Multilevel Residual Learning for Single Image Super Resolution

Xiaole Zhao[1](✉), Hangfei Liu[1], Tao Zhang[1,2], Wei Bian[1,2], and Xueming Zou[1,2]

[1] School of Life Science and Technology, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China
zxlation@foxmail.com
[2] Alltech medical system co. LTD, Chengdu, Sichuan, China
{tao.zhang,mark.zou}@alltechmed.com

Abstract. Single image super-resolution (SISR) methods based on deep learning techniques, especially convolutional neural networks (CNNs) and residual learning, have made great achievements compared with traditional methods. Most of the current work focuses on the structural design to increase the depth of the entire network and thus improve the performance of the models. However, it is also important to improve the efficiency of model parameters, especially in the case of limited resources. To improve the performance of the models when the number of model parameters keeps relatively small and fixed, we propose a novel multilevel residual learning pattern for SISR in this work. The proposed method shows a stable performance improvement over the compared structures on several benchmark datasets with equal model parameters. Besides, we empirically show that simply increasing the number of building blocks (e.g. various residual blocks) to increase the depth of the networks will not obtain the expected improvements of performance, which may imply that the optimal performance of different network depths corresponds to different structures of building blocks.

Keywords: Convolutional Neural Networks · Residual learning
Image Super Resolution · Skip connection

1 Introduction

Single image super resolution (SISR) is a classic ill-posed problem in computer vision community which aims at recovering a high resolution (HR) image from only one low resolution (LR) image. High resolution means that pixel density within an image is higher than its LR counterparts and therefore an HR image can offer more details that may be critical in various applications such as medical imaging [1,2], aerial spectral imaging [3] and remote sensing imaging [4,5], face recognition [6], security and surveillance [7] et al., where high-frequency details are usually critical and greatly desired.

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11256, pp. 537–549, 2018.
https://doi.org/10.1007/978-3-030-03398-9_46

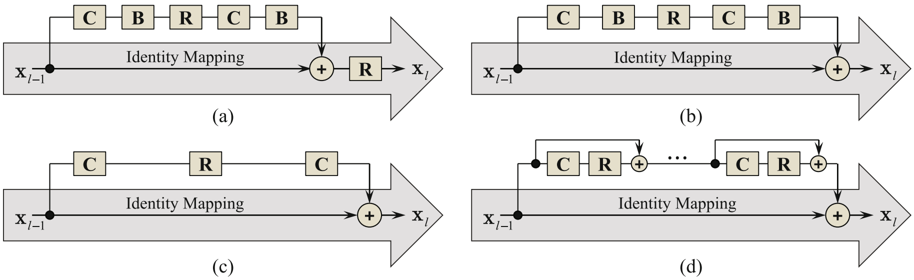


Fig. 1. The structure of several residual blocks. **C**, **B**, **R** and + represent conv, batch normalization, ReLU and element-wise addition respectively. (a) the original residual block [18]. (b) SRResNet [14]. (c) EDSR/MDSR [16]. (d) the proposed residual block.

In recent years, many image super resolution (SR) methods based on deep learning techniques [8], especially convolutional neural networks (CNNs) and residual learning, have emerged and greatly promoted the best state of SR. Some of the most representative are SRCNN [9], DRCN [10], DRRN [11], VDSR [12], ESPCNN [13], SRResNet [14], LapSRN [15], EDSR/MDSR [16] and RDN [17] etc. Residual learning [18,19] is a trick to increase the depth of the networks and thus improve the model performance. It was first proposed for image recognition and has been widely proved to be helpful for gradients propagation and model convergence, thus making it possible to build extremely deep networks. With the increased depth of networks, the expressive power and generalization ability of the models have also been improved. Though many methods based on residual learning (e.g. SRResNet [14], EDSR/MDSR [16] and RDN [17]) have achieved much better results than previous methods, the cost for getting further improvement of model performance becomes more and more expensive as the depth of the network increases. Therefore, it is useful to improve the efficiency of model parameters in the case of limited resources.

A key factor of residual learning that affects model training and performance is *residual connection* (or *skip connection*, *shortcut* [18]). The previous methods have a common feature in network structure design: residual learning is usually applied to the overall structure of the networks or building blocks, but not deep into the information paths of a residual block. Normally, a residual block is composed of a *residual path* (a identity mapping) and a *main path* (Fig. 1). In this work, we present a novel multilevel residual learning pattern for SISR, which we term ML-ResNet. In our model, residual connection is applied not only to the outermost layers and the internal residual blocks, but also to the main path within a residual block (Fig. 1(d)). Thus, the whole structure of the network exhibits the characteristic of multilevel residual learning.

We evaluate the proposed model on several benchmark datasets and compare it with some common block structures. The experimental results show that the multilevel residual structure has a stable performance improvement over the compared methods with equal model parameters. Moreover, we also empirically illustrate that simply increasing the number of building blocks does not achieve the expected performance gain, which implies that the optimal performance of the networks with different depth may correspond to different structures of building blocks. This observation might shed some light on the structural design of deep networks or building blocks.

2 Related Work

2.1 Super Resolution with Deep Learning

Dong et al. proposed the first SR model [9] based on CNNs in the modern sense and built an end-to-end mapping between the (bicubic) interpolated LR images and their HR counterparts. The further improvement based on this pioneering work mainly aimed at increasing network depth or sharing network weights at the beginning [10–12]. These methods use the interpolated version of the LR image as the input of their model, which is convenient for keeping the size of the output image consistent with the target HR image and works well for the fractional scaling factors. However, it hinders establishing end-to-end mappings from the original LR image to the corresponding HR image and suffers the computational and memory constraints as they operate feature maps in the HR image space. This problem can be solved by placing nonlinear mapping in the LR image space. There are two options for the purpose currently, i.e., transpose convolution (or deconvolution [20]) and efficient sub-pixel convolutional neural network (ESPCNN) [13]. As the amount of computation and memory occupancy are greatly reduced, Lim et al. [16] increased the depth and the width (the number of the feature maps' channel) of their networks aggressively (32 residual blocks for EDSR and 80 residual blocks for MDSR).

Although these networks have made great breakthroughs in improving SR results, their performance gains are mainly achieved by increasing network depth and adjusting the structure of the entire network. Changes in the structure of residual blocks also aim at increasing the network depth to a certain extent. On the contrary, the target of this work is to promote the information flow through the entire network and improve the efficiency of the model parameters.

2.2 Residual Learning for Super Resolution

Residual Network (ResNet) [18] is initially proposed for image recognition, which is further applied to a wide range of computer vision problems such as image classification, object detection, image segmentation and image generation. Most of the methods mentioned in Sect. 2.1 apply residual learning, e.g., DRRN [11], VDSR [12], SRResNet [14], EDSR [16] and RDN [17] etc. An impressive work was presented in [21], named HelloSR. Inspired by the effectiveness of learning high frequency residuals for SR, HelloSR presented a novel stacked residual refined network which generated HR image by explicitly learning the multilevel residuals in the HR image space.

These methods employ residual learning in different ways. However, most of them adopt residual connections only between the outermost layers or the middle modules of their network, but not within the information paths of a building block. In this work, the outermost layers, the intermediate building blocks and the information pathes within a block are viewed as different levels of a network and residual learning is applied to all of these levels. Experiments show that this multilevel residual structure is helpful to improve the performance of the model when the network structure is relatively shallow.

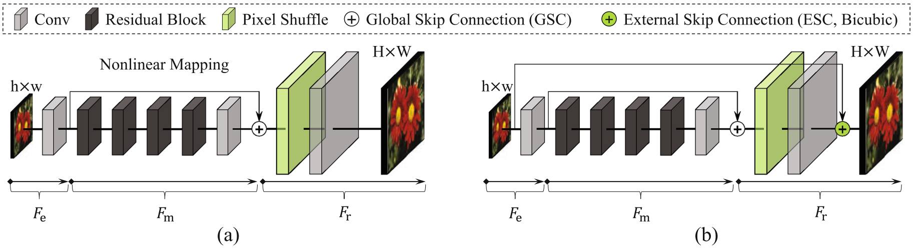


Fig. 2. The overall structure of two networks used in this work. (a) the same structure as EDSR [16] but the number of residual blocks is limited to 4. (b) Extension of (a) with an external skip connection.

3 Multilevel Residual Networks

3.1 Overall Network Structure

The overall structure of ML-ResNet is outlined in Fig. 2. The networks consist of three typical parts: feature extraction network (FEN), nonlinear mapping network (NMN) and HR image reconstruction network (HRN). The FEN is applied to represent the input image as shallow features. These shallow features are then fed into a set of cascaded building blocks, i.e., NMN that produces deep features. Next, a pixel shuffle layer is concatenated to upsample deep features to match the expected size (e.g. SR×2 or SR×4). Finally, the upsampled features are delivered to HRN to generate the HR outputs.

Denote $\mathbf{x}$ and $\mathbf{y}$ as the input and the output of the entire network, $\mathbf{x}_i$ and $\mathbf{y}_i$ as the input and output of the sub networks or building blocks. Formally, the operation for shallow features extraction could be expressed as:

$$\mathbf{y}_0 = F_{\mathrm{e}}(\mathbf{x}) \tag{1}$$

where $F_{\mathrm{e}}(\cdot)$ denotes the first feature extraction network FEN. It extracts the shallow features and expands the dimension along with channel direction. The output of FEN is directly fed into NMN ($\mathbf{x}_0 = \mathbf{y}_0$). Similarly, the operation for the whole nonlinear feature mapping network could be denoted as:

$$\mathbf{y}_n = F_{\mathrm{m}}(\mathbf{x}_0) \tag{2}$$

where n denotes the number of the building blocks, and $\mathbf{y}_n$ indicates the output of the nonlinear feature mapping function $F_{\mathrm{m}}(\cdot)$. Here, $F_{\mathrm{m}}(\cdot)$ includes all the building blocks within nonlinear feature mapping network and the subsequent conv layer, as shown in Fig. 2.

After the global skip connection (GSC), the input of HR image reconstruction network is $\mathbf{x}_{n+1} = \mathbf{y}_n + \mathbf{x}_0$. In EDSR/MDSR [16], the final output of the entire network is as follow (Fig. 2(a)):

$$\mathbf{y} = F_{\mathrm{r}}(\mathbf{x}_{n+1}) = F_{\mathrm{r}}(\mathbf{y}_n + \mathbf{x}_0) \tag{3}$$

where $F_{\mathrm{r}}(\cdot)$ denote HR reconstruction function that consists of a pixel shuffle followed by a conv layer. However, there is an external skip connection (ESC) before the final output of the proposed ML-ResNet, as shown in Fig. 2(b):

$$\mathbf{y} = \mathbf{x} + F_{\mathrm{r}}(\mathbf{x}_{n+1}) = \mathbf{x} + F_{\mathrm{r}}(\mathbf{y}_n + \mathbf{x}_0) \tag{4}$$

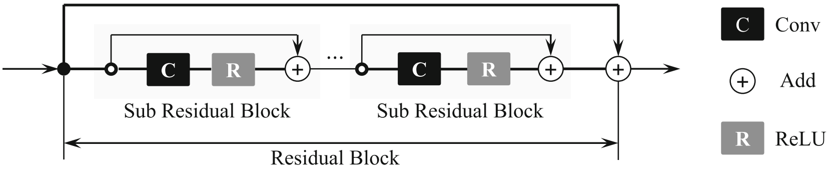


Fig. 3. Detailed illustration of the proposed residual block. Each residual block consists of many sub residual blocks, which are composed of the basic Conv + ReLU operations.

3.2 Building Residual Blocks

ResNet is usually modularized and consists of a series of stacked blocks. In a residual block, the main path augments the expressive ability of the model, while the residual path promotes the information propagation through the entire network. Denote the input and the output of a residual block $\mathcal{B}_l$ as $\mathbf{x}_l$ and $\mathbf{y}_l$ respectively. Then $\mathcal{B}_l$ can be expressed in a general form [18]:

$$\begin{aligned} \mathbf{y}_l &= h(\mathbf{x}_l) + \mathcal{F}_{\mathrm{B}}(\mathbf{x}_l, \mathcal{W}_l) \\ \mathbf{x}_{l+1} &= f(\mathbf{y}_l) \end{aligned} \tag{5}$$

where $h(\cdot)$ and $\mathcal{F}_{\mathrm{B}}(\cdot)$ are the mapping function of residual path and the main path respectively. $f(\cdot)$ is a function that converts the output of $\mathcal{B}_l$ to the input of $\mathcal{B}_{l+1}$. He et al. [19] theoretically explained that a compact information path (the identity mapping in Fig. 1) is helpful for easing optimization, i.e., $h(\mathbf{x}_l) = \mathbf{x}_l$ and $f(\mathbf{y}_l) = \mathbf{y}_l$. This is viewed as a contiguous memory mechanism [17] and most of the current SR models follow this principle.

However, most of the previous methods adopted direct nonlinear mapping in the main path $\mathcal{F}_{\mathrm{B}}(\cdot)$. In this work, residual learning is also applied deep into the main path $\mathcal{F}_{\mathrm{B}}(\cdot)$ of a residual block, as shown in Fig. 3 and Fig. 1(d). We call this ResNet-in-ResNet structure *fine-grained residual learning*, which is expected to promote data flow in the main path of a residual block. One can adjust the number of sub residual blocks (SRB) in a residual block (RB) and thus change the density of residual learning. If the NMN includes x residual blocks and each residual block contains y sub residual blocks, we call it ML-ResNet (BxSy).

3.3 Multilevel Residual Pattern

In addition to the fine-grained residual learning within residual blocks, we also introduce an external skip connection (ESC) between the outermost layers of the entire network, which we call *coarse-grained residual learning*. Thus, the residual pattern is applied to multiple abstract levels of the model and the whole network structure displays the characteristic of multilevel residual learning from fine to coarse grain. This multilevel residual structure is proved to be effective in our experiments, which is probably because it is related to the (multilevel) manifold simplification [22] although there is still no strict theoretical argument.

Interestingly, the experiments show that the external skip connection seems to have no obvious effect on the performance of the network with EDSR residual blocks (Fig. 1(c)), but it can slightly improve the performance of the model built with the proposed residual blocks (Fig. 1(d)). This also shows the validity of the multilevel residual structure to some extent.

4 Experiments

In this section, we first introduce some experiment settings. Next, we study the impact of residual density and the external skip connection on the performance of the model. The overall structure of the network is Fig. 2(b) and the reference structure is Fig. 2(a). The residual blocks shown in Fig. 1(b)−(d) are used for comparison. Finally, we compare the proposed model with several previous methods quantitatively and qualitatively. The performance is evaluated with PSNR and SSIM [23]. They are calculated with the built-in functions of Python skimage module during quick validation, but in the testing phase, we use different calculations for fair comparison.

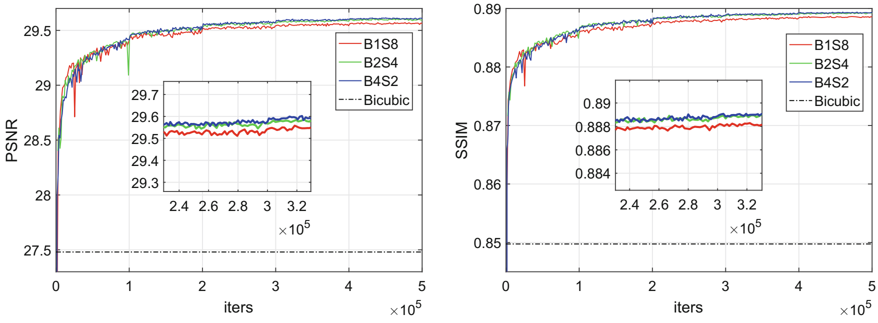
4.1 Training Settings

DIV2K dataset [21,24] is used to train and quickly validate the models (only the first 10 validation images of DIV2K are used). Several standard benchmark datasets are used for testing, including Set5 [25], Set14 [26], B100 [27], Urban100 [28] and DIV2K validation set. For training, the HR images are randomly split into 96 × 96 RGB image patches and the size of LR patches are dynamically adjusted according to SR scales. Data augmentation and mean removal are the same as EDSR/MDSR [16].

Given a training dataset $\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^{|\mathcal{D}|}$, where $|\mathcal{D}|$ is the number of training samples, l_1 loss function is used for model training:

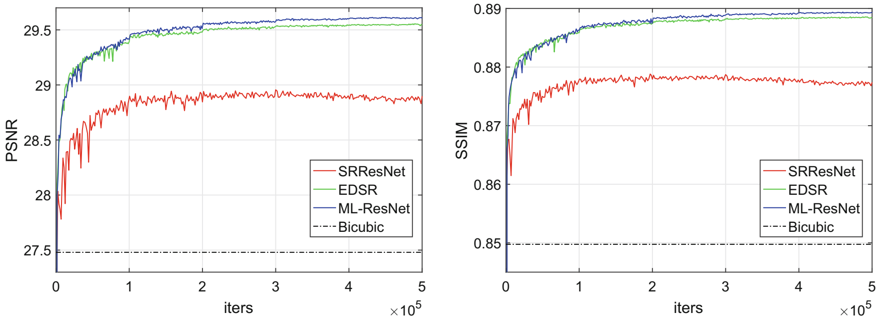
$$L(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} ||\mathbf{y}^i - \hat{\mathbf{y}}^i||_1 \tag{6}$$

where $\hat{\mathbf{y}}$ is the estimate of the model and $\mathbf{y}$ is the corresponding target. $\boldsymbol{\theta}$ denotes the set of model parameters. It is worth noting that the number of parameters is the same for the compared architectures.



(a) PSNR (b) SSIM

Fig. 4. The validation performance of the models with different residual density on the first 10 validation images of DIV2K (SR×4).



(a) PSNR (b) SSIM

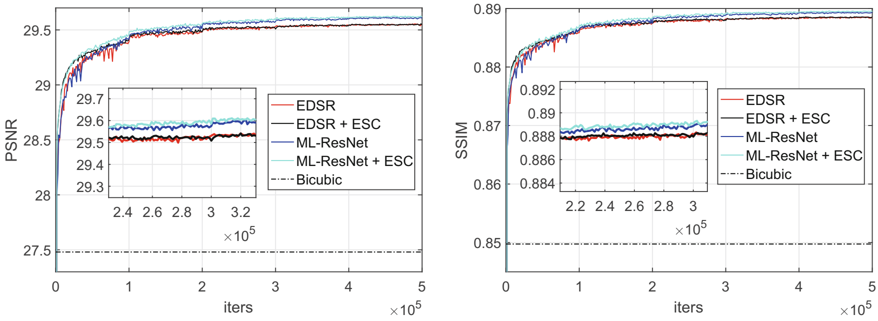
Fig. 5. The validation performance of the models with different residual blocks shown in Fig. 1(b)–(d). Only the first 10 validation images of DIV2K are used for comparison (SR×4).

The size of minibatch is 32 and that of filters is 5×5. The number of residual blocks and feature maps is 4 and 256 respectively. We trained the models with ADAM optimizer [30] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The piecewise constant decay is used for learning rate, i.e., it is initialized as 10^{-4} and halved at every 10^5 iterations. All models are trained for 5×10^5 iterations.

4.2 Residual Density

In our settings, there are multiple combinations of residual blocks and their sub residual blocks when the total number of conv layers is fixed, which forms different residual density of the NMN. For comparison, we used the structure in Fig. 2(a) and set the total number of conv layers in NMN to 8. Thus, we have 4 combinations: B1S8, B2S4, B4S2 and B8S1, where B and S represent the number of residual blocks and sub residual blocks respectively. However, B8S1 is invalid due to the degradation of model structure, as shown in Fig. 3.

From Fig. 4, it can be seen that B2S4 and B4S2 perform almost the same, but obviously better than B1S8. The result is stable in our repeated experiments. It is probably because that a residual network can be viewed as a collection of many paths of differing length [29] and different residual densities lead the actual depth of the entire network to be different. This implies that the optimal performance of different network depths may correspond to different structures of building blocks, and simply increasing the number of building blocks to increase the depth of the network may not achieve expected performance improvements.



(a) PSNR (b) SSIM

Fig. 6. The validation performance of the models with and without ESC. Only the first 10 validation images of DIV2K dataset are used for comparison (SR×4).

4.3 Different Residual Blocks

Because Fig. 1(a) is mainly used for classification, detection and other high-level computer vision problems, we exclude this structure in our experiments. For all of the compared structures, we set 4 residual blocks in the entire network with two convolutional layers in each block for fair comparison.

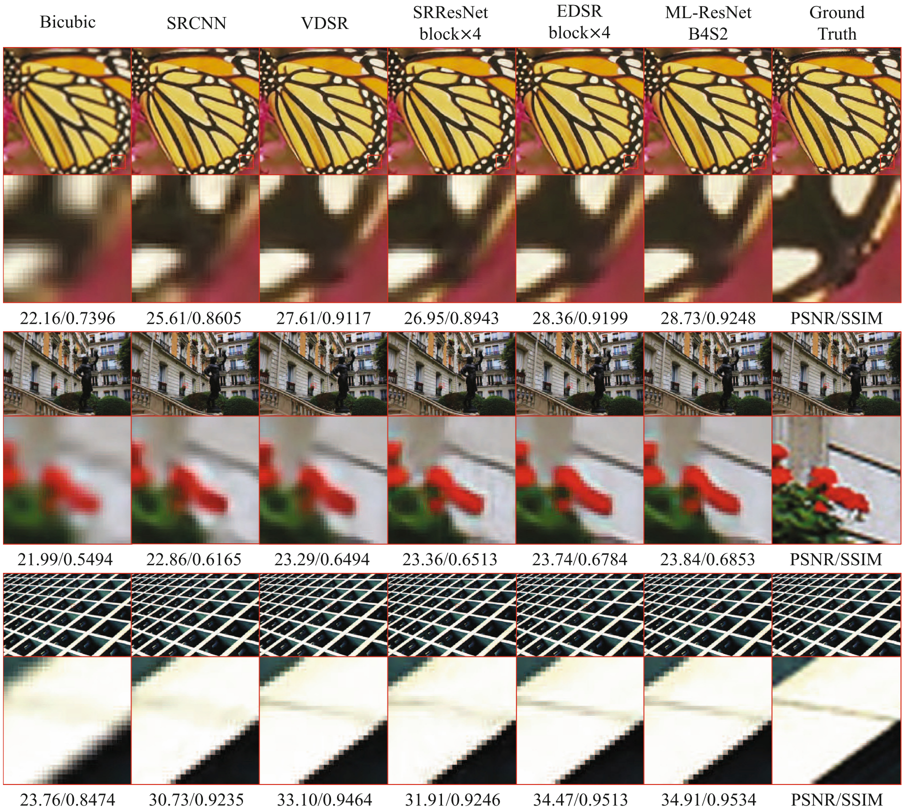


Fig. 7. Visual comparison with some previous SISR methods. (1) The first row shows image “butterfly” in Set5 with scale ×4. (2) The second and the third rows show image“img003” and “img043” in Urban100 with scale ×3.

As shown in Fig. 5 and Table 1, the proposed residual structure achieved the best SR performance. The residual block used in SRResNet [14] is obviously inferior than others. This is probably because the batch norm layer is not suitable for low-level computer vision problems. Although [16,17] removed the batch norm layer and stated its shortcomings (e.g., requires more computational and memory resources), they did not verify it experimentally.

4.4 External Skip Connection

The impact of ESC on the performance of the models is studied in this subsection. EDSR and ML-ResNet residual blocks are used for comparison. The validation performance of different architectures on the first 10 validation images of DIV2K is shown in Fig. 6.

Figure 6 exhibits an interesting phenomenon, i.e., ESC seems to have no obvious effect on the performance of the network with EDSR residual blocks but

Table 1. Quantitative comparison between some previous methods and the proposed ML-ResNet. SRResNet (block×4) and EDSR (block×4) are also included here. The maximal values are bold, and the second ones are underlined (PSNR/SSIM).

Datasets	Scale	Bicubic	SRCNN [9]	DRCN [10]	VDSR [12]
Set5	×2	33.66/0.9299	36.66/0.9542	37.63/0.9588	37.53/0.9587
	×3	30.39/0.8682	32.75/0.9090	33.82/0.9226	33.66/0.9213
	×4	28.42/0.8104	30.48/0.8628	31.53/0.8854	31.35/0.8838
Set14	×2	30.24/0.8688	32.42/0.9063	33.04/0.9118	33.03/0.9124
	×3	27.55/0.7742	29.28/0.8209	29.76/0.8311	29.77/0.8314
	×4	26.00/0.7027	27.49/0.7503	28.02/0.7670	28.01/0.7674
B100	×2	29.56/0.8431	31.36/0.8879	31.85/0.8942	31.90/0.8960
	×3	27.21/0.7385	28.41/0.7863	28.80/0.7963	28.82/0.7976
	×4	25.96/0.6675	26.90/0.7101	27.23/0.7233	27.29/0.7251
Urban 100	×2	26.88/0.8403	29.50/0.8946	30.75/0.9133	30.76/0.9140
	×3	24.46/0.7349	26.24/0.7989	27.15/0.8276	27.14/0.8279
	×4	23.14/0.6577	24.52/0.7221	25.14/0.7510	25.18/0.7524
DIV2K val	×2	31.01/0.9393	33.05/0.9581	33.70/0.9619	33.66/0.9625
	×3	28.22/0.8906	29.64/0.9138	30.15/0.9204	30.09/0.9208
	×4	26.66/0.8521	27.78/0.8753	28.23/0.8835	28.17/0.8841
		DRRN [11] B1U25	SRResNet [14] block×4	EDSR [16] block×4	ML-ResNet B4S2
Set5	×2	37.74/**0.9591**	36.94/0.9537	37.64/0.9586	**37.78**/0.9589
	×3	34.03/0.9244	33.08/0.9080	34.05/0.9234	**34.13/0.9248**
	×4	31.68/0.8888	31.01/0.8733	31.90/0.8903	**32.07/0.8921**
Set14	×2	33.23/0.9136	32.74/0.9087	33.29/0.9148	**33.32/0.9153**
	×3	29.96/0.8349	29.43/0.8232	30.00/0.8367	**30.04/0.8371**
	×4	28.21/0.7720	27.90/0.7620	28.37/0.7772	**28.45/0.7786**
B100	×2	32.05/0.8973	31.58/0.8900	32.01/0.8975	**32.11/0.8980**
	×3	**28.95**/0.8004	28.42/0.7853	28.91/0.8012	**28.95/0.8021**
	×4	27.38/0.7284	27.11/0.7185	27.48/0.7329	**27.54/0.7346**
Urban 100	×2	**31.23**/0.9188	29.84/0.9106	30.94/0.9262	31.17/**0.9278**
	×3	27.53/0.8378	26.69/0.8141	27.59/0.8409	**27.70/0.8440**
	×4	25.44/0.7638	25.01/0.7450	25.77/0.7755	**25.94/0.7808**
DIV2K val	×2	-/-	33.17/0.9582	34.36/0.9665	**34.42/0.9667**
	×3	-/-	29.47/0.9120	30.66/0.9276	**30.71/0.9283**
	×4	-/-	28.08/0.8818	28.84/0.8948	**28.93/0.8962**

it can slightly improve the performance of the model built with the proposed residual blocks. This shows the validity of the multilevel residual structure to some extent.

4.5 Comparison with Other Methods

In this section, we compare the proposed method with several typical methods quantitatively and qualitatively. When evaluating on DIV2K-val, we followed the way of EDSR/MDSR [16] to compute PSNR and SSIM; when testing on other datasets, i.e., Set5, Set14, B100 and Urban100, we followed the calculation of

DRCN [10]. Table 1 collects the quantitative results of the compared methods on the benchmark datasets, where SRResNet (block×4) and EDSR (block×4) are built with the structure shown in Fig. 2(a) and residual blocks shown in Fig. 1(b) and (c) respectively, but the number of residual blocks is limited to 4. The visual comparison is shown in Fig. 7. As we can see, ML-ResNet shows its superiority to the compared methods. It is worth noting that we only used B4S2 structure without ESC for comparison. Actually, the B2S4 structure perform better than B4S2 and ESC can further improve the performance of the model.

However, when we increase the network depth and make it have the same model parameters as the original EDSR, the performance of the proposed method is slightly worse than the original EDSR. This indicates that directly increasing the number of residual blocks to deepen the network will not get the desired performance improvement, and the multilevel residual structure promotes the propagation and the equilibrium of information flow through the network just when the network is relatively shallow.

5 Conclusion

In this paper, we studied several commonly used residual blocks for single image super resolution. Based on this, we proposed a new residual block structure and a multilevel residual learning pattern (ML-ResNet). The proposed ML-ResNet introduced fine-grained residual learning into the main path of a residual block and coarse-grained residual learning (ESC) between the outermost layers of the entire network. This multilevel residual structure seems to be helpful to simplify the structure of feature maps at multiple abstract levels of the deep model and promote the propagation and the equilibrium of information flow throughout the entire network. It shows superior performance over several compared structures when the entire network is relatively shallow. However, directly increasing residual blocks can not achieve the desired performance improvement, which may imply that the depth and internal structure of a network are related.

References

1. Hu, J., Wu, X., Zhou, J.: Single image super resolution of 3D MRI using local regression and intermodality priors. In: 8th International Conference on Digital Image Processing, vol. 10033, p. 100334C (2016)
2. Shi, W., et al.: Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8151, pp. 9–16. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40760-4_2
3. Rangnekar, A., Mokashi, N., Ientilucci, E., Kanan, C., Hoffman, M.: Aerial spectral super-resolution using conditional adversarial networks. arXiv:1712.08690 (2017)
4. Thornton, M.W., Atkinson, P.M., et al.: Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super resolution pixel-swapping. Int. J. Remote Sens. **27**(3), 473–491 (2006)

5. Pan, Z., Yu, J., Huang, H., Zhang, A., Ma, H., Hu, S., et al.: Super-resolution based on compressive sensing and structural self-similarity for remote sensing images. IEEE Trans. Geosci. Remote Sens. **51**(9), 4864–4876 (2013)
6. Juefei-Xu, F., Savvides, M.: Single face image super-resolution via solo dictionary learning. In: IEEE International Conference on Image Processing, vol. 2, pp. 2239–2243 (2015)
7. Ahmad, T., Li, X.M.: An integrated interpolation-based super resolution reconstruction algorithm for video surveillance. J. Commun. **7**(6), 464–472 (2012)
8. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
9. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016)
10. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1637–1645 (2016)
11. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2790–2798 (2017)
12. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1646–1654 (2016)
13. Shi, W., Caballero, J., Huszar, F., Totz, J., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883 (2016)
14. Ledig, C., Wang, Z., Shi, W., Theis, L., Huszar, F., Caballero, J., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 105–114 (2017)
15. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: IEEE conference on Computer Vision and Pattern Recognition, pp. 5835–5843 (2017)
16. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1132–1140 (2017)
17. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. arXiv: 1802.08797 (2018)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
19. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
20. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 391–407. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_25
21. Timofte, R., Lee, K.M., Wang, X., Tian, Y., et al.: NTIRE 2017 challenge on single image super-resolution: methods and results. In: Computer Vision and Pattern Recognition Workshops, pp. 1110–1121 (2017)
22. Bae, W., Yoo, J., Ye, J.C.: Beyond deep residual learning for image restoration: persistent homology-guided manifold simplification. In: Computer Vision and Pattern Recognition Workshops, pp. 1141–1149 (2017)

23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
24. Agustsson, E., Timofte, R.: NITRE2017 challenge on single image super-resolution: dataset and study. In: the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1122–1131 (2017)
25. Bevilacqua, M., Roumy, A., Guillemot, C., Morel, A.: Low-complexity single image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012)
26. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Boissonnat, J.-D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Mazure, M.-L., Schumaker, L. (eds.) Curves and Surfaces 2010. LNCS, vol. 6920, pp. 711–730. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27413-8_47
27. Martin, D.R., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: The 8th IEEE International Conference on Computer Vision (ICCV), pp. 416–423 (2001)
28. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Computer Vision and Pattern Recognition, pp. 5197–5206 (2015)
29. Veit, A., Wilber, M., Belongie, S.: Residual networks behave like ensembles of relatively shallow networks (2016). arXiv:1605.06431 [cs.CV]
30. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization (2014). arXiv: 1412.6980v9 [cs.LG]