Encoder-Decoder Residual Network for Real Super-resolution

Guoan Cheng Ai Matsune Qiuyu Li Leilei Zhu Huaijuan Zang Shu Zhan
Department of CSIE, Hefei University of Technology, 230000, Hefei, China
guoan@mail.hfut.edu.cn, shu_zhan@hfut.edu.cn

Abstract

Real single image super-resolution is a challenging task to restore lost information and attenuate noise from images mixed unknown degradations complicatedly. Classic single image super-resolution, aims to enhance the resolution of bicubically degraded images, has recently obtained great success via deep learning. However, these existing methods do not perform well for real single image super-resolution. In this paper, we propose an Encoder-Decoder Residual Network (EDRN) for real single image super-resolution. We adopt an encoder-decoder structure to encode highly effective features and embed the coarse-to-fine method. The coarse-to-fine structure can gradually restore lost information and reduce noise effects. We empirically rethink and discuss the usage of batch normalization. Compared with state-of-the-art methods in classic single image super-resolution, our EDRN can efficiently restore the corresponding high-resolution image from a degraded input image. Our EDRN achieved the 9th place for PSNR and top 5 for SSIM in the final result of NTIRE 2019 Real Super-resolution Challenge. The source code and the trained model are available at https://github.com/yyknight/NTIRE2019_EDRN.

"cam2_09" from validation dataset

HR (PSNR/SSIM) | LR (25.09 dB/0.7377) | EDRN+ (Ours) (31.53 dB/0.9590)

Figure 1. Our result on "cam2_09", an image comes from NTIRE 2019 Real Super-resolution Challenge. Our result can restore abundant high-frequency details. + denotes the result with self-ensemble.

1. Introduction

Single image super-resolution (SISR) is a fundamental low-level vision task in computer vision. The purpose of SISR is to reconstruct the corresponding high-resolution image from a low-resolution image given. Super-resolution technologies have been leveraged in a wide range of fields such as remote sensing [23], satellite imaging [33], and medical imaging [30]. However, since there are plenty of solutions for any single input image, SISR is a highly ill-posed inverse problem. To model the inverse mapping, numerous methods for SISR, including deep-learning based ones, have been proposed. Deep learning based SISR methods have developed explosively in recent years [5, 6, 14, 15, 41]. These methods are effective for the bicubic degradation. However, the degradations for real-world low-resolution images are blind. Therefore, the recent methods cannot accurately restore real-world low-resolution images.

The real-world low-resolution images can be viewed as operating multi-degradations on the ground truth. Generally, the real-world low-resolution image I_{LR} can be formulated as:

$$I_{LR} = (I_{HR} * \omega_B) \downarrow_s + n, \tag{1}$$

where * represents the convolution operation. I_{HR} is the original high-resolution image. ω_B denotes the kernel modeled blurring and defocus of camera. $\downarrow_s$ denotes the downsampling operation with a scale factor s. n shows noise such as additive white Gaussian noise.

Real SISR can be treated as combinations of image deblurring, image denoising, and classic SISR. Hence for real

SISR, restoring high-quality high-resolution images is very complicated. Real SISR is a new task, and very few works have been proposed for this field [42]. The most related task is image restoration which solves these degradations respectively. In this field, many deep learning based methods have been proposed [41, 24, 32]. However, there remains room for improvement in these methods. First of all, most networks use convolution layers with a fixed receptive field which neglects the implicit relationship among pixels. Second, the input degraded image usually consists of multi-scale features while none of the methods differs the scales of features. Certainly, these drawbacks for image restoration should also be addressed in real SISR. Moreover, to our best knowledge, batch normalization has been treated as an unnecessary part for classic SISR, whereas many methods yet utilize it for image denoising [40, 20, 21]; How about the effects of BN in real SISR?

In this paper, to address the above issues for real SISR, we propose an encoder-decoder residual network (EDRN) for the gradual lost information restoration and noise reduction. We design the encoder-decoder model to capture relationships among large-range pixels. The structure can further encode the original image to features with more context information. For the encoded features, due to the different feature scales, we apply a coarse-to-fine method to restore high-quality image gradually. Our proposed structure can model lost information and noise reduction in each scale by residual learning. In our experiments, we apply batch normalization to the downscaling and upscaling convolution layers. Furthermore, we compare our performance with no batch normalization implementation and applying batch normalization to all the convolution layers. It demonstrates that partly implementing batch normalization can result in more accurate restoration performance.

In conclusion, our contributions can be summarized into three aspects:

1) We propose an encoder-decoder residual network (EDRN) for real SISR. The encoder-decoder structure employs a larger receptive field which improves the context information of the input shallow features.

2) We embed the coarse-to-fine method into our network, thus can restore lost information and remove noise gradually. The coarse-to-fine structure firstly reconstructs the coarse details by small features and further restores the finer details step by step.

3) We discuss the usage of batch normalization. As discussed in Sec. 4.3, applying batch normalization to downscaling/upscaling convolution layers can reduce the effect of noise and relieve overfitting.

2. Related Work

2.1. Image Super-resolution via Deep Learning

Motivated by the success of AlexNet [17] in the image recognition task, many SR methods based on deep learning are proposed. These convolutional neural network (CNN) based methods exceeded the performance of conventional hand-crafted methods in restoration quality and speed.

Dong *et al*. [5, 6] proposed the first CNN based SISR method and achieved superior improvement. FSRCNN [7] is a compact hourglass-shape CNN structure with deconvolution operations for fast training and keeping the performance. With the progress of network architectures, deeper structures have been explored and achieved significant improvement. The structure of ResNet [10] is popularly applied to make deep networks. In VDSR [14], Kim *et al*. introduced skip connection into super-resolution and demonstrated that residual learning is more efficient than direct learning. Ledig *et al*. [19] proposed SRResNet which stacked residual blocks to realize a deep network. Lim *et al*. [22] enhanced SRResNet to a very wide network EDSR and a very deep network MDSR. EDSR and MDSR stacked more residual blocks and demonstrated the unsuitability of batch normalization. Methods such as using recursive structure [15, 31], progressive reconstruction [18], and densely layer connection [35] have also been proposed. WDSR [38, 8] won the 1st place for the NTIRE 2018 Challenge on SISR in all three realistic tracks [34]. More recently, the channel attention mechanism [11] is applied to induce important features by modeling interdependencies among channels. Zhang *et al*. [43] proposed RCAN composed of channel-wise attention mechanism and Residual in Residual (RIR) structure, which performed high accuracy.

The methods mentioned above assume that low-resolution images are down-sampled from high-resolution images given. In the real-world, however, low-resolution images are degraded more complicatedly. Therefore, the performance becomes poor when existing methods are directly used for solving real-world low-resolution image in practical. SRMD [42] is the first approach to focus on super-resolving multiple degraded images by using a dimensionality stretching strategy.

2.2. Image Restoration via Deep Learning

Image restoration is a significant task in computer vision, which contains image super-resolution, image denoising, image deblurring and so on. Recently, thanks to deep learning's superior performance in image processing tasks, the development of image restoration has been promoted. Stacked Denoising Auto-encoders [36] is one of the early deep learning models proposed for image denoising. Recently, Cui *et al*. [4] proposed DNC, a cascade of multiple stacked collaborative local auto-encoders for image super-

resolution. Benefited from residual learning, Mao *et al*. [24] proposed RED, a deep convolutional auto-encoder network for image restoration. Subsequently, Zhang *et al*. [40] proposed DnCNN model for image denoising and compression artifacts reduction. In [41], Zhang *et al*. further introduced the denoiser prior to image restoration. Tai *et al*. [32] proposed a very deep end-to-end persistent memory network (MemNet) to explicitly mine persistent memory through the adaptive learning process. Moreover, NTIRE workshop [2] focuses on the new trends and advances in image restoration and enhancement, which has led state-of-the-art records.

3. Proposed Method

3.1. Encoder-decoder Residual Network

We design a large network for accurate real SISR. As shown in Fig. 2, our proposed architecture mainly contains four parts: feature encoder network (FEN), large-scale residual restoration network (L-SRRN), middle-scale residual restoration network (M-SRRN), and small-scale residual restoration network (S-SRRN). L-SRRN, M-SRRN, and S-SRRN have formed the decoder structure of our network. Inspired by RED [24], we adopt the encoder-decoder structure as our main network architecture. While we downscale/upscale the input features two times and utilize a coarse-to-fine structure to restore the lost information gradually. Moreover, we merely utilize deconvolution layers to implement upscaling operations. We further introduce residual in residual block (RIRB) into our network. We apply different numbers of RIRBs for the different spatial size features. Furthermore, we apply weight normalization [28] for all the convolution layers and batch normalization for specific layers.

We denote the input image by I_{LR}, the corresponding output image by I_{SR}. We assume the scale of lost information and interference noise as 3. First of all, in the FEN, we apply a convolution layer to extract low-level features:

$$I_0 = F_0(I_{LR}), \tag{2}$$

where F_0 denotes a convolution layer which extracts 64 features from RGB channels. I_0 is the extracted low-level features which are further used for encoding and the outermost skip connection. Subsequently, we have

$$I_1 = F_{e1}(I_0), \tag{3}$$

where F_{e1} denotes downscaling process which is composed of three operations: a convolution layer with stride 2, Rectified Linear Units (ReLU) [27], and Batch Normalization (BN) [13]. F_{e1} extracts 128 features and halves the spatial size of input features. I_1 denotes the first downscaled features which are used for second downscaling process and the second skip connection. The second downscaling process is obtained by:

$$I_2 = F_{e2}(I_1), \tag{4}$$

where the definition of F_{e2} is the same as F_{e1}. F_{e2} extracts 256 features and halves the spatial size of input features. I_2 denotes the second downscaled features which are the input of L-SRRN and the innermost skip connection.

The following L-SRRN is composed of four residual in residual blocks (RIRB) and one convolution layer. More details about RIRB is explained in Sec. 3.2. The number of filters is set as 256 in L-SRRN. The output of the L-SRRN I_L can be formulated as:

$$I_L = F_{L,lc}(F_{L,4}(\cdots(F_{L,1}(I_2))\cdots)) + I_2, \tag{5}$$

where $F_{L,lc}$ denotes the last convolution layer of the L-SRRN. $F_{L,1}$, $F_{L,2}$, $F_{L,3}$, and $F_{L,4}$ denote the RIRBs in L-SRRN. After extracting the coarse large-scale residual features, we further add the extracted features with the second downscaled ones and then input to the M-SRRN to refine them.

In M-SRRN, the input has already included the large-scale features. Hence the M-SRRN aims to restore the lost information and suppress the noise at a finer level. Furthermore, the finer features are added to the first downscaled features by long-term skip connection for keeping the memory. The formulation can be represented as:

$$I_M = F_{M,lc}(F_{M,2}(F_{M,1}(F_{dc1}(I_L)))) + I_1, \tag{6}$$

where $F_{M,lc}$ denotes the last convolution layer of the M-SRRN, $F_{M,1}$ and $F_{M,2}$ denote the RIRBs in M-SRRN, F_{dc1} denotes the deconvolution layer with stride 2, followed by ReLU layer and BN layer. F_{dc1} and the convolution layer in M-SRRN extract 128 features respectively. I_M denotes the features restored large-scale and middle-scale lost information.

Eventually, in S-SRRN, we apply one RIRB and a convolution layer for restoring the lost information and suppressing the noise in the finest level. The process can be obtained by:

$$I_S = F_{S,lc}(F_{S,1}(F_{dc2}(I_M))) + I_0, \tag{7}$$

where $F_{S,lc}$ denotes the last convolution layer of the S-SRRN, $F_{S,1}$ denotes the RIRB in S-SRRN, F_{dc2} denotes the deconvolution layer. F_{dc2} and the convolution layer in $F_{S,1}$ extract 64 features. I_S denotes the features restored all three scales lost information, which is further utilized to map onto the RGB color space.

We simply adopt a convolution layer to map the extracted features to the super-resolved high-resolution image. The output of EDRN can be obtained by:

$$I_{SR} = F_{EDRN}(I_{LR}), \tag{8}$$

where F_{EDRN} means the whole architecture of our EDRN.

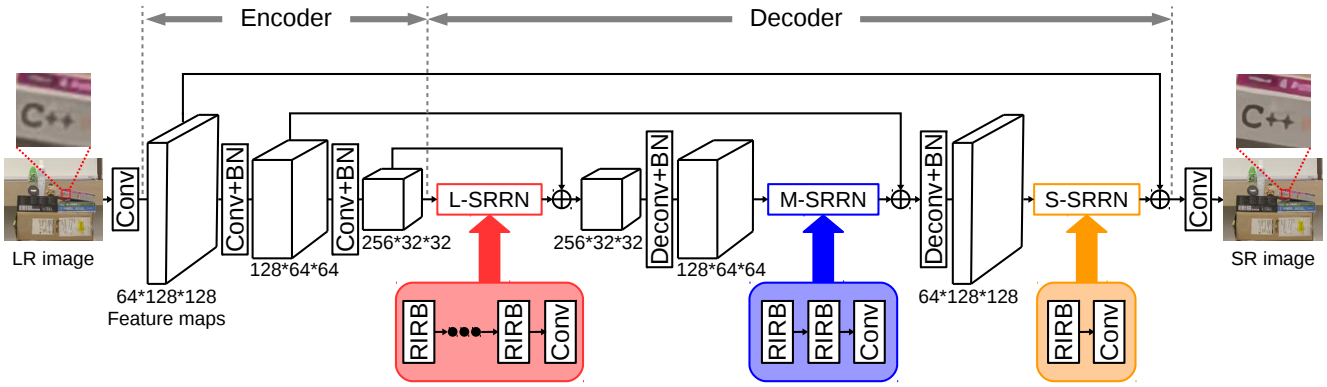

Figure 2. The architecture of Encoder-Decoder Residual Network (EDRN). $\oplus$ denotes the element-wise addition. The network consists of an encoder and a decoder. The decoder contains L-SRRN, M-SRRN, and S-SRRN, which aims to restore different-scale lost information gradually. Three skip connections are applied to preserve long-term memory for residual learning.

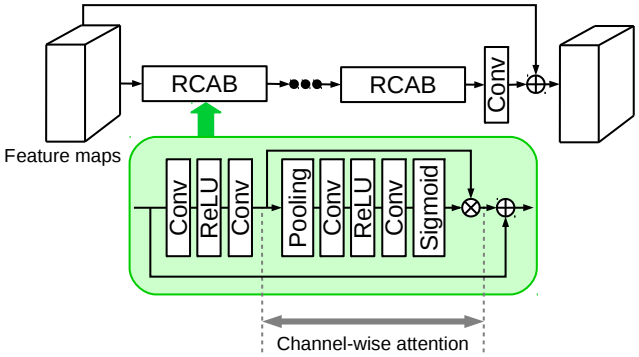

Figure 3. Residual in Residual Block (RIRB). RIRB contains several Residual Channel-wise Attention Blocks (RCAB), a convolution layer and a skip connection. In RCAB, "*Pooling*" denotes global average pooling, $\otimes$ denotes the element-wise product.

3.2. Residual in Residual Block

Zhang *et al*. [43] proposed a residual in residual structure which is composed of several residual channel-wise attention blocks (RCAB). Differed from the commonly used residual block, RCAB adopts channel-wise attention mechanism to distinguish the channel-wise significance adaptively. Hence instead of treating all features fairly, RCAB rescales the extracted residual features by the significance among channels.

Inherited this, we propose a residual in residual block (RIRB). As illustrated in Fig. 3, our RIRB stacks several RCABs, one convolution layer, and one skip connection to keep the shallow information flow. Here, we assume that the number of RCABs in one RIRB is D. The d-th output $I_{R,d}$ of RIRB can be represented as:

$$I_{R,d} = F_{R,d}(I_{R,d-1}) \quad (9)$$

$$= F_{R,d}(F_{R,d-1}(\cdots(F_{R,1}(I_{R,0}))\cdots)), \quad (10)$$

where $F_{R,d}$ denotes the d-th RCAB, $I_{R,0}$ denotes the input of the RIRB. Therefore, the output of the RIRB can be formulated as:

$$I_R = F_{R,lc}(F_{R,D}(\cdots(F_{R,1}(I_{R,0}))\cdots)) + I_{R,0}, \quad (11)$$

where $F_{R,lc}$ denotes the last convolution layer in the RIRB. The skip connection is used for keeping the previous information flow to the consecutive network. It can ease the training process and further improve the robustness of network.

3.3. Implementation Details

Here, we introduce the implementation details of our proposed EDRN. In our network, we set the number of RIRBs in L-SRRN, M-SRRN, S-SRRN as 4, 2, 1 respectively. We set the number of RCABs in our RIRB to 10. We apply the kernel with the size of 3×3 to all the convolution and deconvolution layers but the two convolution layers with 1×1 kernel in the channel-wise attention part. Weight normalization [28] is applied to all the convolution layers to ease the training process. Zero-padding is also applied to all the convolution and deconvolution layers to make the input and output keep the same size. The stride for downscaling convolution and upscaling deconvolution layers is set to 2, while the stride of the other convolution layers is 1. The number of filters for channel-wise attention layers is reduced 16 times. For the test/validation phase, due to the large resolution of test/validation images, we divide the input low-resolution image into four parts and further stitch before the final output.

4. Experimental Results

4.1. Settings

Datasets and Metrics. NTIRE 2019 released a novel datasets obtained in indoor and outdoor environments for

Real Super-Resolution Challenge. The dataset consists of 60 training images, 20 validation images, and 20 test images. The pixel resolution of each image is no less than 1000×1000 px. Owing to the ground truth of test dataset is not released, we compare and demonstrate our performance on the validation dataset. We also train a model for classic single image super-resolution on DIV2K [1], and further compare the performance with state-of-the-art methods for $\times 2$, $\times 3$, and $\times 4$ on five standard benchmark datasets: Set5 [3], Set14 [39], BSD100 [25], Urban100 [12], and Manga109 [26]. We adopt PSNR (peak signal-to-noise ratio) and SSIM (structural similarity) [37] as the evaluation metrics for all the experiments. PSNR and SSIM are calculated on the Y channel of the YC_bC_r space.

Training Settings. we randomly rotate the training images $90°$, $180°$, $270°$ and flip horizontally for data augmentation. In each batch, we input 16 RGB low-resolution (LR) patches which subtract the RGB mean of the dataset. We optimize our network by Adam optimizer [16] ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) with L1 loss and initialize the learning rate to 1×10^{-4}. For NTIRE 2019 Real Super-Resolution Challenge, we crop patches with the size of 128×128 on the training image. The initial learning rate is halved each 5×10^4 iterations. For classic single image super-resolution, we generate LR images by applying the bicubic down-sampling to the high-resolution images. We crop patches with the size of 48×48 on the LR input and halve the learning rate each 2×10^5 iterations. All the networks are implemented on PyTorch framework with an NVIDIA 1080Ti GPU.

4.2. Study of Network Structure

In this section, we discuss the effectiveness of the encoder-decoder and the coarse-to-fine structure in our EDRN.

Encoder-Decoder structure. The encoder-decoder structure is effective for mitigating the useless information and encode the primary information. To demonstrate the effectiveness of the proposed encoder-decoder structure, we first remove the encoder-decoder structure and compare. Furthermore, we investigate structures with different numbers of downscaling/upscaling convolution layers to find the best. For fair comparisons, all the implementation include 7 RIRBs, and the other training settings are strictly the same. When the encoder-decoder structure is removed, the size of intermediate feature maps is fixed. As shown in Table 1, more computations and longer runtime are costed for each image, besides that, the performance is also 0.32 dB lower than the best. Moreover, we compare the performance of downscaling/upscaling once, twice and three times. Table 1 summarizes the results. When we apply the coarse-to-fine method, comparing with downscaling twice, the performance of downscaling once is just 0.01 dB lower. Simultaneously, downscaling once also has more computations and therefore results in longer runtime. When downscaling three times, we observe faster runtime whereas the PSNR performance is 0.2 dB lower. On the other hand, when the coarse-to-fine method is not utilized, as shown in the last three lines, downscaling/upscaling once and three times would be 0.03 dB and 1 dB lower respectively. The comparison above indicates the effectiveness of the encoder-decoder structure and demonstrate applying two downscaling/upscaling operations are the best.

Coarse-to-fine structure. We further demonstrate the effectiveness of the coarse-to-fine structure. We merge all 7 RIRBs into L-SRRN and remove the RIRBs in M-SRRN and S-SRRN. As shown in Table 1, We find the network with coarse-to-fine structure has superior performance over the networks without for all downscaling/upscaling numbers. When the network only downscales/upscales once, the performance without the coarse-to-fine structure is 0.03 dB lower. When the network contains two downscaling/upscaling parts, the coarse-to-fine structure can improve around 0.01 dB. When the network contains three downscaling/upscaling parts, the coarse-to-fine structure can significantly improve the performance. The comparison above severely indicates the positive effect of the coarse-to-fine structure.

Table 1. Investigation of network structure. Note that all the networks here crop inputs with 96×96 patch size for accelerating training process. "*Down/Up*" denotes the number of downscaling/upscaling operations in the encoder-decoder structure. The performance and execution time for different structures are ranked.

Down/Up	Coarse-to-fine	PSNR (dB)	Runtime (s)
1	✓	29.92 (2)	12.40 (6)
2	✓	29.93 (1)	8.13 (5)
3	✓	29.70 (5)	7.25 (3)
0	×	29.51 (6)	18.48 (7)
1	×	29.89 (4)	7.74 (4)
2	×	29.92 (3)	5.55 (2)
3	×	28.97 (7)	4.52 (1)

4.3. Investigation of batch normalization

Batch normalization (BN) has been proved as inefficient for classic single image super-resolution (SISR). However, for real SISR, the input low-resolution image contains unknown noise, and the given training dataset is relatively small. Therefore, we decide to utilize BN to reduce the effect of unknown noise and relieve overfitting phenomenon. To our best knowledge, small batch/patch size is not suitable for BN, and different formulations for test and train are not suitable for image super-resolution. Hence we balance these ideas and use BN meticulously. To demonstrate the

validity of BN usage, we keep the same training settings and compare the performance. As shown in Table 2, When BN is not utilized, the performance is 29.66 dB. When we apply BN to all the convolution layers, the positive gain is 0.3 dB while the execution time is 0.45 seconds more. When we apply BN to downscaling/upscaling convolution layers, compared with applying BN to all the convolution layers, we observe 0.02 dB improvement and 0.3 seconds faster. The comparison above indicates that applying BN to downscaling/upscaling convolution layers is sufficient to attenuate noise. However, applying BN to all the convolution layers cannot gain improvement more. It can merely increase the execution time.

Table 2. Investigation of BN usage. "*BN(all)*" denotes applying BN to all the convolution layers, "*BN(part)*" denotes applying BN to the downscaling/upscaling convolution layers.

	PSNR (dB)	Runtime (s)
BN (all)	29.96	8.43
BN (part)	29.98	8.13
without BN	29.66	7.98

4.4. Results for NTIRE 2019 Challenge

Our EDRN is designed as a solution for the NTIRE 2019 on Real Super-resolution (SR) Challenge. In the competition, the input is low-resolution image suffered from unknown multi-degradations. Our proposed EDRN is a robust and adaptive network which can effectively restore the high-resolution images from the real-world low-resolution images. In order to verify the outstanding ability of our proposed network, owing to the particularity of the new dataset, we retrain RED30 [24], EDSR [22], RCAN [43] as comparison[1]. The quantitative results are illustrated in Table 3, our results outperform the other methods, and the results utilized self-ensemble trick as [22] can further improve. In our experiments, RED30 [24] gets the worst results due to the low parameter numbers and coarse restoration. EDSR [22] and RCAN [43] are designed for images degraded by one or two degradations. Hence they cannot achieve comparable performance. Moreover, our results achieved the 9th place in PSNR (28.79 dB) and top 5 of SSIM (0.84) in the final rank of NTIRE 2019 Real SR Challenge.

The visual comparison is illustrated in Fig. 4. Our EDRN performs better than RED30 [24], EDSR [22], and RCAN [43] on the validation dataset. For image "cam2_05", the compared methods cannot reconstruct straight and connected grid lines while our EDRN can recover most of the connected grid lines with a smooth edge and less blurring. For "cam2_04", we observe the most smooth and clean character restoration over the other compared methods. For "cam1_04", RED30 [24] and EDSR [22] reconstruct nothing but blurring, RCAN [43] reconstructs a wrong symbol. In contrast, our EDRN reconstructs the approximate shape of the symbol. For "cam2_02", RED30 [24] and EDSR [22] reconstruct the clock with little blurring, RCAN [43] cannot reconstruct the curve of the clock well whereas our EDRN reconstructs the faithful curve. As we can see, the compared methods normally suffer from blurring or noise, fail to reconstruct more lost information and even restore wrong details. However, our EDRN can accurately recover more lost information and subject to minimum noise/blurring influence. All the above comparison can demonstrate the superior restoration ability of our EDRN.

[1]Due to the explosive memory usage in test phase, we retrain smaller EDSR (64 filters with 80 residual blocks) and RCAN (7 residual groups, and 7 RCABs for each group) to maximum use the memory of our GPU.

Table 3. Quantitative results for real single image super-resolution. The results of RED30 [24], EDSR [22], and RCAN [43] are collected from our reproductivity. Best results are **highlighted**. + denotes the result with self-ensemble.

Method	PSNR (dB)
RED30 [24]	29.13
EDSR [22]	29.21
RCAN [43]	29.49
EDRN (ours)	29.98
EDRN+ (ours)	**30.10**

4.5. Results for classic single image super-resolution

To further demonstrate the effectiveness and robustness of our proposed methods, we implement our network on classic single image super-resolution. We simply replace the last convolution layer of our EDRN with an up-sample network consists of convolution layers and pixelshuffler [29]. We compare with nine state-of-the-art classic single image super-resolution methods: SRCNN[6], RED [24], VDSR [14], LapSRN [18], MemNet [32], EDSR [22], SRMDNF [42], D-DBPN [42], and RCAN [43].

In Table 4, we show the quantitative comparison for $\times 2$, $\times 3$, and $\times 4$. The results of the other methods are collected from their paper. Compared with SRMD [42], which is proposed for addressing super-resolving of multi-degradation, our results achieve a higher result to all scales in PSNR and SSIM. When comparing with the other methods, for scaling $\times 2$, our EDRN achieves similar results with EDSR [22] and D-DBPN [42] while little worse than RCAN [43]. Our EDRN just has around 74 total convolution layers. However, RCAN [43] stacks 10 residual groups consisted of 20 residual channel-wise attention blocks, which contains more than 400 convolution layers. For scaling $\times 3$ and $\times 4$, our results cannot achieve similar performance with RCAN [43], D-DBPN [42], and EDSR [22]. This situa-

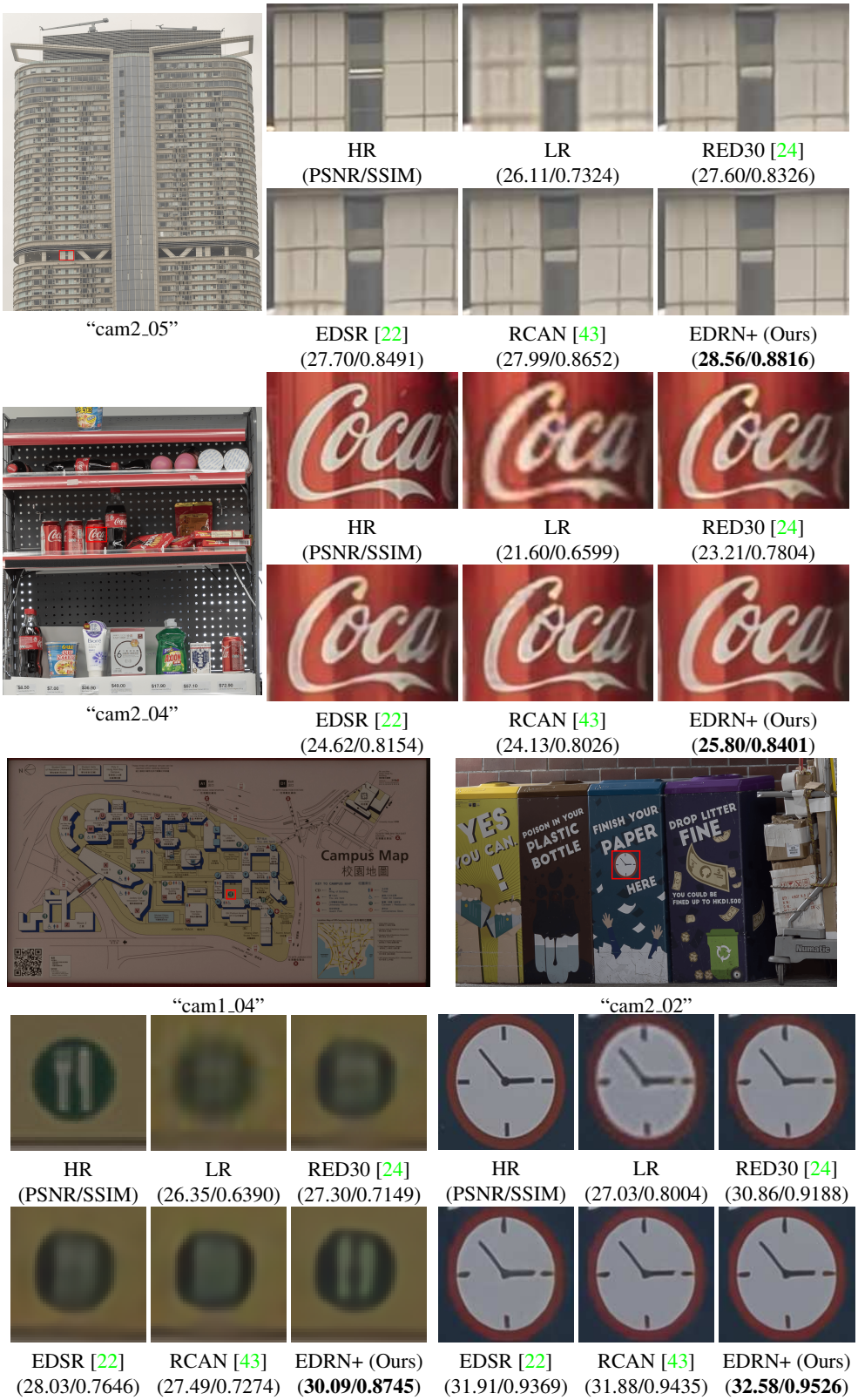

Figure 4. Visual comparison for real single image super-resolution on validation datasets of NTIRE 2019 Real Super-resolution Challenge. The best results are **highlighted**. + denotes the result with self-ensemble.

Table 4. Quantitative benchmark results for classic single image super-resolution with the bicubic degradation (average PSNR/SSIM). + denotes the result with self-ensemble.

Method	Scale	Set5 [3]	Set14 [39]	BSD100 [25]	Urban100 [12]	Manga109 [26]
Bicubic	× 2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
SRCNN [6]	× 2	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
RED [24]	× 2	37.66/0.9599	32.94/0.9144	31.99/0.8974	–/–	–/–
VDSR [14]	× 2	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140	37.22/0.9750
LapSRN [18]	× 2	37.52/0.9591	33.08/0.9130	31.08/0.8950	30.41/0.9101	37.27/0.9740
MemNet [32]	× 2	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
EDSR [22]	× 2	38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773
SRMDNF [42]	× 2	37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9761
D-DBPN [42]	× 2	38.09/0.9600	33.85/0.9190	32.27/0.9000	32.55/0.9324	38.89/0.9775
RCAN [43]	× 2	38.27/0.9614	34.12/0.9216	32.41/0.9027	33.34/0.9384	39.44/0.9786
EDRN (ours)	× 2	38.13/0.9609	33.65/0.9185	32.29/0.9010	32.35/0.9307	38.88/0.9775
EDRN+ (ours)	× 2	38.18/0.9612	33.73/0.9192	32.34/0.9016	32.52/0.9321	39.09/0.9780
Bicubic	× 3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN [6]	× 3	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
RED [24]	× 3	33.82/0.9230	29.61/0.8341	28.93/0.7994	–/–	–/–
VDSR [14]	× 3	33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290	32.01/0.9340
LapSRN [18]	× 3	33.82/0.9227	29.87/0.8320	28.82/0.7980	27.07/0.8280	32.21/0.9350
MemNet [32]	× 3	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
EDSR [22]	× 3	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
SRMDNF [42]	× 3	34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
RCAN [43]	× 3	34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702	34.44/0.9499
EDRN (ours)	× 3	34.44/0.9277	30.30/0.8420	29.11/0.8058	28.15/0.8537	33.41/0.9439
EDRN+ (ours)	× 3	34.51/0.9283	30.41/0.8436	29.18/0.8071	28.33/0.8566	33.73/0.9458
Bicubic	× 4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [6]	× 4	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
RED [24]	× 4	31.51/0.8869	27.86/0.7718	27.40/0.7290	–/–	–/–
VDSR [14]	× 4	31.35/0.8830	28.02/0.7680	27.29/0.7260	25.18/0.7540	28.83/0.8870
LapSRN [18]	× 4	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560	29.09/0.8900
MemNet [32]	× 4	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
EDSR [22]	× 4	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
SRMDNF [42]	× 4	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
D-DBPN [9]	× 4	32.47/0.8980	28.82/0.7860	27.72/0.7400	26.38/0.7946	30.91/0.9137
RCAN [43]	× 4	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	31.22/0.9173
EDRN (ours)	× 4	32.24/0.8951	28.53/0.7811	27.54/0.7355	25.92/0.7831	30.13/0.9051
EDRN+ (ours)	× 4	32.29/0.8962	28.64/0.7830	27.61/0.7372	26.11/0.7877	30.47/0.9087

tion is mainly caused by three reasons. First of all, BN is not suitable for classic single image super-resolution due to the large dataset and no noise impact. Second, the encoder-decoder structure is designed for capturing the relationship among large-range pixels. However, the input itself has a large receptive field when scaling $\times 3$ and $\times 4$; hence the operation of downscaling would lose abundant details, which make it more difficult to restore finer lost information. Third, compared with EDSR [22], D-DBPN [42], and RCAN [43], our EDRN is relatively smaller with faster execution time. As the strictly fair comparison shown, even though utilizing some inappropriate parts and a smaller network, our EDRN can still achieve comparable results. The comparison of classic single image super-resolution can further demonstrate the effectiveness and robustness of our EDRN.

5. Conclusion

In this work, we proposed an encoder-decoder residual network (EDRN) for real single image super-resolution. We introduced an encoder-decoder structure with coarse-to-fine methods. The encoder-decoder structure can extract features with more context information by the larger receptive field. The coarse-to-fine structure can gradually restore lost information and attenuate the effects of noise. We also discussed the usage of normalization. The implemented batch normalization for downscaling/upscaling convolution layers can reduce the effect of noise and relieved overfitting. In the NTIRE 2019 challenge, our EDRN could accurately restore more high-frequency details and smooth edges. In the classic single image super-resolution, our EDRN could also achieve comparable results with state-of-the-art methods.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1122–1131, July 2017. 5

[2] C. Ancuti, C. O. Ancuti, R. Timofte, L. Van Gool, L. Zhang, and M. Yang. Ntire 2018 challenge on image dehazing: Methods and results. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1004–100410, June 2018. 3

[3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. A. Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012. 5, 8

[4] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *Computer Vision – ECCV 2014*, pages 49–64, Cham, 2014. Springer International Publishing. 2

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision – ECCV 2014*, pages 184–199, Cham, Sep 2014. Springer International Publishing. 1, 2

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, Feb 2016. 1, 2, 6, 8

[7] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *Computer Vision – ECCV 2016*, pages 391–407, Cham, 2016. Springer International Publishing. 2

[8] Y. Fan, J. Yu, and T. S. Huang. Wide-activated deep residual networks based restoration for bpg-compressed images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2621–2624, 2018. 2

[9] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1664–1673, June 2018. 8

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. 2

[11] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, June 2018. 2

[12] J. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, June 2015. 5, 8

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *the 32nd International Conference on Machine Learning (ICML)*, pages 448–456, July 2015. 3

[14] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, June 2016. 1, 2, 6, 8

[15] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1645, June 2016. 1, 2

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR) 2015*, December 2015. 5

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 2

[18] W. S. Lai, J. B. Huang, N. Ahuja, and M. H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5835–5843, July 2017. 2, 6, 8

[19] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017. 2

[20] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5882–5891, July 2017. 2

[21] V. Lempitsky, A. Vedaldi, and D. Ulyanov. Deep image prior. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9446–9454, June 2018. 2

[22] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140, July 2017. 2, 6, 7, 8

[23] W. Ma, Z. Pan, J. Guo, and B. Lei. Super-resolution of remote sensing images based on transferred generative adversarial network. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1148–1151, July 2018. 1

[24] X. Mao, C. Shen, and Y. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems 29*, pages 2802–2810. Curran Associates, Inc., 2016. 2, 3, 6, 7, 8

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 5, 8

[26] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, Oct 2017. 5, 8

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, Haifa, Israel, June 2010. Omnipress. 3

[28] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems 29*, pages 901–909. Curran Associates, Inc., 2016. 3, 4

[29] W. Shi, J. Caballero, F. Huszr, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, June 2016. 6

[30] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. Marvao, T. Dawes, D. O'Regan, and D. Rueckert. Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 9–16, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 1

[31] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2798, July 2017. 2

[32] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4549–4557, Oct 2017. 2, 3, 6, 8

[33] M. W. Thornton, P. M. Atkinson, and D. A. Holland. Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping. *International Journal of Remote Sensing*, 27(3):473–491, 2006. 1

[34] R. Timofte, S. Gu, L. Van Gool, L. Zhang, and M. Yang. Ntire 2018 challenge on single image super-resolution: Methods and results. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 965–96511, June 2018. 2

[35] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4809–4817, Oct 2017. 2

[36] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 1096–1103. Omnipress, 2008. 2

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 5

[38] J. Yu, Y. Fan, J. Yang, N. Xu, X. Wang, and T. S. Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018. 2

[39] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 5, 8

[40] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017. 2, 3

[41] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, July 2017. 1, 2, 3

[42] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3262–3271, June 2018. 2, 6, 8

[43] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Computer Vision – ECCV 2018*, pages 294–310, Cham, 2018. Springer International Publishing. 2, 4, 6, 7, 8