Self-Adaptive Weighted Skip Connections for Image Super-Resolution

Jiachen Wang Yingyun Yang
School of Information and Communication Engineering, Communication University of China
356056849@qq.com yingyun6903@163.com

***Abstract*—Recently, the introduction of deep convolutional neural networks has achieved great performance in low-level vision tasks like single image super-resolution. However, deeper networks tend to have larger number of parameters and be more difficult to be trained. Considering massive low-frequency information in low-resolution inputs, we propose a self-adaptive weighted skip connections (SAWSC) structure to make full use of both the low-level and high-level features in order to better the representation ability of super-resolution networks. In our proposed network, we follow a coarse-to-fine strategy, which reconstructs high-resolution images progressively based on the Laplacian pyramid. At each upscale level, feature maps of each block are connected to subsequent blocks with self-adaptive weights. During each block, several residual channel attention layers are cascaded. Evaluations on five public benchmark datasets show that this algorithm achieves better performance than some other existing methods.**

Keywords-single image super-resolution; deep convolutional neural network; Laplacian pyramid; self-adaptive weighted skip connections

I. INTRODUCTION

Image super-resolution (SR) is one of the branches of image restoration technology, belonging to low-level vision tasks, and aims to reconstruct high-resolution (HR) images from the low-resolution images. Usually, we refer to the problem of generating a HR image with the input of a single LR image as single image super-resolution (SISR). Considering the ill-posed inverse problem that there exist many different mapping functions from a single LR image to its corresponding HR image, more and more methods are exampled-based, trying to learn an accurate mapping from LR image patches to their HR counterparts on datasets which consist of LR and HR image pairs.

Since Dong et al. [1] firstly introduce the use of convolutional neural network (CNN) into image super-resolution, despite its shallow structure with only three layers, following CNN-based methods have greatly enhanced both the reconstruction accuracy and visual details. Some works [2, 3, 4, 5, 6] succeed in building relatively deep and wide CNNs to improve the quality of image SR, proving that increasing the complexity of networks which is widely used in other high-level vision tasks still works in this field. In order to solve the problem of vanishing gradient caused by increasing the depth of CNNS, the residual architecture proposed by He et al. [7] is modified and applied into image SR. Compared with tasks like classification, detection and segmentation, however, the quality of reconstructed HR images is more relative to low-level features extracted by shallow layers instead of high-level features extracted by deep layers. Although some recent methods [8, 9, 10, 11] find that combining low-level features with high-level features is of critical importance to help reconstruct better HR images, most proposed networks set static skip connections between different blocks or layers, which means some feature map will be added or concatenated to another no matter what the input is. As a result, this kind of skip connection may constrain the ability of representation to some extent. In additional, most CNN-based methods treat features from different channels equally and this may affect the information flow of low and high frequency. In terms of image SR, we need to reconstruct as more high-frequency details from LR image as possible. So, we want SR network pays more attention to the recovery of high-frequency information instead of the delivery of low-frequency information. Besides, many researches usually place the upscaling layer at the end to build a deeper network, reduce the number of parameters and faster the inference speed. However, they cannot generate multi-scale predictions in one inference with one trained model.

To solve these problems, we first draw lessons from Laplacian pyramid super-resolution structure proposed by Lai et al. [12] and follow a similar coarse-to-fine SR strategy. That is to say, feature maps are progressively upscaled to the desired size in more than one step. Second, we introduce residual channel attention into each basic block to explore the most important features from different channels for image SR. Third, self-adaptive weighted skip connections (SAWSC), a more flexible way of residual connection, is proposed, with which we hope to better promote the effective flow of information between different levels of features. Specifically, a feature map will be multiplied by a weight produced by self-attention mechanism and then added to another feature map through skip connections.

We evaluate our proposed method on five widely-used benchmark datasets and the results show that our structure achieves great performance.

II. RELATED WORKS

A. CNN-based Image SR

In recent years, more and more CNN-based researches have proposed state-of-the-art methods since the pioneer work SRCNN [1] introduce the convolutional network into SISR. Although SRCNN consists of only three convolutional layers, it still outperforms conventional methods [13, 14, 15].

978-1-7281-8138-7/20/$31.00 ©2020 IEEE
DOI 10.1109/ICCST50977.2020.00043

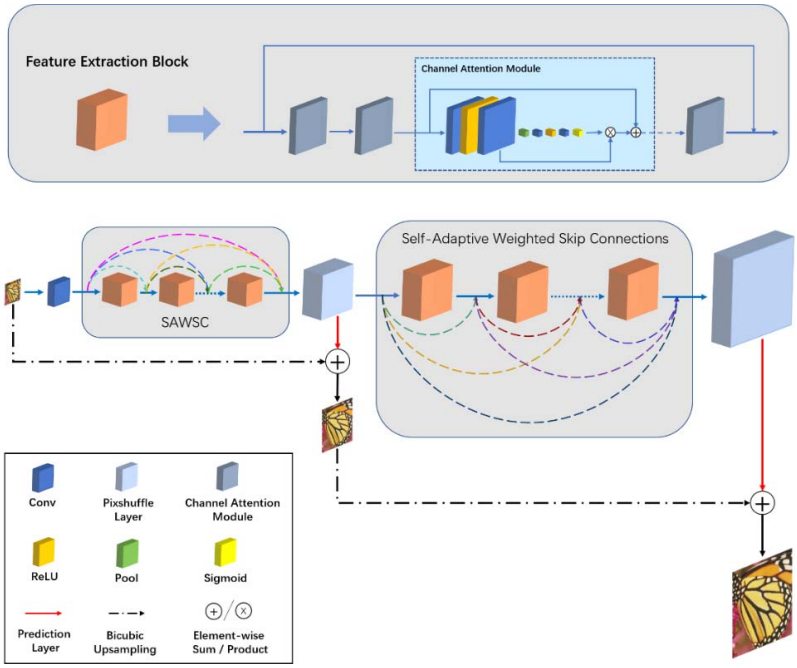


Figure 1. The architecture of our proposed network. Dotted lines with different colors means skip connections with different weights.

FSRCNN [3], the improved version of SRCNN, increases the depth of network and adopt deconvolutional layer, making it possible to input the LR image as its original size. Similar to other tasks, the residual structure and its modified version is also introduced into image SR. VDSR [2] cascades 20 layers and add the interpolated input LR image to the output.

As the concept of sub-pixel is introduced in ESPCN [16], many methods start to feed LR image directly into the network and add the sub-pixel layer at the end to reduce the cost of inference time. Meanwhile, the reduction of number of parameters also makes it possible to design a deeper network. DRRN [17] uses one recursive block and twenty-five residual blocks to build a fifty-two-layer deep CNN. Lim et al. [5] remove all batch normalization layers and ReLU activation layers outside the residual blocks of SRResNet [18]. They further propose the EDSR for single-scale image SR and MDSR for multi-scale image SR. In their study, the depth of the SR network is extended to 32 while the width is extended to 256 and the model capacity is largely increased in this way. Zhang et al. [19] use the basic channel attention layer to establish RCAN, an extremely deep network, and achieve significant improvement.

There are some methods try to make better use of low-level features from shallow layers besides increasing the model capacity. Tone el al. [20] propose the SRDenseNet based on DenseNet [11], which concatenates feature maps from different layers with skip connections. Further, MSRN [6] let features extracted by convolution layers of different kernel size interact with each other and RDN [8] introduces residual structure into dense skip connection. Besides, SAN by Dai et al. [10] introduces non-local network, another kind of self-attention mechanism, into image SR and uses the shared-source structure to make full use of the information from the original LR image and explore the intrinsic correlation of different features.

To improve the visual quality, generative adversarial networks is used in SRGAN [18] and ESRGAN [21], although this may cause unnatural artifacts.

B. *Skip Connections*

As the depth of CNNs continually grows, the problem of vanishing gradient makes the training become harder. ResNet [7] tries to solve this problem by bypassing features. Many researches have proved that this kind of structure could achieve better performance because it makes the design of deeper networks possible. Residual scaling, meaning that the residual blocks are multiplied by a decimal after convolution processing before adding them, is proposed by Szegedy et al. [22] and introduced to EDSR and RCAN. The results indicate that this trick could stabilize the training process. Huang et al. propose DenseNet to alleviate the problem of vanishing gradient, encourage the reuse of features, and greatly reduces the number of parameters. DenseNet allows each layer to get additional input from all previous layers and passes its own feature map to all subsequent layers. Different from ResNet, DenseNet replaces the operation of sum with concatenation in terms of the method of skip connections.

III. PROPOSED METHOD

A. *Network Architecture*

In our proposed method, the reconstruction of SR images follows a coarse-to-fine strategy. As shown in Fig. 1, the framework of our SR network, similar to LapSRN, is based

on the Laplacian pyramid. To be specific, the original LR image is directly fed into the network, and after a sequence of complex operations, a reconstructed HR image will be generated. During the process of inference, feature maps are upscaled progressively instead of in one step. Assuming the upscale factor is f, then the number of steps s is calculated as $log_2 f$. For example, the network is divided into 2 upscale levels and at each level, a sub-network is embedded if the scale factor is 4. At different upscale level, sub-networks share the similar structure but different heights and widths of feature maps.

Let's denote the input LR image as I_{LR} and the output SR image as I_{SR}^{f} if scale factor is f. One convolution layer is used to extract shallow features

$$F_0 = H_0(I_{LR}) \tag{1}$$

where $H_0(\cdot)$ denotes convolution operation of the first convolution layer. Then the F_0 is fed into following sub-networks.

At upscale level i, feature maps from level $i-1$ are upscaled with a factor of 2 and then input to the level $i+1$ for further operations. This process can be denoted as

$$F_i = H_i(F_{i-1}) \tag{2}$$

where F_{i} is the output of the i-th sub-network and $H_i(\cdot)$ denotes the operation of sub-network at level i, including the feature extraction, self-adaptive weighted skip connections and sub-pixel convolution upscaling. Further, in terms of the scale level s, we can have

$$F_s = H_s(H_{s-1}(\cdots H_2(H_1(F_0))\cdots)) \tag{3}$$

where F_s denotes the output of the s-th sub-network, as well as the end of feature extraction part.

In the prediction branch, at the end of each upscale sub-network, the last feature maps are input into a prediction layer, which consists of several cascaded convolution layers. The output of the prediction layer is a residual and it is summed with the upsampled LR image, predicted in the previous level, using the bicubic kernel. So, we can get all intermediate predictions with the scale factor of $f, f = 2,3\cdots 2^s$. And the prediction at level s can be represented as

$$I_{SR}^{s} = P_s(H_s(F_{\mathrm{s-1}})) + U_s(I_{SR}^{s-1}) \tag{4}$$

where $P_s(\cdot)$ denotes prediction operation and $U_s(\cdot)$ denotes upsampling with bicubic interpolation.

Our proposed network is optimized with multi-scale L_1 loss function. At different level, we downscale the HR image to specific size as the ground truth. So, we have total loss

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{s}\left\|I_{SR}^{n,i} - D_i(I_{HR}^{n})\right\|_1 \tag{5}$$

where θ denotes the parameter needed to be updated of our network, $D_i(\cdot)$ denotes downscale operation using bicubic kernel and N denotes the number of image pairs in one iteration. Note that $I_{SR}^{n,i}$ and I_{HR}^{n} means predicted SR image at level i and downscaled HR image which has the same size as $I_{SR}^{n,i}$ respectively in terms of the n-th training sample in each batch.

B. Feature Extraction Block (FEB)

Here we give more details about feature extraction blocks, basic blocks making up a single sub-network, whose structure is shown in Fig. 2.

Supposing there are m feature extraction blocks at each level. In each feature extraction block, we connect b channel attention modules in series and set a long skip connection from block's input directly to its output.

$$F_i^m = H_i^{m,b}\left(\cdots H_i^{m,1}(F_i^{m-1})\cdots\right) + F_i^{m-1} \tag{6}$$

where $H_i^{m,b}(\cdot)$ denotes the operation of the b-th channel attention module in the m-th feature extraction block at level i and F_i^m denotes features from the m-th feature extraction block at level i.

Our channel attention module is a modified version based on residual channel attention block [12]. Here, in order to speed up the inference and reduce the number of parameters of the network, we delete the channel reduction process. The operation of our channel attention module $H_i^{m,b}(\cdot)$ is concretized below from equation 7 to equation 9.

$$H_i^{m,b}(x) = w \cdot z + x \tag{7}$$

$$w = \delta\left(C_4\left(R_2\left(C_3\left(P(z)\right)\right)\right)\right) \tag{8}$$

$$z = C_2\left(R_1\left(C_1(x)\right)\right) \tag{9}$$

where $C(\cdot)$, $R(\cdot)$, $P(\cdot)$, $\delta(\cdot)$ denotes operations of convolution, ReLU activation, global pooling and sigmoid activation respectively. Note that x is used here to represent the input of the channel attention module.

C. Self-Adaptive Weighted Skip Connections (SAWSC)

Self-adaptive weighted skip connections (SAWSC) is our proposed structure, with which residuals are rescaled by adaptive factor ranging from 0 to 1. We use colored dotted arrows above in Fig. 1 to simply represent this kind of skip connections and then we present how it works concretely.

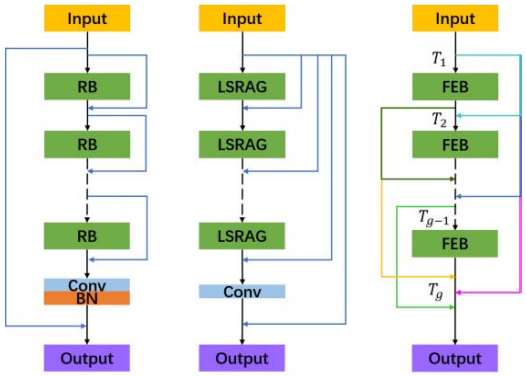
For demonstration convenience, we compare SAWSC with other kinds of skip connections in prior SR networks in Fig. 2. In this figure, (a) (b) (c) presents long and short skip connections used in SRResNet, share-source skip connections proposed in SAN and our SAWSC. Note that in Fig. 2, we use blue arrows in (a) and (b) to denote residual addition with a constant rescaling factor, which can be recognized as a weight from 0 to 1. However, arrows with different colors in our SAWSC, as shown in (c), means that different weight is put on different skip connections. The output of SAWSC can be iterated with the equation 10.

$$T_m = \sum_{j=1}^{m-1}\alpha_j T_j + FEB(T_{m-1}) \tag{10}$$

where $FEB(\cdot)$ denotes the operation of a feature extraction block, and its concrete function is given above. T_i is used to represent aggregated features after the i-th FEB and α_j is a rescaling factor.

In the limit case, when the weight is 0, it is equivalent to that there is no skip connection between two blocks. Hence, proposed SAWSC is a particularly flexible structure that has dynamic skip connections so that it can make the full use of features from different levels.

Therefore, the core of SAWSC lies in how to produce the rescaling factors of residuals. In our proposed method, SAWSC is based on self-attention mechanism and its operation is shown in Fig. 3. A rescaling factor is generated



(a) SRResNet (b) SAN (c) SAWSC (Ours)

Figure 2. Comparison of different skip connections

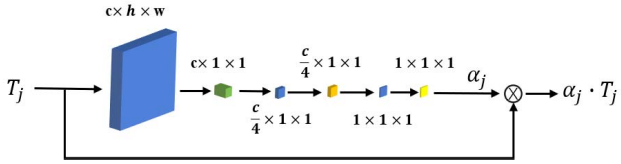


Figure 3. Self-Adaptive Weighted Skip Connections

according to the input features. Note that SAWSCs do not share weights which means that there are $\frac{m\cdot(1+m)}{2}$ SAWSCs given that the number of FEBs is m.

D. Implementation Details

The implementation details of our proposed network are given here. We set the number of levels as $s = 2$ and the scale factor f is 4. At each level, there are $m = 5$ FEBs and in each FEB, the number of channel attention modules is set as $b = 5$. So, the number of SAWSCs in one sub-network can be calculated as $\frac{m\cdot(1+m)}{2} = 15$.

The kernel size of all convolution layers is set 3×3 except for that after global pooling. Instead, those convolution layers' kernel size is 1×1. We also take a zero-padding strategy to maintain features' size constant at the same level. Most convolution layers have C_b filters. However, each prediction layer is made up with two cascaded convolution layers and the first one has $\frac{C_b}{2}$ filters whereas the second one has 3 filters. Note that because we use sub-pixel convolution to upscale feature maps, the channel size of feature maps is increased to $4 \cdot C_b$ and then reduced back to C_b. In our experiments, we train two versions of our proposed network with C_b set to 64 and 128 respectively.

IV. Experiment Results

A. Datasets and Metrics

Considering the small number of some conventional datasets, recent researches mainly use DIV2K dataset [25] as raining set, which consists of 800 high-quality images with 2K resolution. To be specific, we crop non-overlapping patches with the size 256×256 from these 800 training images as HR images. And they are degraded with the bicubic model to simulate LR counterparts. We additionally randomly augment the training patches by flipping both vertically and horizontally, rotating 90° and resizing the size to half or quarter.

Our proposed method is evaluated on public benchmarks including Set5 [24], Set14 [25], B100 [26], Urban100 [27] and Manga109 [28]. The same as other researches, PSNR and SSIM are used on Y channel of YCbCr color space.

B. Training Settings

Our network is optimized using Adam [29] with the initial learning rate set to 0.0001. After 100 epochs, the learning rate is decreased to one tenth of its initial set.

We use PyTorch as our framework to perform experiments with a GeForce RTX 2080 Ti GPU. The training stops after 200 epochs.

C. Comparison with Some Other Methods

We compare our proposed network with 8 other SISR methods: Bicubic, A+ [30], SRCNN [1], VDSR [2], DRCN [4], LapSRN [12], SRDenseNet [20] and CARN [9]. Table. 1 quantitatively shows the average PSNR and SSIM values on five different benchmark datasets. We use SAWSC (·) to represent our proposed method and results of all other methods are provided by their corresponding articles or public codes. Specifically, SAWSC (64) denotes our proposed network with the number of block channels set as $C_b = 64$. Similarly, C_b is 128 in SAWSC (128). Red text indicates the best score and blue text indicates the second best.

It can be found that our algorithm outperforms other listed methods for scale factor × 4, especially in Manga109 dataset. For the scale factor of 2, our network consists of only one sub-network, meaning less parameters. And we do not train the model of scale × 2 independently. Instead, the first sub-network trained for scale × 4 is used directly. Even so, our model still achieves a comparable performance to other methods in the case that the number of filters is set to 64. But when the number of filters increases to 128, our method outperforms most of others. Our method is at least 0.1 dB higher than them on average PSNR even in the case that Manga109 is excluded. It means that results generated by our model are more realistic. So, our method proves to be able to generate multi-scale SR images with relatively high quality and generalization.

To further present the visual performance of our method, we show some comparisons on Urban100 and Manga109 (mainly containing urban scenes, natural scenes and Japanese manga) with a scale factor of 4 in Fig. 4 and Fig. 5. We observe that SR images generated by our methods have richer details and shaper edges. Although the degradation distorts the window frame, the reconstructed image by our method is much clearer. Also, the super-resolution on faces is usually more difficult because of the rich details. However, our method is proved to be successfully applied into the anime face super-resolution and can generate more natural fake images. This indicates that our network has better visual results.

TABLE I. Quantitative results of some SR algorithms: average PSNR/SSIM for scale factor × 2 and × 4.

Algorithm	Scale	Set5	Set14	BSDS100	Urban100	Manga109
Bicubic	2	33.65 / 0.930	30.34 / 0.870	29.56 / 0.844	26.88 / 0.841	30.84 / 0.935
A+	2	36.54 / 0.954	32.40 / 0.906	31.22 / 0.887	29.23 / 0.894	35.33 / 0.967
SRCNN	2	36.65 / 0.954	32.29 / 0.903	31.36 / 0.888	29.52 / 0.895	35.72 / 0.968
VDSR	2	37.53 / 0.958	33.05 / 0.913	31.90/0.896	30.77/0.914	37.16 / 0.974
DRCN	2	37.63/ 0.959	32.98 / 0.913	31.85 / 0.894	30.76 / 0.913	37.57/ 0.973
LapSRN	2	37.52 / 0.959	33.08 / 0.913	31.80 / 0.895	30.41 / 0.910	37.27 / 0.974
CARN	2	**37.76 / 0.960**	33.52 / 0.920	**32.09 / 0.898**	**31.92 / 0.926**	—
SAWSC (64)	2	37.76 / 0.959	**33.64 / 0.921**	32.06 / 0.897	31.87 / 0.925	**39.30 / 0.980**
SAWSC (128)	2	**37.83 / 0.959**	**33.74 / 0.921**	**32.14 / 0.898**	**32.10 / 0.928**	**39.56 / 0.981**
Bicubic	4	28.42 / 0.810	26.10 / 0.704	25.96 / 0.669	23.15 / 0.659	24.92 / 0.789
A+	4	30.30 / 0.859	27.43 / 0.752	26.82 / 0.710	24.34 / 0.720	27.02 / 0.850
SRCNN	4	30.49 / 0.862	27.61 / 0.754	26.91 / 0.712	24.53 / 0.724	27.66 / 0.858
VDSR	4	31.35 / 0.882	28.03 / 0.770	27.29 / 0.726	25.18 / 0.753	28.82 / 0.886
DRCN	4	31.53 / 0.884	28.04 / 0.770	27.24 / 0.724	25.14 / 0.752	28.97 / 0.886
LapSRN	4	31.54 / 0.885	28.19 / 0.772	27.32 / 0.728	25.21 / 0.756	29.09 / 0.890
SRDenseNet	4	32.02 / 0.893	28.50 / 0.778	27.57 / 0.734	26.05 / 0.782	—
CARN	4	32.13 / 0.894	28.60 / 0.781	27.58 / 0.735	26.07 / 0.784	—
SAWSC (64)	4	**32.23 / 0.894**	**28.63 / 0.796**	**27.62 / 0.726**	**26.39 / 0.796**	**31.33 / 0.920**
SAWSC (128)	4	**32.45 / 0.897**	**28.74 / 0.799**	**27.75 / 0.743**	**26.61 / 0.803**	**31.59 / 0.923**

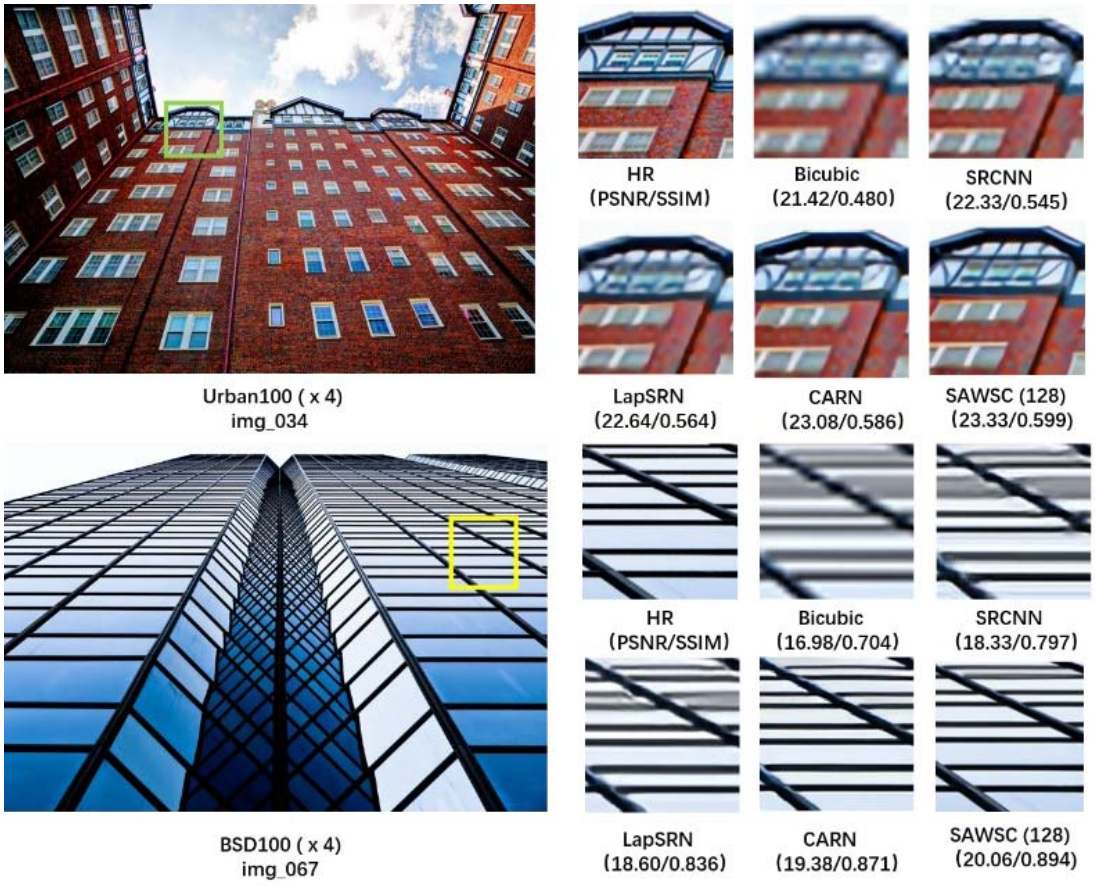


Figure 4. Visual comparison for scale × 4 on Urban100

Figure 5. Visual comparison for scale × 4 on Manga109

V. CONCLUSIONS

In this paper, we propose the self-adaptive weighted skip-connections structure for image SR. Further, based on which, a SR network following a coarse-to-fine strategy is proposed. It combines features self-adaptively and makes full use of them. The experimental results indicate that our method achieves better quantitative and visual performance than some other algorithms. This structure is efficient and has potential in other computer vision tasks.

VI. REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, " Learning a deep convolutional network for image super-resolution", in *Proc. Eur. Conf. Comput. Vis. (ECCV),* 2014, pp. 184-199.

[2] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super resolution using very deep convolutional networks", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1646-1654.

[3] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super resolution convolutional neural network", in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 391-407.

[4] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1637-1645.

[5] B. Lim, S. Son, H. Kim, S. Nah, and K. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017, pp. 136-144.

[6] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale Residual Network for Image Super-Resolution", in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 517-532.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770-778.

[8] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual Dense Network for Image Super-Resolution", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2472-2481.

[9] N. Ahn, B. Kang, and K. Sohn, Fast, accurate, and Lightweight Super-Resolution with Cascading Residual Network", in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 252-268.

[10] T. Dai, J. Cai, Y. Zhang, S. Xia, and L. Zhang, "Second-Order Attention Network for Single Image Super-Resolution", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 11065-11074.

[11] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4700-4708.

[12] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 624-632.

[13] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image", in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 349-356.

[14] K.I. Kim, and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior", *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 6, 2010, pp. 1127-1133.

[15] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 1059-1066.

[16] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1874-1883.

[17] Y. Tai, J. Yang, and X. Liu, "Image Super-Resolution via Deep Recursive Residual Network", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 3147-3155.

[18] C. Ledig, L. Theis, F. Huszár, and J. Caballero, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4681-4690.

[19] Y. Zhang, K. Li, K. Li, and L. Wang, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 286-301.

[20] T. Tone, G. Li, X. Liu, and Q. Gao, "Image Super-Resolution Using Dense Skip Connections", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4799-4807, 2017.

[21] X. Wang, K. Yu, S. Wu, J Gu, Y. Liu, C. Dong, C. Loy, Y. Qiao, and X. Tang, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks", in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 0-0.

[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inceptionv4, inception-resnet and the impact of residual connections on learning", *in Thirty-first AAAI conference on artificial intelligence*, 2017.

[23] R. Timofte, E. Agustsson, L. Van Gool, M.H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, et al, "Ntire 2017 challenge on single image super-resolution: Methods and results", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017, pp. 114-125.

[24] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding", in *British Machine Vision Conference*, 2012.

[25] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations", in *Proc. Int. Conf. Curves Surf.*, 2010, pp. 711-730.

[26] 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. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2001, pp. 416-423.

[27] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super resolution from transformed self-exemplars", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 5197-5206.

[28] 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 Application,* vol. 76, no.20, 2017, pp. 21811-21838.

[29] D. Kingma and J. Ba. Adam, "A method for stochastic optimization", 2014, *arXiv:1412.6980*. [Online]. Avaliable: https://arxiv.org/abs/1412.6980.

[30] R. Timofte, V. De Smet , and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution", in *Asian conference on computer vision (ACCV)*, 2014, pp. 111-126.