Multimedia Tools and Applications (2021) 80:4327–4339
https://doi.org/10.1007/s11042-020-09954-8

Fast image super-resolution with the simplified residual network

Chunmeng Wang[1] · Lingqiang Ran[2] · Chen He[3]

Received: 7 June 2020 / Revised: 11 September 2020 / Accepted: 17 September 2020 /
Published online: 29 September 2020

Abstract
Recently, the image super-resolution (SR) methods based on residual learning have obtained remarkable quality performance. However, the current residual-learning methods have low computational performance and slow convergence rate. In this paper, we propose a high-efficiency two-level residual network to make the network learn more useful high-frequency information. Only 5 convolution layers in the LR space are used in our residual network, and no parameters are introduced in the other layers. Compared with the long training time up to several hours or days of previous deep residual networks, our simplified network can make the training time reduce to half an hour. Besides, our simplified network achieves satisfactory quality performance. The evaluation on the public datasets shows that our method can process SR of ultra-high definition (UHD) videos in real-time (more than 24 frames per second) on a generic graphical processing unit (GPU).

Keywords Super resolution · Convolutional neural networks · Simplified residual network

1 Introduction

Single image super-resolution (SISR) is an image restoration problem that reconstructs one high-quality high-resolution (HR) image from one single low-resolution (LR) image. SISR is a highly ill-posed problem because there is an infinite number of solutions from an LR image to an HR image. Recently, deep learning methods provide the promise to recover the lost details in the LR image by using the information of the training data.

✉ Chunmeng Wang
wchm87@jit.edu.cn

[1] School of Computer Engineering, Jinling Institute of Technology, Nanjing, Jiangsu 211169, China

[2] School of Computer Science and Technology, Shandong University of Finance and Economics, 250014 Jinan, Shandong, China

[3] Media and Communication College, Weifang University, Weifang, Shandong 261061, China

Recently, a variety of SR methods based on convolutional neural networks (CNN) have been proposed since SRCNN [3] introduced deep learning to solve the SISR problem. Among them, deep residual-learning methods [2, 8, 10-12, 18, 27] have provided outstanding reconstruction quality. Instead of directly learning features from the LR image to the HR image, they learned the residual between the HR image and the corresponding LR image. These deep residual-learning methods achieved significantly better quality performance than the standard CNN based method. However, the large number of parameters of these methods made the training time up to many hours or several days and the processing speed far from real-time for UHD videos on one generic GPU. These methods had large number of parameters and low efficiency for the two following reasons:

Firstly, many residual-learning methods [8, 18] transformed the LR image to HR space with the interpolation operation and train the network on the high resolution space, which enlarged the feature maps and increased computational complexity. Besides, interpolation methods do not increase substantial useful information because the interpolation operation still uses the input LR image pixels information.

Secondly, most residual-learning methods [2, 8, 27] used very deep layers for better learning quality but the number of parameters and computational complexity increased significantly. Furthermore, whether deeper residual networks can further obtain better SR performance improvement remains to be explored. Therefore, it is necessary to design an effective residual network with fewer convolution layers to reduce the number of training parameters and improve the learning efficiency. In addition, in order to keep the quality performance with fewer convolution layers, the new residual network and new training strategy are needed to make the network learn useful high-frequency residual efficiently.

We have proposed a simplified residual network architecture to solve the above problems. The main contributions of this paper are listed as follows:

(1) We propose a new two-level residual network including both external residual level and internal residual level to make the network learn more useful high-frequency information. The network further abandons the unnecessary low-frequency information in the LR image for residual learning and improve the quality performance with few convolution layers.
(2) We simplifies the very deep residual network to only 5 convolution layers with two internal residual units (IRUs) to reduce the number of parameters, and no parameters are introduced in the channel-average layer and upscale layer.
(3) We introduce the Laplacian filter to extract high-frequency from the LR image as the input, which is much closer to residual values, and then propose the soft gradient clipping (SGC) strategy to prevent the gradient explosion problem and speed up the convergence of the training process.

All the above innovations not only speed up the convergence significantly, but also gain satisfactory SR visual quality. Our method achieves much faster training and processing speed than previous residual-learning SR methods and can run SR of UHD videos in real time with a generic NVIDIA Pascal Titan X GPU.

2 Related work

The problem of single image super-resolution is a typical ill-posed problem that do not have the unique solution from LR to HR without additional constraint. The simple interpolation

operations [21, 28], such as linear interpolation, bicubic interpolation and so on, cannot restore high-frequency details in the image, so lots of SR methods have been proposed to recover details and gain high reconstruction quality.

Recently, learning methods are widely used for super resolution by modeling the mapping from LR to HR images. For example, neighbor embedding based methods [1, 6], image self-similarities based methods [5, 22] and sparse coding based methods [23-26] are all proved to be feasible. Timofte et al. [19, 20] proposed to use anchor neighborhood regression by combining sparse representation and nearest neighbor embedding. However, these methods do not obtain high-quality edge details and it is difficult to learn a versatile dictionary from a large data set.

As the development of deep learning, Dong et al. [3] first introduced convolutional neural network (CNN) to learn the end-to-end mapping from an LR image to an HR image and achieved better results than early methods. But the training network, termed SRCNN, only contains three layers but the reconstructed image quality is not ideal, and the training process takes up to one week.

Deep residual learning has been successfully applied for many fields, such as image recognition [7], object tracking [13] and video object segmentation [14, 15]. In the field of super resolution, Kim et al. [8] first proposed a very deep residual network (VDSR) with 20 layers to improve the quality of SR reconstruction. Because the LR image with low-frequency components has been given as the input, it is enough to learn the high-frequency components for SR. The residual image that contains high-frequency components can be defined as the difference between HR and bicubic interpolation of LR. Most values of residual image are likely to be zero or small, thus making easier for the network to learn. Besides, residual learning has been proved to solve the exploding gradients problem because the end-to-end training require very long-term memory with lots of weight layers. Ledig et al. [11] proposed SRGAN with perceptual losses and generative adversarial network (GAN) for photo-realistic SR. Although the generated image by SRGAN looks more detailed according to the evaluation based on MOS (Mean Opinion Score), it may produce unpleasing distortion and artifacts, and the reconstruction efficiency is low. Kim et al. [9] proposed a deeply-recursive convolutional network (DRCN) and Tai et al. [18] proposed a deep recursive residual network (DRRN) to improve DRCN by combining a residual architecture. Lim et al. [12] built a very wide network EDSR and a very deep one MDSR by using simplified residual blocks. These methods all have higher PSNR performance than SRCNN, but they interpolate the input image to a HR-spatial image before feeding it into the network and train the networks in the HR space, which increases the size of receptive field and computational complexity.

Some methods trained upscaling modules at the end of the network rather than using an interpolated image as the input. Shi et al. [17] proposed an efficient sub-pixel convolutional neural network (ESPCN) to increase the resolution from LR to HR only at the end and super-resolve HR data from LR feature maps, and it directly calculated convolution on low-resolution images to obtain high-resolution images. Dong et al. [4] proposed FSRCNN to accelerate the original SRCNN by introducing a deconvolution layer at the end of the network, and the learning is also processed on the LR space. However, all these networks learned the end-to-end mapping between LR image and downsampled HR image without residual learning, so the reconstruction quality was not so good as residual learning based methods.

Lai et al. [10] used a Laplacian pyramid architecture (LapSRN) with 27 convolution layers to increase the image size gradually to reduce the number of operations. Zhang et al. [27] proposed a residual channel attention network (RCAN) with over 400 convolution layers to achieve better accuracy by introducing residual in residual (RIR) structure and channel

attention (CA) mechanism. Dai et al. [2] proposed a deep second-order attention network (SAN) for accurate SR. These methods achieve better SR accuracy than previous residual learning methods, but the complex networks with very deep layers increase the number of parameters and cannot reconstruct for ultra-high definition (UHD) videos in real-time.

3 Our approach

3.1 The overall network

The SISR method aims to recover a high resolution image I_{SR} given a LR image I_{LR} downscaled from the original HR image I_{HR}. To downscale I_{HR} to I_{LR}, we convolve I_{HR} with a Bicubic (BI) filter model and then downsample the filtered image by the scale factor r.

We propose a simplified residual network architecture, as illustrated in Fig. 1. The residual image in the LR space I_R that contains high-frequency components can be defined as $I_R = downsampled(I_{HR}) - I_{LR}$. Most values of I_R are likely to be zero or small, thus making easier for the network to learn. In order to make initial network prediction values be close to I_R for fast convergence, we introduce one Laplacian layer to convolve I_{LR} with a Laplacian filter. Then we propose a two-level residual network including external and internal residual level to make the network learn efficiently useful high-frequency information with only 5 convolution layers, and one channel-average layer followed by convolution layers. At the end of network, we introduce one simple upscale layer to rearrange the LR pixels to construct the final I_{SR}.

3.2 Laplacian layer

The residual image could be learned by minimizing the loss function mean squared error averaged for all pixels, so the initial prediction values are important for the convergence rate. The network prediction of previous methods are always initialized as intensity values of LR for the input, but the intensity values might not be necessary as residual learning aims to predict the residual values. The residual values contain mostly zeros or small values, which are far

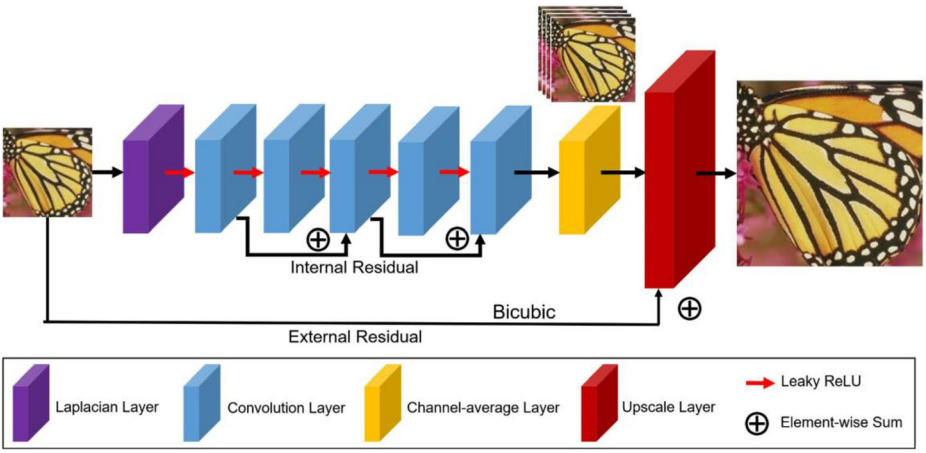


Fig. 1 The whole architecture of our simplified residual network

away from intensity values. Therefore, one Laplacian layer is introduced to optimize the initial prediction value for the network by convolving the input image I_{LR} with a Laplacian filter $LapF$ masking as following:

$$X_0 = I_{LR} \otimes LapF \tag{1}$$

where X_0 is the output feature map for the 0-th layer before convolution layers and $\otimes$ is the convolution operation. Laplacian filter can be used to detect edge and detail feature of an image, so the output values of Laplacian layer are much closer to the residual values to speed up the convergence rate.

3.3 Two-level residual learning

The low-frequency components in an image contain most complanate regions. The high-frequency components would usually be edges, texture, and details. Our method directly feeds the convolving result of LR-spatial image with Laplacian filter X_0 as the input into the following network.

In order to further speed up the convergence and improve the quality, we proposed the two-level residual network including external residual level and internal residual level in our network. The external residual level is introduced to stabilize the training of the network and makes better performance possible with residual learning at the end of the network, which makes it possible to learn residual information in a coarse level.

Internal residual unit (IRU) The internal residual level contains several internal residual units (IRUs), and each IRU links 3 convolution layers that allows the network to learn the residual between the current feature map and the previous feature map. Figure 2 shows the network architecture of one IRU. To express the implementation formally, f be a convolution function and δ be an activation function. Then, we can define the i-th residual unit R_i as follows:

$$R_i(X_j; W_j, b_j) = \delta(f_{i-1}(W_{j-2}, b_{j-2})) \oplus X_{j-2} \tag{2}$$

where W_j, b_j are the network weight and bias parameters of the j-th ($j \in 1 \cdots L$) convolution layer, respectively. X_j is the output feature map of j-th convolution layer. With this notation, we denote one channel feature of the final residual unit as $X_L(n)$, which means the n-th channel of feature map and the input to the following channel-average layer. The batch size or the channel number of our model is 64, so $n \in 1 \cdots 64$.

In order to learn more effective high-frequency information fast, the IRU allows adding the previous residual generated by the internal residual level to the current residual generated by the external residual level. Our network has only five convolution layers with two internal residual units. We can also increase the number of IRUs for more deeper residual learning, but it will increase the training and reconstruction time and improve very limited quality performance.

Channel-average layer Following the IRUs, we add one channel-average layer. We do not introduce any parameter in this layer. We use simple average operation instead of convolutional filters at the end, and all 64 channels are simply averaged into r^2 channels with the upscale factor r. For $r = 2$, every 16 input channels are averaged into one output channel, then 4 output channels are generated from 64 input channels as follows:

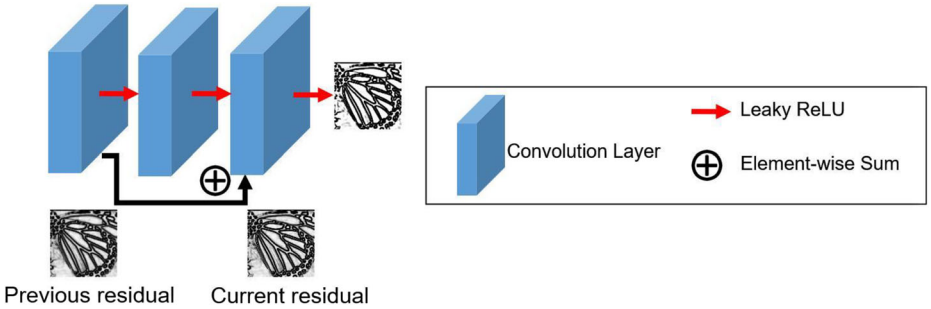


Fig. 2 One internal residual unit (IRU) in the network

$$O_L(j) = \frac{1}{16}\sum_{i=1}^{16} X_L(j*16+i-16) \tag{3}$$

For $r=3$, every 8 input channels are averaged into one output channel, but 8 input channels of them are used twice, then 9 output channels are generated from 64 input channels as follows:

$$O_L(j) = \frac{1}{8}\sum_{i=1}^{8} X_L(j*8+i-j-7) \tag{4}$$

For $r=4$, every 4 input channels are averaged into one output channel, then 16 output channels are generated from 64 input channels as follows:

$$O_L(j) = \frac{1}{4}\sum_{i=1}^{4} X_L(j*4+i-4) \tag{5}$$

where $X_L(n)$ denotes n-th ($n \in 1\cdots 64$) input channel of feature map and $O_L(j)$ denotes j-th ($j \in 1\cdots r^2$) output channel of feature map.

The two-level residual in our network helps to bypass abundant low-frequency information and make the main network learn more effective high-frequency information. It is necessary for the network to converge nicely with fewer convolution layers.

Each convolutional layer is followed by one leaky rectified linear unit (LReLU) with a negative slope of 0.1. We calculate the pixel-wise mean squared error (MSE) of the reconstruction as an objective function to train the network:

$$l(W_j, b_j) = \frac{1}{r^2HW}\sum_{x=1}^{rH}\sum_{y=1}^{rW}\left(I_{HR} - f_j(x,y)X_L(j)\right)^2 \tag{6}$$

where H and W are the height and width of the input LR image, respectively.

Soft gradient clipping (SGC) strategy Many gradient clipping strategies are often used for training CNN to prevent the gradient explosion problem and speed up the convergence of the training process, such as the hard gradient clipping (HGC) strategy to limit the gradient to a certain range $[-\theta, \theta]$, and the adjustable gradient clipping (AGC) strategy in VDSR [8] to make

the gradient threshold multiplied by learning rate as $[-\theta\gamma, \theta\gamma]$. However, it is difficult to set the uniformly suitable threshold θ because the gradient values have the large range without normalization. Instead of using the HGC or AGC strategy, we propose the soft gradient clipping (SGC) strategy to update convolutional filters by gradient norm normalization, in which gradients are normalized to the certain values as follows:

$$g^* = \frac{g}{\|g\|} \times \beta \tag{7}$$

where g is the gradient value of each filter weight, g^* is the normalized value and β is the parameter to control the clipping threshold. We use this strategy to limit the gradient components with larger absolute values more smoothly by the normalization operation for stable and fast training. We find the SGC strategy makes our convergence procedure much faster than the AGC strategy [8] as shown in Table 2.

3.4 Upscale layer

Instead of upscaling the input LR image to the high resolution space with interpolation operation at the first of the network, we increase the resolution from LR to HR only at the end of the network by a simplified upscale layer. This network avoids to perform most of the training and construction operation in the HR space and significantly reduces the computational complexity. There are several methods to upscale the feature map as the reconstruction layer, such as ESPCN [17] or transposed convolution [4]. Our upscale layer is similar to the sub-pixel convolution layer of ESPCN, but we simplify the deconvolution layer to produce an SR image I_{SR} from r^2 LR channels directly with a periodic shuffling operator in the following way:

$$I_{SR} = S(O_L(1) \cdots O_L(r^2)) \tag{8}$$

where $O_L(j)$ denotes j-th ($j \in 1 \cdots r^2$) output channel of feature map and S is the periodic shuffling operator with the same as ESPCN that rearranges the elements of r^2 channeled $H \times W \times C$ to the final elements 1-channeled $rH \times rW \times C$. The final elements construct one HR sized image. The simplified upscale layer is more efficient for low computation complexity because there is no parameter in this layer.

Finally, we produce the SR images I_{SR} by element-wise adding the constructed residual back into the original I_{LR} image with interpolation operation $interp(I_{LR})$ at the last. With the same as many previous models, such as VDSR [8], RCAN [27], SAN [2], and so on, we also use the bicubic interpolation model.

3.5 Training parameters

We only consider the luminance channel the same as previous networks because human eyes are more sensitive to the luminance than colors. We pad zeros around the boundaries before performing convolution to keep the size of all feature maps the same as the input. Each filter in a convolution layer operates with a local receptive field. Due to the reduced input resolution, we achieve smaller receptive field effectively with a smaller filter size to integrate the same information while maintaining a given contextual area. Our model uses a smaller kernel of size 3×3 for all convolutions layers except for that in the channel-average layer and upscale layer,

with the kernel size 1×1. The resolution and filter size reduction lower the computational and memory complexity.

The GPU is used for training and 30 sec per epoch for total 50 epochs. Learning rate is initially set to 0.01 and then decreases by a factor of 10 every 20 epochs.

4 Experimental results and comparisons

Our experiments run on a PC with a 3.5 GHz Intel core i7-4770 k CPU and 16 GB memory and a Pascal Titan X GPU. We use several public image datasets and one UHD video dataset as our testing database. We compare our method with several state-of-the-art SISR methods.

4.1 Datasets for training and testing

As the same as VDSR, we use 91 training images from Yang et al. [25] and 200 training images from the Berkeley Segmentation Dataset BSDS300 [16] as our training data.

For image testing experiments, we use several standard public image datasets including Set5, Set14, BSD100 and Urban100, each of which has different characteristics. For video experiments we select 10 representative ultra-high definition (UHD) videos that contains various scenes, such as landscape, buildings, people, texts, and so on. Among them, 8 UHD videos are selected from the publicly available website http://demo-uhd3d.com/ with the resolution 3840×2160, and the other two UHD videos are from Netflix UHD video database with the resolution 4096×2160, as shown in Table 3. All these videos are in .yuv format with 300 frames in length.

4.2 Comparisons with previous state-of-the-art methods

We evaluate our SR method with ground-truth HR image, bicubic and four other state-of-the-art methods including SRCNN [3], VDSR [8], RCAN [27], SAN [2] by both subjective observation and objective PSNR quality measure. We provide a summary of quantitative evaluation on several datasets.

We show comparisons for visual observation in Figs. 3, 4, 5 and 6. There are severely blurred or distorted artifacts in bicubic, SRCNN [3] and VDSR [8] results, but our method alleviates it to some degree and recovers more details because our method recovers more high-frequency information. For example, the details of the UHD video frame in Fig. 3 are clearer and vivid in our method, and the lines and lattices of the buildings in Figs. 4, 5 and 6 are reconstructed better than SRCNN [3] and VDSR [8]. Although the two state-of-the-art methods RCAN [27] and SAN [2] have gained slightly clearer SR images and better quality than our method, we still achieve satisfactory visual quality for all the testing database.

We also evaluate the SR quality performance for different methods by the objective quality measure PSNR. Table 1 shows the average PSNR values for scale factor 2, 3 and 4 on datasets Set5, Set14, BSD100, Urban100 and UHD videos dataset, and the quality ranking of these SR algorithms. Our method achieves much higher PSNR values than SRCNN [3] and VDSR [8] in these datasets. Though our method has slightly lower PSNR values than RCAN [27] and SAN [2], our simplified network is much more efficient than them and achieves a good trade-off between the algorithm complexity and construction quality.

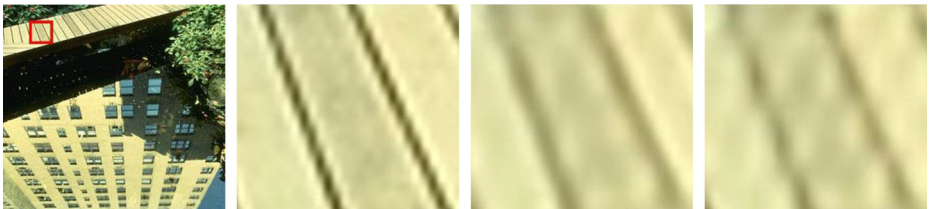
(a) UHD video(2×SR) (b) Ground-truth HR (c) Bicubic/26.83 (d) SRCNN/27.91

(e) VDSR/28.76 (f) RCAN/29.51 (g) SAN/29.64 (h) Ours/29.47

Fig. 3 The subjective observation comparison for scale factor 2 on "Hongkong.yuv" in the UHD video dataset. The PSNR values for all results of different methods are also shown

4.3 Runtime evaluations

Our method adopts the simplified residual network to improve the reconstruction efficiency significantly. The fast processing or training speed is the most important advantage of our method. Our acceleration technique includes the LR-spatial two-level residual network, the Laplacian filter for network input, fewer convolution layers, the channel-average layer instead of convolutional filters and the soft gradient cropping strategy. Our model uses a smaller kernel of size 3×3 for 5 convolutions layers with 64 channels, and there is no parameter in the



(a) BSD100(3×SR) (b) Ground-truth HR (c) Bicubic/22.42 (d) SRCNN/23.15

(e) VDSR/23.50 (f) RCAN/24.96 (g) SAN/25.17 (h) Ours/24.59

Fig. 4 The subjective observation comparison for scale factor 3 on "148026" in the dataset BSD100. The PSNR values for all results of different methods are also shown

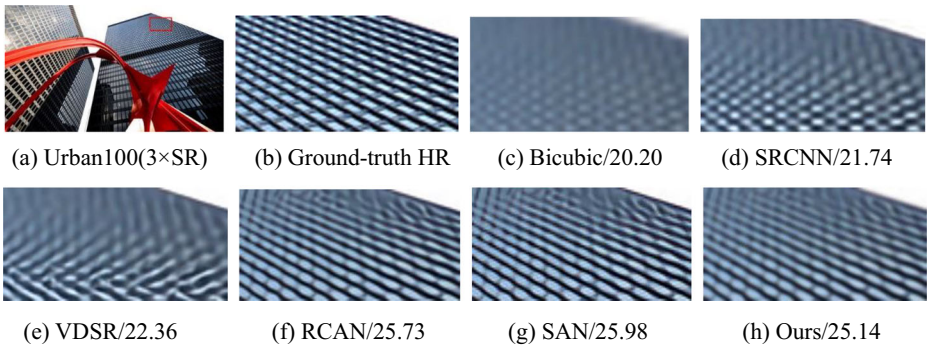


Fig. 5 The subjective observation comparison for scale factor 3 on"img062" in the dataset Urban100. The PSNR values for all results of different methods are also shown

channel-average layer and upscale layer, so the whole parameter number of our network is $3 \times 3 \times 64 \times 64 \times 4$, that is only about 147K.

We have compared the number of convolution layers, the parameter number and the training time of different networks including VDSR [8], ESPCN [17], RCAN [27], SAN [2] and ours on Set5 with an upscale factor of 2 as shown in Table 2. All these methods are trained or tested on the same GPU. Compared to VDSR model, the number of convolution required to super-resolve one image is $r \times r$ times smaller and the number of total parameters of our model is about 4.5 times smaller. The RCAN and SAN both contain more than 400 convolution layers and 15M parameters, while the convolution layers of our network are much fewer and the number of total parameters of our network is about two orders of magnitude smaller than those of RCAN and SAN.

ESPCN model [17] also performs the feature extraction stages in the LR space instead of HR space with only 3 convolution layers and 23K parameters, which achieves real time for 1080 HD (high definition) videos on a single GPU and higher computation performance than other methods, but it does not use the residual learning and treats low and high frequency information equally, which wastes unnecessary computations of low frequency features. Our model has more parameters than ESPCN, but our two-level residual network abandons low frequency information and learns more useful high-frequency information. The experiment

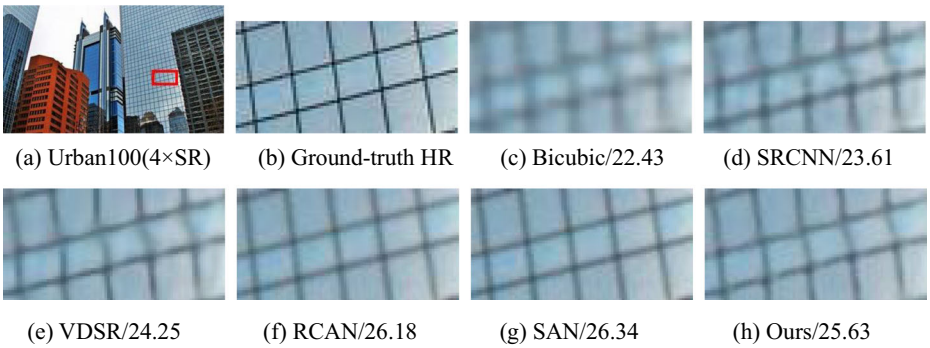


Fig. 6 The subjective observation comparison for scale factor 4 on "img035" in the dataset Urban100. The PSNR values for all results of different methods are also shown

Table 1 Average PSNR values for different scale factors on different datasets

Dataset	Scale	Bicubic	SRCNN	VDSR	RCAN	SAN	Ours
Set5	2	33.66	36.66	37.53	38.27	38.35	37.85
Set14	2	30.24	32.45	33.05	34.11	34.44	33.74
BSD100	2	29.56	31.36	31.90	32.41	32.50	32.26
Urban100	2	26.88	29.50	30.77	33.34	33.73	32.99
UHD videos	2	40.07	41.92	42.49	43.48	43.54	43.18
Average	2	32.08	34.38	35.15	36.32	36.51	36.00
Set5	3	30.39	32.75	33.67	34.74	34.89	34.25
Set14	3	27.54	29.30	29.78	30.64	30.77	30.33
BSD100	3	27.21	28.41	28.83	29.32	29.38	29.20
Urban100	3	24.46	26.24	27.14	29.08	29.29	28.51
UHD videos	3	38.11	39.32	40.03	42.01	42.17	41.46
Average	3	29.54	31.20	31.89	33.16	33.30	32.75
Set5	4	28.42	30.48	31.35	32.62	32.70	32.24
Set14	4	26.00	27.50	28.02	28.86	29.05	28.53
BSD100	4	25.96	26.90	27.29	27.76	27.86	27.58
Urban100	4	23.14	24.52	25.18	26.82	27.23	26.39
UHD videos	4	35.97	37.47	38.12	38.99	39.12	38.71
Average	4	27.90	29.37	29.99	31.01	31.19	30.69
Ranking	2,3,4	6	5	4	2	1	3

shows that the processing speed of our method is about 1.0 ms per image, about five times faster than ESPCN with the speed of 5.5 ms per image on Set5. Besides, our method achieves about 0.85 higher PSNR value than ESPCN on Set5 with the upscale factor 2.

Our LR-spatial two-level residual network achieves very fast training speed, and our soft gradient clipping (SGC) strategy also achieves much faster training speed than the adjustable gradient clipping (AGC) strategy [8]. We have recorded the time to train our network by using the two different gradient clipping strategies. It takes 1.6 hours to train our network by using the AGC strategy and if we use our SGC strategy to train our network, the training time can be reduced to half an hour. Our training speed is much faster than the VDSR, ESPCN, RCAN and the SAN models as shown in Table 2.

We also evaluate the efficiency of our super-resolution method on 10 representative UHD videos of different scenes. Table 3 shows the processing time and the frame rate when LR videos are upscaled to the target UHD videos for different scale factors. Note that the processing time is only the super-resolution time and not including the file read and write time. Our model achieves the average speed of 41 ms per frame with upscale factor of 2, 31 ms per frame with upscale factor of 3 and 22 ms per frame with upscale factor of 4. Therefore, it achieves SR of UHD videos in real-time, that is, more than 24 frames per second with the upscale factor of 2, 3 or 4.

Table 2 Computational and parameter comparison (2 × Set5)

	VDSR	ESPCN	RCAN	SAN	Ours with
Convolution layers	20	3	> 400	> 400	5
Para.	665K	23K	16M	15.7M	147K
PSNR	37.53	37.01	38.27	38.35	37.85
Training time	4 hours	3 hours	2 days	1 day	0.5 hour

Table 3 The super-resolution time (second) and the frame rate (frames per second) of our method for different scale factors

UHD video	Resolution	Scale	Time	Time Per Frame	Frame Rate
Showreel.yuv	3840 × 2160	2	12.3	0.041	24.4
Hongkong.yuv	3840 × 2160	2	12.1	0.040	24.8
Europe.yuv	3840 × 2160	2	12.2	0.041	24.6
Surfing.yuv	3840 × 2160	2	12.3	0.041	24.4
StoryofEarth.yuv	3840 × 2160	3	9.1	0.030	33.0
Fireworks.yuv	3840 × 2160	3	9.2	0.031	32.6
Skyworth.yuv	3840 × 2160	3	9.2	0.031	32.6
Shanghai.yuv	3840 × 2160	3	9.1	0.030	33.0
NETFLIX_ Chimera01.yuv	4096 × 2160	4	6.6	0.022	45.5
NETFLIX_Chimera02.yuv	4096 × 2160	4	6.7	0.022	44.8

5 Discussion and future work

In this paper, we have proposed a simplified residual network for image super-resolution. Our two-level residual network includes external residual level and internal residual level to learn more effective high-frequency information. We first introduce a Laplacian layer to process the input LR image with the Laplacian filter, which makes the input value closer to the residual value to speed up the training, and introduce the efficient channel-average layer instead of using the convolutional filters. We achieve fast training speed with the soft gradient clipping (SGC) strategy. Our approach improves the training and construction efficiency than the previous residual networks significantly. We evaluate our approach from publicly datasets to show that it achieves real-time for UHD videos and satisfactory visual quality.

Acknowledgements We would like to thank Prof. Kim Munchurl in Korea Advanced Institute of Science and Technology (KAIST) for the initial inspiration of this work. This work is supported by the Project of High-level Talents Research Foundation of Jinling Institute of Technology (No. jit-b-201802), the General Program of Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 19KJB520007), the Shandong Provincial Natural Science Foundation (Grant No. ZR2019PF023), the Project of Shandong Province Higher Educational Science and Technology Program under grant (No. J17KB184) and the Science and Technology Development Plan Project of Weifang City (No. 2019GX005).

References

1. Chang H, Yeung DY, Xiong Y (2004) Super-resolution through neighbor embedding. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 275–282
2. Dai T, Cai J, Zhang Y, Xia ST, Zhang L (2019) Second-order attention network for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR
3. Dong C, Chen CL, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision, ECCV, pp 184–199
4. Dong C, Chen CL, Tang X (2016) Accelerating the super-resolution convolutional neural network. In: European Conference on Computer Vision, ECCV, pp 391–407
5. Freedman G, Fattal R (2011) Image and video upscaling from local self-examples. ACM Trans Graph 30(2):1–11
6. Gao X, Zhang K, Tao D, Li X (2012) Image super-resolution with sparse neighbor embedding. IEEE Trans Image Process 21(7):3194–3205
7. He K, Zhang X, Ren S (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR

8. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 1646–1654
9. Kim J, Lee JK, Lee KM (2016) Deeply-recursive convolutional network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 1637–1645
10. Lai WS, Huang JB, Ahuja N, Yang MH (2017) Deep laplacian pyramid networks for fast and accurate super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 5835–5843
11. Ledig C, Theis L, Huszar F et al (2016) Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 105–114
12. Lim B, Son S, Kim H et al (2017) Enhanced deep residual networks for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 1132–1140
13. Lu X, Ma C, Ni B (2018) Deep regression tracking with shrinkage loss. In: European Conference on Computer Vision, ECCV
14. Lu X, Wang W, Ma C et al (2019) See more, know more: unsupervised video object segmentation with co-attention siamese networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR
15. Lu X, Wang W, Shen J, Tai YW et al (2020) Learning video object segmentation from unlabeled videos. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR
16. Pablo A, Michael M, Charless F, Jitendra M (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
17. Shi W, Caballero J, Huszr F et al (2016) 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, CVPR, pp 1874–1883
18. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 2790–2798
19. Timofte R, Smet VD, Gool LV (2013) Anchored neighborhood regression for fast example-based super-resolution. In: IEEE International Conference on Computer Vision, ICCV, pp 1920–1927
20. Timofte R, Smet VD, Gool LV (2014) A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Asian Conference on Computer Vision, ACCV, pp 111–126
21. Tong CS, Leung KT (2007) Super-resolution reconstruction based on linear interpolation of wavelet coefficients. Multidim Syst Signal Process 18(2):153–171
22. Wang Z, Yang Y, Wang Z, Chang S, Yang J, Huang TS (2015) Learning super-resolution jointly from external and internal examples. IEEE Trans Image Process 24(11):4359–4371
23. Wang Z, Liu D, Yang J, Han W, Huang T (2015) Deeply improved sparse coding for image super-resolution. In: IEEE International Conference on Computer Vision, ICCV, pp 370–378
24. Yang J, Wang Z, Lin Z, Cohen S, Huang T (2012) Coupled dictionary training for image super-resolution. IEEE Trans Image Process 21(8):3467–3478
25. Yang J, Wright J, Huang TS, Ma Y (2008) Image super-resolution as sparse representation of raw image patches. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR
26. Yang J, Wright J, Huang TS, Ma Y (2010) Image super-resolution via sparse representation. IEEE Trans Image Process 19(11):2861–2873
27. Zhang Y, Li K, Li K, Wang L, Zhong B, Fu Y (2018) Image super-resolution using very deep residual channel attention networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR
28. Zhou D (2010) An edge-directed bicubic interpolation algorithm. In: International Congress on Image and Signal Processing, pp 1186–1189