Deep Residual Network with Enhanced Upscaling Module for Super-Resolution

Jun-Hyuk Kim and Jong-Seok Lee
School of Integrated Technology, Yonsei University, Korea
{junhyuk.kim, jong-seok.lee}@yonsei.ac.kr

Abstract

Single image super-resolution (SR) have recently shown great performance thanks to the advances in deep learning. In the middle of the deep networks for SR, a part that increases image resolution is required, for which a sub-pixel convolution layer is known as an efficient way. However, we argue that the method has room for improvement, and propose an enhanced upscaling module (EUM), which achieves improvement by utilizing nonlinear operations and skip connections. Employing our proposed EUM, we propose a novel deep residual network for SR, called EUSR. Our proposed EUSR was ranked in the 9th place among 24 teams in terms of SSIM in track 1 of the NTIRE 2018 SR Challenge [25]. In addition, we experimentally show that EUSR has comparable performance on $\times 2$ and $\times 4$ SR for four benchmark datasets to the state-of-the-art methods, and outperforms them on a large scaling factor ($\times 8$).

1. Introduction

Single image super-resolution (SR) is a fundamental computer vision problem, which refers to the process of obtaining a high-resolution (HR) image from a single low-resolution (LR) image. It is applicable to various situations where high-frequency components of images are required, including satellite and aerial imaging [24, 29], face recognition [31], medical imaging [19], and 4K-HDTV [5]. It is an ill-posed problem because it is possible to get multiple HR images from a single LR image. To deal with this problem, various conventional SR methods have been proposed based on signal processing including the methods exploiting internal information of LR images [4, 9] or information of external pairs of LR and HR images [28, 30, 1, 27].

In recent years, since the first application of convolutional neural network (CNN) to SR, called SRCNN [2], various deep learning-based SR methods that exceed the performance of the classical ones have been proposed [2, 3, 11, 12, 20, 16, 22, 14, 15, 26, 17, 6]. The deep networks for SR typically consist of two parts as illustrated in Fig. 1: 1) feature extraction part, and 2) upscaling part. The deep learning-based SR methods can be categorized into three groups according to the characteristics of the upscaling part:

- Pre-upscaling [2, 11, 12, 22, 23]: The input LR images are super-resolved by using bicubic interpolation before entering the networks, i.e., they are blurred images having the same resolution of the target HR images. As the input images pass through the networks, details with high-frequency components are restored. Since this upscaling method is not end-to-end learning but relies on hand-crafted interpolation, the performance may be limited. In addition, since the resolution of the input images is increased, it has a disadvantage in terms of computational complexity of the networks.
- Post-upscaling [3, 20, 16, 17, 26]: The input LR images are fed into the networks without changing their resolution, and the upscaling part is located at the end of the networks. Compared to the pre-upscaling method, this method utilizes end-to-end learning, which gives the possibility to go beyond the conventional hand-crafted interpolation methods. It also overcomes the disadvantage of the pre-upscaling method in terms of computational complexity.
- Progressive upscaling [15, 14]: While the above two methods increase the image resolution instantaneously, the progressive upscaling method gradually increases the resolution through the networks where the layers for feature extraction and upscaling are interleaved.

Therefore, previous researches have been conducted by changing the position of the upscaling part rather than improving the upscaling method itself. In this work, we focus on designing a new upscaling method.

In learning-based upscaling methods, transposed convolution layers [3] and sub-pixel convolution layers [20] are widely used. Since the latter uses more trainable parameters than the former for the same computational complexity [21], it has higher representation power. However, we address three issues regarding the method as follows. First, although it has a relatively good representation capability, there can be restrictions on nonlinear representation because it relies solely on linear kernels. Second, skip

connection is not used, which is used in most feature extraction parts thanks to its superiority for performance enhancement. Third, the upscaling part has relatively simple structure compared with the feature extraction part. For example, while Lim et al. [17] employs 66 convolution layers for the feature extraction part, only one convolution layer is used in the upscaling part ($\times 2$).

In order to resolve these issues, we propose a novel nonlinear upscaling module, called enhanced upscaling module (EUM). The proposed EUM utilizes residual learning and multi-path concatenation, which are described in detail in Section 3. Experimental results show that a network with EUM has better performance in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), compared to one with sub-pixel convolution layers. By utilizing the EUM with a deep residual network and multi-scale learning [17], we propose a novel deep network for SR (EUSR).

Our proposed EUSR was ranked in the 9th place among 24 teams in terms of SSIM in track 1 of the NTIRE 2018 SR Challenge [25]. In addition, experimental results show that the proposed method outperforms state-of-the-art deep learning-based SR ($\times 8$) methods on benchmark datasets.

2. Related works

2.1. CNN architectures for SR

Similarly to the superior performance of CNN in other computer vision fields, CNN-based SR methods also have shown higher performance compared with conventional methods [4, 9, 28, 30, 1, 27]. They can be divided into four categories according to the characteristics of the employed CNN structures.

First, an early attempt, i.e., SRCNN, employs a very simple structure of CNN, which consists of only three convolution layers. Even with the simple structure, it outperforms other hand-crafted methods.

Second, ResNet [7], the CNN structure that enables overwhelming performance improvement in various computer vision problems including image classification, is popularly employed [11, 12, 16, 17, 22, 23, 14]. The structure contributes to make it possible to design CNNs with deeper structures. As deeper CNNs can exploit more contextual information of images with larger receptive fields, the ResNet-based SR methods achieve higher reconstruction quality compared with SRCNN. In VDSR [11], 20 convolution layers are used with global skip connection. To overcome the slow convergence rate due to its complicated structure, the learning rate is set high. The network also uses gradient clipping to make training phase stable. By applying several techniques including batch normalization [10] and post-upscaling with the sub-pixel convolution layers, Ledig et al. [16] propose a deeper and superior ResNet-based SR method than VDSR. In particular, global skip connection as well as local ones are used to ensure stable training. Lim et al. [17] experimentally investigate the optimal ResNet structure for SR and consequently some unnecessary parts including batch normalization are eliminated. Utilizing the obtained optimized structure, they propose two networks: EDSR and MDSR, which show the state-of-the-art reconstruction image quality. Especially, MDSR exploits multi-scale information simultaneously, which enables lower complexity while maintaining comparable performance when compared with EDSR. In addition to the above, most recent approaches utilize the ResNet architecture to ensure high performance.

In addition to the deep CNNs for SR based on ResNet, Tong et al. [26] first employ the structure of DenseNet [8] for SR, called SRDenseNet. While ResNet aggregates features via addition operations, this structure combines features via concatenation, which allows features to pass through to the subsequent layers without modification.

While the methods mentioned above focus on performance in terms of PSNR and SSIM, there also exist deep CNNs that consider both reconstruction quality and complexity of the networks [12, 22]. To solve the problem that deep CNNs are too complex to be applied in practice, they use recursive structures and skip connections at the same time, which effectively maintains performance while reducing the number of trainable parameters.

2.2. Sub-pixel convolution layer for SR

The sub-pixel convolution layer is first introduced in [20], which is an upscaling method conducted in the LR space. It consists of two sequential operations, i.e., one convolution layer and one periodic shuffling operator. First, the convolution layer extracts r^2C feature maps having a size of $H \times W$, where H, W, and C are the height, width, and number of channels of the LR image and r is the target scaling factor. After the convolution layer, the periodic shuffling operator rearranges the feature maps into the final HR image having a size of $rH \times rW \times C$.

In EDSR and MDSR, the sub-pixel convolution layer is applied slightly differently from the originally proposed method. The red box in Fig. 1 shows its structure. While the original method yields the final HR image as a result, the sub-pixel convolution layer in both networks outputs super-resolved feature maps, which are reconstructed into the final HR image through one additional convolution layer (the last blue box in Fig. 1). For example, if the size of the input fed into the sub-pixel convolution layer is $H \times W \times F$, the sub-pixel convolutional layer yields F feature maps having a size of $rH \times rW$. From this, the additional convolution layer extracts the final HR image having a size of $rH \times rW \times C$.

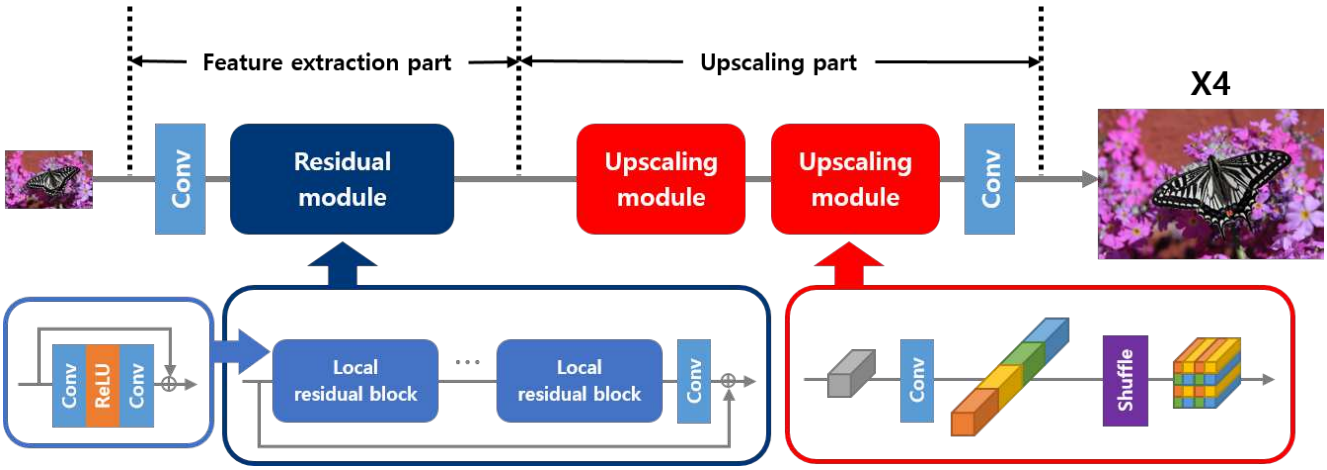


Figure 1: Overall architecture of our baseline. Each rectangular parallelepiped represents feature maps obtained at the corresponding location.

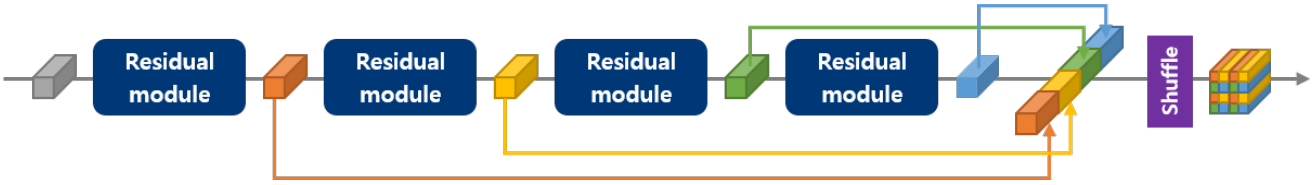


Figure 2: Structure of our proposed EUM to replace the upscaling module (red box) in Fig. 1. Each rectangular parallelepiped represents feature maps obtained at the corresponding location.

3. Proposed methods

In this paper, there are two newly named modules to describe deep networks for SR:

- Residual module: The navy box in Fig. 1 shows its structure. It consists of several local residual blocks, one convolution layer, and global skip connection. The configuration of the residual module depends on the number of local residual blocks.
- Upscaling module: Its structure is shown as the red box in Fig. 1. It corresponds to a sub-pixel convolution layer that doubles the resolution.

3.1. Enhanced upscaling module

Regarding the three issues mentioned in the introduction, we pose the following questions related to the upscaling modules:

- Does adding nonlinear operations to the upscaling module help to improve performance?
- Is the skip connection applicable and effective to the upscaling module as in feature extraction?
- Is it helpful to allocate more trainable parameters to the upscaling module?

Inspired by the above questions, we present a novel EUM and experimentally verify its effectiveness, where the experiments are conducted with the baseline (single-scale) used in [17] and the target scaling factor is set to four (in each of the horizontal and vertical axes). Fig. 1 shows the architecture of the baseline. It can be divided into four parts: 1) the first convolution layer that extracts feature maps from input RGB images, 2) a residual module that has 16 local residual blocks, 3) two upscaling modules, and 4) the last convolution layer that converts feature maps into output RGB images.

The proposed EUM is illustrated in Fig. 2. There are some differences between the original upscaling module (the red box in Fig. 1) and our proposed EUM. First, it replaces the complex convolution layer (the blue rectangle in the upscaling module) by concatenating the outputs of four consecutive modules (the orange, yellow, green, and blue rectangular parallelepipeds in Fig. 2). While the complex convolution layer in the original upscaling module produces four times as many feature maps as its input, each of the four modules in EUM produces the same number of feature maps as its input. Therefore, a global skip connection can be included in each module, which makes it a "residual" module. In addition, since a residual module, which includes at least one ReLU activation function, is utilized

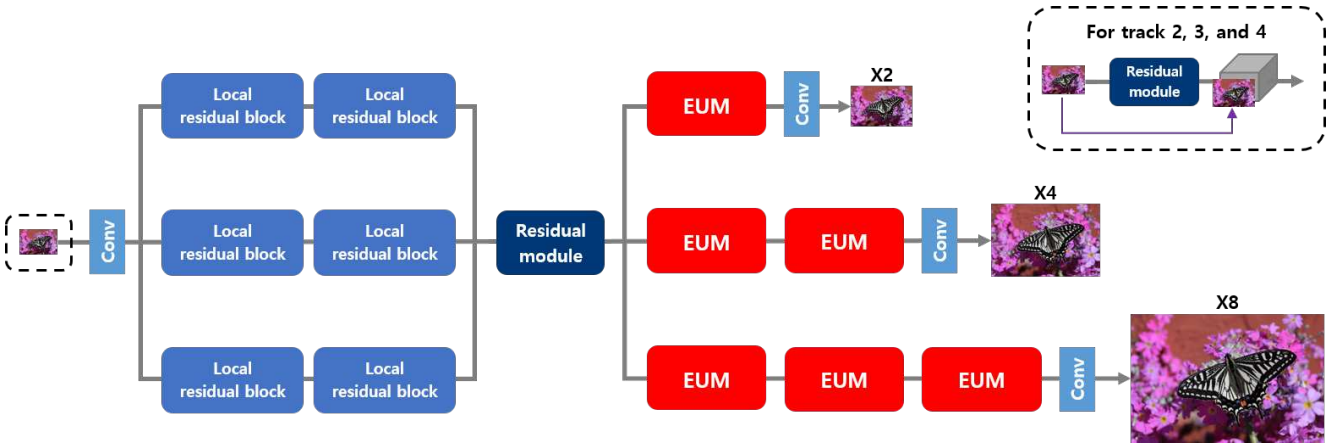


Figure 3: Overall architecture of our proposed EUSR. For tracks 2, 3, and 4 of the NTIRE 2018 SR Challenge [25], the input images are replaced by the dashed box at the top right. The target scaling factors are unified to four, and the data of the different tracks is fed into each path ($\times 2$, $\times 4$, and $\times 8$).

as each module, it is possible to perform nonlinear computation. After the concatenation of multi-path feature maps, a periodic shuffling operation (the purple rectangle in Fig. 2) proceeds as in the original architecture. The characteristics of the proposed EUM are summarized as follows:

- It is a module that doubles the resolution, consisting of four residual modules having the same structure and one periodic shuffling operator. As the input to the periodic shuffling operator, a concatenation of outputs from the four residual modules is used.
- Its configuration varies with the number of local residual blocks in each residual module.
- It can take advantages of skip connection and nonlinear operations thanks to the use of residual modules.

We experimentally show that the EUM helps improve SR performance on four different datasets. In addition, we also measure the performance change according to the relative amounts of parameters of the feature extraction part and the upscaling part in order to show the importance of the upscaling part. The experimental results are described in Section 4.

3.2. Proposed network architecture

The overall architecture of our proposed EUSR is illustrated in Fig. 3. Previous studies [11, 17] have shown that SR for different scales are related to each other, and therefore training multiple scales together is effective in terms of both reconstruction quality and network complexity. Inspired by this, we also use the multi-scale learning strategy using three different scales ($\times 2$, $\times 4$, and $\times 8$). While our network is based on the structure of MDSR, we use fewer local residual blocks in the residual module (80 vs. 48) of the feature extraction part and use the proposed EUM in the upscaling part. Each residual module used in each EUM has two local residual blocks.

For tracks 2, 3, and 4 in the NTIRE 2018 SR Challege [25], there are a few additions. First, we use three additional feature maps as input, which are obtained by a residual module having three local residual blocks. By doing so, we try to measure the degradation information of input images. Second, we use more local residual blocks (64 instead of 48) in the residual module for the feature extraction part.

4. Experiments

4.1. Datasets

We use the DIV2K dataset [25], which consists of 1000 2K resolution RGB images. LR training images are provided in different ways for each track of the NTIRE 2018 SR Challenge [25]. For track 1, 800 LR images ($\times 8$) downscaled by using bicubic interpolation are provided to participants. To exploit the multi-scale learning strategy, we also use 800 LR images ($\times 2$) and 800 LR images ($\times 4$). For tracks 2 and 3, 800 LR images ($\times 4$) are provided, respectively. The degree of degradation is different: mild and difficult for each track, respectively. The same degradation process is applied in all the images of each track. 3200 LR images ($\times 4$) are provided in track 4, where unlike the other tracks, the degradation process varies from one image to another.

While the original 800 HR images are used as HR training images in track 1, data pre-processing is required in other tracks because there is a problem that LR images and HR images are not aligned correctly due to non-bicubic downscaling methods. We match the corresponding LR

and HR pairs using the following ways: 1) We convert the LR and their corresponding bicubic-downscaled LR images from RGB to grayscale. 2) The center of the LR images is cropped except for 40 pixels from each edge. 3) We measure the normalized cross-correlation between the cropped LR images and the corresponding bicubic-downscaled LR images. 4) The regions of the bicubic-downscaled LR images that have the highest correlation with the cropped LR images are detected, and only the corresponding regions are cropped from the HR images. 5) The cropped LR and HR image pairs are used for training.

For testing, we evaluate performance of our networks on four datasets widely used for SR benchmark[1]: Set5 [1], Set14 [30], BSD100 [18], and Urban100 [9]. Set5 and Set14 consist of 5 and 14 images, respectively. The BSD100 consists of 100 natural images taken from the Berkeley segmentation dataset [18], which is created for research on image segmentation and boundary detection. Urban100 includes 100 challenging images with indoor, urban, architectural scenes, etc., which rarely appear in other datasets.

4.2. Implementation Details

In each mini-batch, we feed 16 randomly cropped 32×32 patches from LR images into our networks. Before being fed into the networks, the patches are transformed through random rotation by three angles (90°, 180°, and 270°) or random horizontal flips. The process has the effect of augmenting the data by eight times. It is noted that the process is applied only to track 1. Each patch is normalized by subtracting the mean value of all training images for each RGB channel. For track 1, the target scaling factor is randomly selected among three different scales (×2, ×4, and ×8). For the other three tracks, the target degradation method is randomly selected among three different methods (mild, difficult, and wild).

In all convolution layers except the first two local residual blocks, the size of filters is set to 3×3. For the two blocks, 5×5 filters are used. The convolution layers that are used to make the input for the three tracks (the dashed box in Fig. 3) and final HR images produce three feature maps, and all the remaining convolution layers generate 64 feature maps. We use the L1 loss function and the Adam optimizer [13] with $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\epsilon = 10^{-8}$ for training. The initial learning rate is set to 10^{-4}, which falls in half every 2×10^5 mini-batch updates. All networks for analyzing EUM (Section 4.3) are trained with 10^3 mini-batch updates. The two final networks, one for track 1 and the other for the other tracks, are trained with 10^6 mini-batch updates for track 1 and others, respectively. We implement the networks using the Tensorflow framework.

[1]To improve performance, we used the DIV2K validation images during training, so DIV2K is excluded from the evaluation.

Dataset	baseline	baseline with EUM
Set5	32.07 / 0.892	32.06 / 0.893
Set14	28.37 / 0.776	28.39 / 0.777
BSD100	27.52 / 0.734	27.54 / 0.735
Urban100	25.91 / 0.780	25.99 / 0.783

Table 1: Performance of our proposed EUM in terms of PSNR/SSIM. Both networks have about 1518K parameters. The red color indicates the better one.

Dataset	F16U1	F8U2
Set5	32.14 / 0.894	32.17 / 0.894
Set14	28.42 / 0.778	28.44 / 0.779
BSD100	27.56 / 0.735	27.57 / 0.736
Urban100	26.07 / 0.786	26.09 / 0.786

Table 2: Comparison of F16U1 and F8U2 in terms of PSNR/SSIM. Both networks have about 2108K parameters. The red color indicates the better one.

It roughly takes 4 days with NVIDIA GeForce GTX 1080 GPU to train each final network.

4.3. Analysis

Effectiveness of EUM. To demonstrate the effectiveness of the proposed EUM, we replace the upscaling module, i.e., sub-pixel convolution layer, in the baseline with EUM while the feature extraction part remains the same structure. For fair comparison, each residual module in each EUM uses only one local residual block with bottleneck structure to maintain a similar number of parameters. Table 1 shows that the baseline with EUM outperforms the original baseline on the four different datasets.

Importance of upscaling part. We construct two networks to demonstrate the importance of the upscaling part. Both networks employ two EUMs (i.e, ×4 SR) and have the same number of parameters. They are distinguished by the configuration of the residual modules in the feature extraction part and upscaling part:

- F16U1: Its feature extraction part is the same as the baseline, where the residual module (the navy box in Fig. 1) has 16 local residual blocks. Each residual module of the upscaling part (the navy box in Fig. 2) has one local residual block.

- F8U2: Compared to F16U1, while the residual module of the feature extraction part has less local residual blocks (8 instead of 16), each residual module of the upscaling part has more local residual blocks (2 instead

of 1).[2]

Table 2 shows that F8U2 achieves better performance for all datasets, which confirms that a certain level of complexity is required in the upscaling part.

4.4. Comparison with state-of-the-art methods

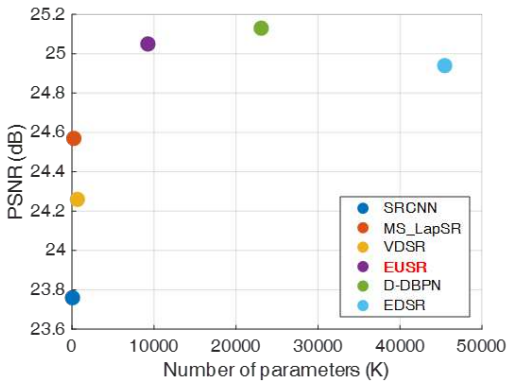
To evaluate the performance of our proposed network (EUSR), we use five state-of-the-art SR methods: SRCNN [2], VDSR [11], MS_LapSRN [15], EDSR [17], and D-DBPN [6].

Parameter efficiency. Fig. 4 shows the parameter efficiency of EUSR on Set14 and BSD100. Among the six networks, EDSR, D_DBPN, and EUSR show higher PSNR values than the others. In particular, EUSR records the second best performance in Set14 and the best performance in BSD100. It should be noted that EUSR achieves this performance with only about 20% and 40% of parameters of EDSR and D-DBPN, respectively.

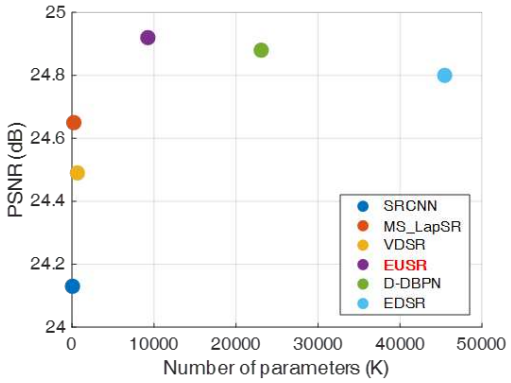
Performance comparsion. We use PSNR and SSIM to quantitatively evaluate EUSR. For fair comparison, we measure the performance in the same way to that used in the other methods: 1) All images are converted to YCbCr channels and evaluated using only the Y channel. 2) All images are cropped by the same amount of pixels, i.e., target scaling factor, from the boundary before evaluation.

We provide the quantitative evaluation result of our EUSR in Table 3. In addition, we present the result of EUSR+, which uses the geometric self-ensemble [17] in the testing process, in the last column of Table 3. EUSR is superior to the state-of-the-art methods for all datasets and all scales except for EDSR and D-DBPN. The three networks have comparable performance, but each shows strength in different aspects. For $\times 2$ and $\times 4$ SR, EDSR exhibits relative superiority in all datasets. Note that D-DBPN and EUSR have similar performance, but EUSR is more effective in more difficult datasets (BSD100 and Urban100). For example, for $\times 4$ SR, EUSR has lower PSNRs (-0.01dB and -0.16dB) in Set5 and Set14 than D-DBPN, while it has higher PSNRs ($+0.02$dB and $+0.15$dB) in BSD100 and Urban100. D-DBPN and EUSR show particularly good performance on $\times 8$ SR compared to the others. As in the other scales, it is worth noting that the proposed EUSR is particularly effective on difficult datasets. In addition, EUSR+ shows overall performance improvement over EUSR.

We also provide the qualitative evaluation results of $\times 8$ SR in Fig. 5. One test image is selected from each of the four datasets. It can be observed that EUSR successfully restores HR images compared to the other methods. As confirmed by the quantitative comparisons, EUSR shows better performance, especially in more difficult cases. For example, when the results of D-DBPN and EUSR in the third and last rows are compared, D-DBPN reconstructs a pattern that is different from the ground truth images, or a pattern that does not exist, whereas the results of EUSR are close to the patterns in the ground truth.



(a) Set14



(b) BSD100

Figure 4: PSNR (dB) vs. number of parameter (K) for $\times 8$ SR.

5. Conclusion

In this paper, we proposed an enhanced upscaling module (EUM), which can handle nonlinear operations and exploit skip connections. We experimentally showed that our proposed EUM is more efficient than the existing method, sub-pixel convolution layer.

Furthermore, we proposed a novel deep residual network with EUM (EUSR). By utilizing EUM and multi-scale learning, it uses parameters more efficiently and also shows comparable performance on benchmark datasets compared to the state-of-the-art methods (EDSR and D-DBPN). In

[2]Further increasing the number of local residual blocks in each residual module of the upscaling part (i.e., making U3) is not possible while the total number of parameters is maintained, because in that case all the parameters are spent in the upscaling part and the feature extraction part is vanished.

Dataset	Scale	Bicubic	SRCNN [2]	VDSR [11]	MS_LapSRN [15]	EDSR [17]	D-DBPN [6]	**EUSR**	**EUSR+**
Set5	×2	33.65 / 0.930	36.65 / 0.954	37.53 / 0.958	37.78 / 0.960	38.11 / 0.960	38.05 / 0.960	37.98 / 0.960	38.08 / 0.960
	×4	28.42 / 0.810	30.49 / 0.862	31.35 / 0.882	31.74 / 0.889	32.46 / 0.897	32.40 / 0.897	32.39 / 0.897	32.51 / 0.898
	×8	24.39 / 0.657	25.33 / 0.689	25.72 / 0.711	26.34 / 0.753	26.97 / 0.775	27.25 / 0.785	27.20 / 0.785	27.29 / 0.788
Set14	×2	30.34 / 0.870	32.29 / 0.903	32.97 / 0.913	33.28 / 0.915	33.92 / 0.919	33.79 / 0.919	33.53 / 0.916	33.64 / 0.917
	×4	26.10 / 0.704	27.61 / 0.754	28.03 / 0.770	28.26 / 0.774	28.80 / 0.788	28.75 / 0.785	28.59 / 0.783	28.70 / 0.784
	×8	23.19 / 0.568	23.85 / 0.593	24.21 / 0.609	24.57 / 0.629	24.94 / 0.640	25.14 / 0.649	25.05 / 0.644	25.13 / 0.646
BSD100	×2	29.56 / 0.844	31.36 / 0.888	31.90 / 0.896	32.05 / 0.898	32.32 / 0.901	32.25 / 0.900	32.24 / 0.900	32.30 / 0.901
	×4	25.96 / 0.669	26.91 / 0.712	27.29 / 0.726	27.43 / 0.731	27.71 / 0.742	27.67 / 0.738	27.69 / 0.739	27.74 / 0.741
	×8	23.67 / 0.547	24.13 / 0.565	24.37 / 0.576	24.65 / 0.592	24.80 / 0.596	24.91 / 0.602	24.92 / 0.601	24.96 / 0.603
Urban100	×2	26.88 / 0.841	29.52 / 0.895	30.77 / 0.914	31.15 / 0.919	32.93 / 0.935	32.51 / 0.932	32.54 / 0.932	32.74 / 0.933
	×4	23.15 / 0.659	24.53 / 0.724	25.18 / 0.753	25.51 / 0.768	26.64 / 0.803	26.38 / 0.793	26.53 / 0.799	26.68 / 0.802
	×8	20.74 / 0.516	21.29 / 0.543	21.54 / 0.560	22.06 / 0.598	22.47 / 0.620	22.72 / 0.630	22.73 / 0.633	22.89 / 0.637

Table 3: Quantitative evaluation results in terms of PSNR/SSIM. Red and blue colors indicate the best and second best methods, respectively, except for EUSR+.

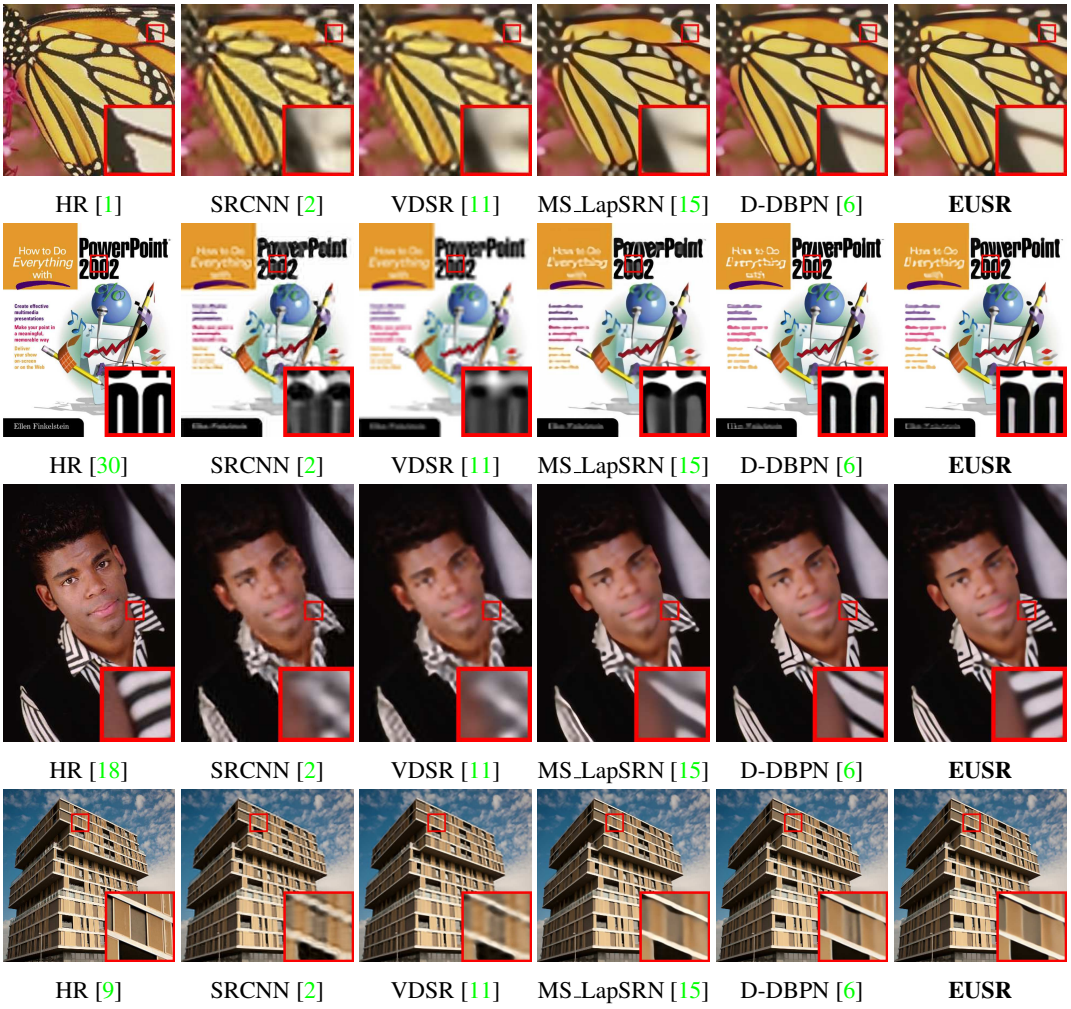


Figure 5: Qualitative evaluation results of the five methods on ×8 SR. From the top: butterfly.png from Set5, ppt3.png from Set14, 302008.png from BSD100, and img_087.png from Urban100.

particular, it has the advantage of solving more difficult SR problems better compared with the other methods.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (NRF-2016R1E1A1A01943283).

References

[1] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 135.1–135.10, 2012. 1, 2, 5, 7

[2] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199, 2014. 1, 6, 7

[3] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 391–407, 2016. 1

[4] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics*, 30(2):12:1–12:11, 2011. 1, 2

[5] T. Goto, T. Fukuoka, F. Nagashima, S. Hirano, and M. Sakurai. Super-resolution system for 4K-HDTV. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 4453–4458, 2014. 1

[6] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. *arXiv preprint arXiv:1803.02735*, 2018. 1, 6, 7

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

[8] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 2

[9] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015. 1, 2, 5, 7

[10] S. Ioffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, 2015. 2

[11] J. Kim, J. Lee, and K. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016. 1, 2, 4, 6, 7

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

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 624–632, 2017. 1, 2

[15] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *arXiv:1710.01992*, 2017. 1, 6, 7

[16] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 1, 2

[17] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140, 2017. 1, 2, 3, 4, 6, 7

[18] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423, 2001. 5, 7

[19] Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, and A. Ozcan. Deep learning microscopy. *Optica*, 4(11):1437–1443, 2017. 1

[20] W. Shi, J. Caballero, F. Huszár, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 1, 2

[21] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang. Is the deconvolution layer the same as a convolutional layer? *arXiv preprint arXiv:1609.07009*, 2016. 1

[22] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2798, 2017. 1, 2

[23] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4539–4547, 2017. 1, 2

[24] M. W. Thornton, P. M. Atkinson, and D. 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

[25] R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, et al. Ntire 2018 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 1, 2, 4

[26] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4809–4817, 2017. 1, 2

[27] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, 2012. 1, 2

[28] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010. 1, 2

[29] Q. Yuan, L. Zhang, and H. Shen. Multiframe super-resolution employing a spatially weighted total variation model. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(3):379–392, 2012. 1

[30] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Proceedings of the International Conference on Curves and Surfaces*, pages 711–730, 2010. 1, 2, 5, 7

[31] W. W. Zou and P. C. Yuen. Very low resolution face recognition problem. *IEEE Transactions on Image Processing*, 21(1):327–340, 2012. 1