alleviates the vanishing-gradient problem, but also strengthens the feature propagation based on a small amount of convolution kernel due to encouraging feature reuse.
- To achieve lightweight networks, we introduced the Bottleneck structure, which tends to greatly reduce the actual amount of computing in networks and improve network computing efficiency, and which tend to have better performance than the usual structure of just 3×3 convolutional layers.

Fig. 1 Visual comparison of our model with other state-of-art models on Set5/Set14 at the $\times 4$ super-resolution

- We propose the RBDN, which is generated by combining the aforementioned schemes. We combine the improved residual block with the network architecture in [19], which can not only improve the performance of the network, but also restore the output results more appropriate to the real world.

2 Related Works

Super-resolution reconstruction methods can be divided into two categories: traditional methods, such as sparse coding, and the deep learning-based approach that has become popular in recent years. Due to the strong learning ability of deep learning method, traditional learning methods are gradually being replaced by methods based on deep learning. Therefore, we focus on using deep neural network to solve the super-resolution reconstruction problem. The pioneer in this field is Dong et al [1, 2]. They proposed a shallow rerouted neural network (SRCNN) for SISR tasks to learn the mapping of LR-HR images in an end-to-end manner that achieved better performance than previous work, and later emerged in this area a variety of network architectures that have been used to this day, such as residual blocks, densely connected networks [9], and residual dense networks [10]. Especially Lim et al. [12] whose EDSR network model first proposed the idea of removing the BN layer from the network structure. Ledig et al. [5] came up with SRResNet for SR tasks using the idea of ResNet [6]. Zhang et al. [10] proposed residual dense blocks for SISR reconstruction, and proposed a deep network of attention channel mechanism, so that PSNR reached a fairly high level. However, we can also note that although the aforementioned model can get very ideal PSNR indicators, but in the actual application of the image they recovered there will be distortion indicators and perceived quality contradictions, image edge sharpening and other issues, resulting in the actual recovery of the picture effect is not in line with our expectations.

Today, perception-driven algorithms are mature and used to improve the visual quality of SR results. Based on the idea of proximity to perceived similarity [13, 14], perceived loss is raised to enhance visual effects by minimizing perceived loss. Ledig et al. [5] proposed SRGAN, which achieves multi-image output by perceived loss and counter loss. How SRGAN works also facilitates what we do next. On the basis of SRGAN, ESRGAN [15] came into being, which introduced dense residual blocks (RRDBs) into the original structure and removed the BN layer [19, 20]. At the same time, the perceived loss was adjusted properly. The advent of ESRGAN allowed the super-resolution reconstruction to reach a higher level.

In the development process of SR reconstruction method based on deep learning, many network architectures appear as the above mentioned deep network. However, it can also be noted that although the above models can obtain very ideal PSNR or SSIM indicators in the actual application process, the images restored by them will have problems such as perceived quality, image edge sharpening and so on. The image effect that resulted in the actual recovery did not meet our expectations.

The pros and cons of the SR algorithms are usually evaluated using several widely used distortion-based metrics, such as PSNR and SSIM. Combined with our above analysis, it is unreasonable for the restored image to consider

only the distortion index. Especially in recent years, Blau et al.[38] have found that distortion and perceived quality are contradictory, so it is more reasonable to consider the distortion index and perceived quality assessment in a comprehensive way. In this work, we have introduced perceptual loss to make the generated image results more in line with human perception.

3 Methodology

In this section, we will elaborate the design details of RBDN to achieve the purpose of this paper that would improve the overall perceptual quality of SR when the distortion-based metrics (PSNR, SSIM) of the image are maintained at a reasonable level. In the following, we first explain our improved network architecture; then we explain the design method of the generator (the objective function of its source); and finally list the loss function we used.

3.1 Network Architecture

Fig. 2 The results of our approach are higher clarity and better perception

Our RBDN architecture mainly consists four parts: encoder, the residual bottleneck dense network blocks (RBDN Block), decoder and the clipping layer, as shown in Fig. 2. The encoder contains the up-sampling and one convolutional layer, and the decoder contains one T-convolutional layer and one trainable project layer. Denote $\mathbf{I}_{LR}$ and $\mathbf{I}_{SR}$ as the input and output of RBDN, respectively. The first convolutional layer extracts the features from the LR input as

$$\mathbf{F}_0 = \mathbf{H}_{\text{Encoder}}(\mathbf{I}_{LR}) \quad (1)$$

where $\mathbf{H}_{\text{Encoder}}(\cdot)$ denotes up-sampling and convolution operation, $\mathbf{F}_0$ is the input to the basic RBDN block for further shallow feature extraction and global residual learning.

Suppose there is M basic blocks in the RBDN, then the output of the m -th RRBB ($m \in \mathbb{N}[1, M]$) can be obtained by

$$\mathbf{F}_m = \mathbf{H}_{\text{RRBB},m}(\mathbf{F}_{m-1}) = \mathbf{H}_{\text{RRBB},m}(\mathbf{H}_{\text{RRBB},m-1}(\cdots(\mathbf{H}_{\text{RRBB},1}(\mathbf{F}_0))\cdots)) \quad (2)$$

where $\mathbf{H}_{\text{RRBB},m}$ denotes the operation of the m -th RRBB, which can be a composite function of operations, such as convolution and exponential linear units (ELU). Since $\mathbf{F}_m$ is produced by the m -th RRBB via fully utilizing each convolutional layer within the block, we can view $\mathbf{F}_m$ as local feature. More details about RRBB and RBDN will be given in Section 3.2 and Section 3.3.

Now the output of the RBDN blocks will pass through the decoder, which tries to reconstruct the input data as

$$\mathbf{F}_d = \mathbf{H}_{\text{Decoder}}(\mathbf{F}_m).$$

After extracting hierarchical features with a set of RRBB blocks, we further conduct dense feature fusion (DFF) among global feature fusion and global residual learning. Dense feature fusion makes full use of features from all the preceding layers and can be represented as

$$\mathbf{F}_{\text{DFF}} = \mathbf{H}_{\text{DFF}}(\mathbf{F}_0, \mathbf{F}_1, \cdots, \mathbf{F}_d)$$

where $\mathbf{F}_{\text{DFF}}$ is the output feature-maps of dense feature fusion by utilizing a composite function $\mathbf{H}_{\text{DFF}}$.

Finally, after extracting local and global features in the LR space, we use a clipping layer in the HR space to obtain the output of RBDN as

$$\mathbf{I}_{\text{SR}} = \mathbf{H}_{\text{RBDN}}(\mathbf{I}_{\text{LR}})$$

where $\mathbf{H}_{\text{RBDN}}$ denotes the function of our RBDN.

3.2 Residual-in-Residual Bottleneck Block (RRBB)

We refer to the residual block (RB) in ResNet [6] to propose our Residual-in-Residual Bottleneck Block (RRBB). Compared to the original RB, we have made the following improvements in the RRBB: (1) we delete the BN layer; (2) we employ the dense connection between residual blocks, similar to the DenseNet structure [10, 12]; (3) we introduce the Bottleneck structure to reduce the number of parameters and the amount of calculation brought by the dense connections; (4) we replace the ReLU activation function in the basic block with ELU. These improvements enable our RRBB to be really different with residual block in SRGAN[5] and dense block in ESRGAN[15], as shown in Fig. 3. The following will give reasons for these changes and how they allow better results.

Fig. 3 Basic units of SRGAN, ESRGAN, and RRBB. (a). Residual Block in SRGAN[5] (b). Dense Block in ESRGAN[15] (c). our RRBB

First of all, the presence of the BN layer has beneficial effects on the entire network, such as speeding up training and convergence, avoiding gradient vanishing, and prevent overfitting. But the BN layer also has negative effects that it reduces the absolute difference between data samples but highlights the relative difference. For instance, when the training and test data sets are quite different, the BN layer will introduce artifacts and further limit the generalization ability of the network [15]. That's why the BN structure performs well in image classification and related tasks, but does not play well enough in the SISR task. Hence, deleting BN is helpful to enhance model performance and reduce computational complexity [15].

In the next place, in order to strengthen the feature propagation and feature reuse, we also use basic blocks similar to dense block [10]. [10, 12] have proved that the dense connection between different convolutional layers can improve the performance in feature fusion. While introducing dense connections will increase the calculation cost, we use the Bottleneck structure to reduce the number of parameters and the amount of calculation.

The last thing is that the ReLU function in the general RB is replaced with a better-performing ELU function, which improves the network performance while offsets the negative effects of removing the BN layer. The ELU function can not only alleviate the gradient vanishing problem, but also enables the data mean to be closer to 0 like

where $\ell_{\mathcal{C}(\mathbf{x}, \epsilon)}(\cdot)$ is the indicator function on the convex set $\mathcal{C}$, which can be computed by a trainable projection layer [18], defined as:

$$\ell_{\mathcal{C}(\mathbf{x}, \epsilon)}(\mathbf{x}) = \begin{cases} 0, & \text{if } \|\mathbf{y} - \mathbf{x}\|_2 \leq \epsilon \\ +\infty, & \text{otherwise} \end{cases} \quad (5)$$

where $\epsilon = \exp(\delta)\sigma\sqrt{N_t - 1}$ is the parameterized threshold with the trainable parameter δ and the total number of pixels in the image N_t . By the proximal gradient algorithm [20], the recursive solution of (4) is as follows:

$$\mathbf{x}^t = \mathbf{prox}_{\eta^t \ell_{\mathcal{C}}}(\mathbf{x}^{t-1} - \eta^t \nabla_{\mathbf{x}} \mathbf{F}(\mathbf{x}^{t-1})), \quad (6)$$

where η^t is the iteration step size, $\mathbf{prox}_{\eta^t \ell_{\mathcal{C}}}$ is the proximal operator of the gradient algorithm associated with the indicator function $\ell_{\mathcal{C}}(\mathbf{x}, \epsilon)$, defined as:

$$\mathbf{P}_{\mathcal{C}}(\mathbf{z}) = \arg \min_{\mathbf{x} \in \mathcal{C}} \|\mathbf{x} - \mathbf{z}\|_2^2 + \ell_{\mathcal{C}(\mathbf{y}, \epsilon)}(\mathbf{x}). \quad (7)$$

By simple calculation, we obtain the gradient of $\mathbf{F}(\mathbf{x})$ as $\nabla_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = \mathbf{H}^T (\mathbf{H}\mathbf{x} - \mathbf{y}) + \lambda \sum_{k=1}^K \mathbf{L}_k^T \phi_k(\mathbf{L}_k \mathbf{x})$, in which $\phi_k(\cdot)$ is the gradient of $\rho_k(\cdot)$, where ϕ_k corresponds to the ELU nonlinear activation function in the network we designed. Based on the analysis of Eqs. (3)-(7) above, the final solution of the objective function can be expressed as:

$$\mathbf{x}^t = \mathbf{P}_{\mathcal{C}} \left((1 - \eta^t \mathbf{H}^T \mathbf{H}) \mathbf{x}^{t-1} + \eta^t \mathbf{H}^T \mathbf{y} - \delta \sum_{k=1}^K \mathbf{L}_k^T \phi_k(\mathbf{L}_k \mathbf{x}^{t-1}) \right), \quad (8)$$

where $\delta = \lambda \eta^t$ denotes the trainable projection layer parameter.

To sum up, we design our generator network according to Eq. (8).

3.4 Network Losses

We employ four classes of loss functions to measure reconstruction error and train our generative adversarial network [21]. The overall loss function is formulated as:

$$\mathcal{L}_{\text{RBDN}} = \mathcal{L}_{\text{per}} + \alpha \mathcal{L}_{\text{GAN}}^{\text{Ra}} + \beta \mathcal{L}_{\text{tv}} + \gamma \mathcal{L}_1 \quad (9)$$

where α , β and γ in Eq. (9) are coefficients used to balance different losses.

- **Perceptual loss** ($\mathcal{L}_{\text{per}}$): It measures the semantic differences between images and help the results produced by our trained network have better perceptual effects, defined as:

$$\mathcal{L}_{\text{per}} = \frac{1}{N} \sum_{i=1}^N \|\phi(\mathbf{G}_{\text{SR}}(\hat{\mathbf{x}}_i)) - \phi(\mathbf{y}_i)\|_1 \quad (10)$$

where ϕ is the feature extracted from the pretrained network.

- **Texture loss** ($\mathcal{L}_{\text{GAN}}^{\text{Ra}}$): This term focuses on the high frequencies of the output image. According to the framework of the generator based on the relativistic generative adversarial network [15], we define the relativistic average generator loss:

$$\mathcal{L}_{\text{GAN}}^{\text{Ra}} = -\mathbb{E}_{\mathbf{y}}[\log(1 - \mathbf{D}_{\text{Ra}}(\mathbf{y}, \mathbf{G}_{\text{SR}}(\hat{\mathbf{x}})))] - \mathbb{E}_{\hat{\mathbf{y}}}[\log(\mathbf{D}_{\text{Ra}}(\mathbf{G}_{\text{SR}}(\hat{\mathbf{x}}), \mathbf{y}))] \quad (11)$$

where $\mathbb{E}_{\mathbf{y}}$ and $\mathbb{E}_{\hat{\mathbf{y}}}$ represent the operation of taking average for all real data $\mathbf{y}$ and fake data $\hat{\mathbf{y}}$ in the mini-batch, respectively.

- **Total-variation loss** ($\mathcal{L}_{\text{tv}}$): This term mainly focuses on suppressing noise in generated images [19]. In this paper, it is defined as the sum of the absolute differences of the gradient discrepancy so as to produce sharpness in the output image, as follows:

$$\mathcal{L}_{\text{TV}} = \frac{1}{N} \sum_{i=1}^N (\|\nabla_h \mathbf{G}_{\text{SR}}(\hat{\mathbf{x}}_i) - \nabla_h(\mathbf{y}_i)\|_1 + \|\nabla_v \mathbf{G}_{\text{SR}}(\hat{\mathbf{x}}_i) - \nabla_v(\mathbf{y}_i)\|_1) \quad (12)$$

where ∇_h and ∇_v respectively represent the horizontal and vertical gradient of the image.

- **Content loss** ($\mathcal{L}_1$): This item is suitable for evaluating the 1-norm distance content loss between the generated result and the real image.

$$\mathcal{L}_1 = \frac{1}{N} \sum_{i=1}^N \|\mathbf{G}_{\text{SR}}(\hat{\mathbf{x}}_i) - \mathbf{y}_i\|_1. \quad (13)$$

4 Results and Discussions

4.1 Datasets and Evaluation Metrics

We mainly used DIV2K [29] as training datasets, which is a dataset that contains 800 images with 2K resolution, mainly used for super-resolution reconstruction. The Flickr2K [32] and OutdoorSceneTraining (OST) [17] datasets were used to enrich the training set with more diverse textures. Previous research has found that the richer texture information help the generator produce more natural results. Our final training set consisted of 13,297 images, and the LR images we need are obtained by subsampling the images from the above data set. In the testing process, we used three benchmark test sets: Set5 [35], Set14 [36], and BSD100 [37].

In this article, we evaluate our training model using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [30], as well as training time. PSNR and SSIM are distortion-based measures, while LPIPS is a perceptual metric based on human similarity judgments and thus is a subjective measurement. However, it is difficult to declare one metric to be an absolute winner compared to the rest as each metric is tendentiously and partially evaluates the model in one dimension. A fair comparison is only possible by synthetically considering among the subjective and objective metrics. In addition, we also have considered the training time (TIME), it is a simple and efficient metric to evaluate all model with consistent parameters.

4.2 Training Details

The training process is divided into two phases. First, we train a model for PSNR through ℓ_1 loss. The learning rate is initialized as 2×10^{-4} , and decayed by 1/2 for every 2×10^5 small-volume updates. Then we use the trained PSNR-oriented model as our initial generator via the loss function (9). The superparameter α and γ in Eq. (9) are 5×10^{-3} and 1×10^{-2} , respectively. Here we set the initial learning rate to 1×10^{-4} , and it is halved at the iteration of [125k, 250k, 500k, 750k] where we did a total of 100,000 iterations.

Throughout the training, we used the PGM method [20], and we updated the generators and discriminator in the network alternately until the model reached convergence. We ended up with the generator internal structure: we used a hierarchy and number of layers similar to ESRGAN [15] – a deeper model of 23 RRBB blocks, each containing 27 levels of functional layers. The above work is implemented using the PyTorch [34] framework, and trained by the NVIDIA RTX8000.

In particular, the introduction of perceptual loss in Eq. (9) we used is also one of the details of our training: Pre-training with perceptual loss helps GAN-base methods to obtain more visually pleasing results. The reasons are that: 1) it can avoid undesired local optima for the generator; 2) after pre-training, the discriminator receives relatively good super-resolved images instead of extreme fake ones (black or noisy images) at the very beginning, which helps it to focus more on texture discrimination [15].

4.3 Results and Analysis

Quantitative Comparison: We compare our methods with state-of-the-art perceptual-driven SRs for quantitative analysis methods including EDSR [12], SRResNet [6], and also with perceptual-drivers including SRGAN [5] and ESRGAN [15]. Table 1 gives their LPIPS, PSNR and SSIM values on three test datasets. As shown in the table, EDSR and SRResNet have the highest PSNR and SSIM scores in all datasets, while SRGAN, ESRGAN and RBDN are not prominent in this type of distortion-based metrics. When we looked closely, we found that SRGAN, ESRGAN and RBDN alternately win the highest PSNR and SSIM scores among these three, thus it is still a close score on these two metrics.

Now we turn to the LPIPS metric, RBDN exceeds that of EDSR and SRResNet in the metric of LPIPS. In addition, EDSR and SRResNet cannot achieve the second best LPIPS value under any circumstances, so EDSR and SRResNet are more like PSNR-oriented SR methods, The PSNR-oriented method can obtain higher PSNR values than other methods, but it is easy to produce fuzzy results. In addition, our performance is equally superior to that of SRGAN and ESRGAN, where we achieve the highest LPIPS values for each set of test sets while achieving higher levels of PSNR and SSIM. Therefore, the results show that the method has a significant advantage, and our RBDN method achieves excellent sensory quality while slightly distorting.

Table 1 Quantitative comparison results of our model with other models on three test datasets. The best score in each line is highlighted in red color and the sub-optimal is highlighted in blue color

Dataset	Metic	Bicubic	SRGAN	SRResNet	EDSR	ESRGAN	RBDN (ours)
Set5	PSNR $\uparrow$	-	28.0289	29.2508	29.1822	28.9790	27.8422
	SSIM $\uparrow$	-	0.8078	0.8642	0.8632	0.7016	0.7882
	LPIPS $\downarrow$	-	0.0882	0.1354	0.1387	0.0788	0.0748
Set14	PSNR $\uparrow$	-	24.7499	26.2395	26.7959	20.2951	24.3409
	SSIM $\uparrow$	-	0.6127	0.7122	0.7448	0.5796	0.6618
	LPIPS $\downarrow$	-	0.1663	0.2309	0.2349	0.1329	0.1319
BSD100	PSNR $\uparrow$	-	24.1956	26.2395	26.2265	23.8162	25.3409
	SSIM $\uparrow$	-	0.6245	0.7122	0.7448	0.6592	0.6127
	LPIPS $\downarrow$	-	0.1980	0.3094	0.3146	0.1614	0.1612

Qualitative Comparison: We also make visual comparisons to perception-driven SR methods. From the comparison, we see that our results are more natural and realistic than other methods. For Fig. 8, the sharp edges indicate that our method is feasible to capture the structural characteristics of objects in the image. In comparison with other images, our method also restored better textures, such as the birds in Fig. 6 with a well-layered feather texture, for Fig. 7 where we restored the straw more clearly, and Fig. 9, where the plants we recovered were most in line with the real-world plant effect. Comparing it with other SR method, the structure in our results is clear and there is no serious distortion, while other methods fail to give the object a satisfactory appearance. Qualitative comparison verifies that our proposed RBDN method can learn more about structural information in gradient space, which helps to generate realistic SRs by saving images of geometry. At the same time, our model has a more impressive convergence speed. ESRGAN took 6 minutes and 23 seconds for the first 500 iterations during the pre-training process, while our model took only 5 minutes and 19 seconds in the same environment. While getting good visuals, faster convergence speeds are achieved, which means that when we train deeper network models, the training time is greatly reduced.

5 Conclusion

In this article, we proposed an improved RBDN model for the SISR problem in response to the slow convergence speed of the model of SRGAN type and the artifacts in the restored image, which are still different from human perception: First, while introducing the dense connection, we use the bottleneck structure; Secondly, we no longer use the batch normalization layer that will cause artifacts, use residual scaling, and replace the ReLU function with the ELU function. We apply our designed RRBB basic block to the residual network of [19], so that the results obtained conform to both the real world and human perception. The quantitative and qualitative experimental results on three commonly used benchmark test sets show that the effectiveness of our method. The future work

Fig. 6 Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The ‘PSNR/LPIPS’ metrics: (b). -/-; (c). 29.4174/0.2127; (d). 29.1325/0.2159; (e). 26.5269/0.2043; (f). 26.0123/0.1657; (g). 26.0674/0.1239

Fig. 7 Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The ‘PSNR/LPIPS’ metrics: (b). -/-; (c). 28.5119/0.2537; (d). 28.5198/0.2534; (e). 26.4736/0.1549; (f). 26.6683/0.1397; (g). 26.0073/0.1356

are exploring the optimization of low-definition images. The method of image texture and detail, and hope to further increase the training speed of the image to improve the generalization of the model.

Abbreviation

The table below is a description of the abbreviations used in this article

Acknowledgements This work was supported in part by the Natural Science Foundation of Chongqing cstc2020jcyj-msxmX0057, and in part by Fundamental Research Funds for the Central Universities under Grant XDJK2018B013. The statements made herein are solely the responsibility of the authors.

Fig. 8 Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The ‘PSNR/LPIPS’ metrics: (b). -/-; (c). 25.2165/0.2994; (d). 25.2780/0.2908; (e). 23.6720/0.2078; (f). 24.1437/0.1861; (g). 22.7985/0.1635

Fig. 9 Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The ‘PSNR/LPIPS’ metrics: (b). -/-; (c). 27.2098/0.2436; (d). 27.2285/0.2427; (e). 24.0769/0.2403; (f). 25.0842/0.2396; (g). 30.0662/0.2377

Authors’ contributions

Zeyu An: Data curation, Validation, Writing-Original draft preparation, Conceptualization, Methodology Junyuan Zhang: Writing-Original draft preparation, Conceptualization, Methodology, Visualization Ziyu Sheng: Writing-Reviewing and Editing, Supervision, Investigation Xuanhe Er: Data curation, Visualization Junjie Lv: Data curation, Writing- Reviewing and Editing, Supervision, Investigation

Authors’ information

Zeyu An is currently pursuing an undergraduate degree at Southwest University, Chongqing, China. His major is information security. His research interests include cyberspace security, residual networks, super-resolution reconstruction, and visual SLAM. Junyuan Zhang is currently pursuing an undergraduate degree at Southwest University, Chongqing, China. His major is information security. His research interests include cyberspace security,

Abbreviations	Concrete meaning
SRGAN	Super-Resolution Generative Adversarial Network
RRBB	Residual-in-Residual Bottleneck Block
RBDN	Residual-in-Residual Bottleneck Dense Network
BN	batch normalization
SR	Super-resolution
HR	Higher-resolution
LR	lower-resolution
SISR	single-image super-resolution
SRCNN	Super-Resolution Convolution Neural Network
VDSR	Very Deep Convolution Network
DRCN	Deeply-Recursive Convolution Network for image Super-Resolution
CNN	Convolution Neural Network
PSNR	peak signal-to-noise ratio
SSIM	Structural SIMilarity
ESRGAN	Enhanced Super-Resolution Generative Adversarial Networks
RGAN	Recurrent Generative Adversarial Networks
ResNet	Residual Network
SRResNet	Super-Resolution Residual Network
ELU	Exponential Linear Units
DFE	Dense Feature Fusion
OST	OutdoorSceneTraining
LPIPS	Learned Perceptual Image Patch Similarity
TIME	training time

residual networks, super-resolution reconstruction, and machine olfaction. Ziyu Sheng is currently a postgraduate student at Southwest University. His research interests include applications of neural networks in the field of load forecasting. Xuanhe Er currently studied in Chengdu University of information engineering, Sichuan, China. His major is data science and big data technology. His current research interests include deep learning, data science, and image processing. Junjie Lv is currently pursuing the M.Sc. degree at Southwest University. His research interests are deep learning and its applications in computer vision.

Funding

No funding was received.

Availability of data and materials

The image datasets used to support the findings of this study can be downloaded from the public websites.

Competing interests

The authors declare that there are no conflict of interests, we do not have any possible conflicts of interest.

References

1. Dong C, Loy CC, He K, et al (2014) Learning a deep convolutional network for image super-resolution. European conference on computer vision. Springer, Cham, pp 184–199
2. Dong C, Loy CC, He K, et al (2015) Image super-resolution using deep convolutional networks. IEEE Transaction on Pattern Analysis and Machine Intelligence 38(2): 295–307

3. Kim J, Lee JK, Lee KM (2016) Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 1637–1645
4. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 1646–1654
5. Ledig C, Theis L, Huszar F, et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 4681–4690
6. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 770–778
7. Lai WS, Huang JB, Ahuja N, et al (2017) Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 624–632
8. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3147–3155
9. Tai Y, Yang J, Liu X, et al (2017) Memnet: A persistent memory network for image restoration. In: Proceedings of the IEEE international conference on computer vision. IEEE, pp 4539–4547
10. Zhang Y, Tian Y, Kong Y, et al (2018) Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 2472–2481
11. Gu J, Wang Z, Kuen J, et al (2018) Recent advances in convolutional neural networks. Pattern Recognition 77: 354–377
12. Lim B, Son S, Kim H, et al (2017) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. IEEE, pp 136–144
13. Gatys LA, Ecker AS, Bethge M (2015) Texture synthesis using convolutional neural networks. arXiv preprint arXiv: 1505.07376
14. Bruna J, Sprechmann P, LeCun Y (2015) Super-resolution with deep convolutional sufficient statistics. arXiv preprint arXiv: 1511.05666
15. Wang X, Yu K, Wu S, et al (2018) ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: Proceedings of the European conference on computer vision (ECCV) workshops. ECCV, pp 0–0
16. Nah S, Kim TH, Lee KM (2017) Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3883–3891
17. Chen Y, Pock T (2017) Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6): 1256–1272
18. Lefkimmiatis S (2018) Universal denoising networks: A novel can architecture for image denoising. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3204–3213
19. Umer RM, Foresti GL, Micheloni C (2020) Deep Generative Adversarial Residual Convolutional Networks for Real-World Super-Resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. IEEE, pp 438–439
20. Parikh N, Boyd S, et al (2014) Proxiaml algorithms. Foundations and Trends in Optimization 1(3): 127–239
21. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets. In: Proceedings of Advances in neural information processing systems. New York: ACM, 2014, 2672–2680
22. Lai WS, Huang JB, Ahuja N, et al (2017) Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 624–632
23. Zhang Y, Li K, Li K, et al (2018) Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV), pp 286–301
24. Dai T, Cai J, Zhang Y, et al (2019) Second-order attention network for single image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. IEEE, pp 11065–11074
25. Li Q, Li Z, Lu L, et al (2019) Feedback network for image superresolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3867–3876
26. He J, Dong C, Qiao Y (2019) Modulating image restoration with continual levels via adaptive feature modification layers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. IEEE, pp 11056–11064
27. Hu X, Mu H, Zhang X, et al (2020) Meta-sr: A magnification-arbitrary network for super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. IEEE, pp 1575–1584

28. Zhang Y, Li K, Li K, et al (2019) Residual non-local attention networks for image restoration. arXiv preprint arXiv: 1903.10082
29. Agustsson E, Timofte R (2017) Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. IEEE, pp 126–135
30. Zhang R, Isola P, Efros AA, et al (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 586–595
31. Szegedy C, Ioffe S, Vanhoucke V (2016) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI conference on artificial intelligence. IEEE, 31(1)
32. Timofte R, Agustsson E, Van Gool L, et al (2017) Ntire 2017 challenge on single image super-resolution: Methods and results. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. IEEE, pp 114–125
33. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556
34. Paszke A, Gross S, Chintala S, et al (2017) Automatic differentiation in pytorch.
35. Bevilacqua M, Roumy A, Guillemot C, et al (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding.
36. Zeyde R, Elad M, Protter M (2012) On single image scale-up using sparse-representations. International conference on curves and surfaces. Springer, Berlin, Heidelberg, pp 711–730
37. 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. In: Proceedings Eighth IEEE international conference on computer vision (ICCV). IEEE, 2: 416–425
38. Blau Y, Michaeli T (2017) The perception-distortion tradeoff. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 6228–6237
39. Zagoruyko S, Komodakis N (2016) Wide residual networks. arXiv preprint arXiv: 1605.07146
40. Zhang C, Rameau F, Lee S, et al (2019) Revisiting residual networks with nonlinear shortcuts. In: The British Machine Vision Conference. BMCV, pp 12
41. Philipp G, Song D, Carbonell JG (2018) Gradients explode-deep networks are shallow-resnet explained.
42. Huang G, Liu Z, Laurens VDM, et al (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 4700–4708
43. Pleiss G, Chen D, Huang G, et al (2017) Memoryefficient implementation of densenets. arXiv preprint arXiv: 1707.06990

Figures

Figure 1

Visual comparison of our model with other state-of-art models on Set5/Set14 at the $\times 4$ super-resolution

Figure 2

The results of our approach are higher clarity and better perception

(a) Residual Block

(b) Dense Block

(c) RRBB

Figure 3

Basic units of SRGAN, ESRGAN, and RRDB. (a). Residual Block in SRGAN[5] (b). Dense Block in ESRGAN[15] (c). our RRBB

Figure 4

The internal structure of RRBB's basic block

Figure 5

RRBB

Figure 6

Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The 'PSNR/LPIPS' metrics: (b). -/-; (c). 29.4174/0.2127; (d). 29.1325/0.2159; (e). 26.5269/0.2043; (f). 26.0123/0.1657; (g). 26.0674/0.1239

Figure 7

Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The 'PSNR/LPIPS' metrics: (b). -/-; (c). 28.5119/0.2537; (d). 28.5198/0.2534; (e). 26.4736/0.1549; (f). 26.6683/0.1397; (g). 26.0073/0.1356

Figure 8

Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The 'PSNR/LPIPS' metrics: (b). -/-; (c). 25.2165/0.2994; (d). 25.2780/0.2908; (e). 23.6720/0.2078; (f). 24.1437/0.1861; (g). 22.7985/0.1635

Figure 9

Visual comparison of our model with other state-of-art models on BSD100 test dataset at the $\times 4$ super-resolution. The 'PSNR/LPIPS' metrics: (b). -/-; (c). 27.2098/0.2436; (d). 27.2285/0.2427; (e). 24.0769/0.2403; (f). 25.0842/0.2396; (g). 30.0662/0.2377

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Highlights.doc](#)