An Effective Single-Image Super-Resolution Model Using Squeeze-and-Excitation Networks

Kangfu Mei[1], Juncheng Li[2], Luyao[1], Mingwen Wang[1], Aiwen Jiang[1*]
Jiangxi Normal University[1]
East China Normal University[2]
{kfm, jcli, luyao}@clfstudio.com, {mwwang, jiangaiwen}@jxnu.edu.cn

***Abstract*—Recent works on single image super-resolution are concentrated on improving performance through enhancing spatial encoding between convolutional layers. In this paper, we focus on modeling the correlations between channels of convolutional features. We present an effective deep residual network based on squeeze-and-excitation blocks (called SEBlock) to reconstruct high-resolution image (HR) from low-resolution image (LR). SEBlock is used to adaptively recalibrate channel-wise feature mappings. Further, short connections between each SEBlock are used to remedy information loss. We evaluate our proposed method on several popular benchmarks. The experiment results show that our method can achieve state-of-the-art performance and get finer texture details through the test in some datasets. It outperforms several currently popular methods in most cases with less depth network and in a more flexible way, especially in case of high upscale rate.**

I. Introduction

Single-image super-resolution(SISR) is a popular computer vision problem, which aims at reconstructing a high resolution (HR) image from low-resolution(LR) image. Because of high-level information loss during image downsampling, SISR is an ill-posed inverse problem. There have many algorithms been proposed to try to solve it.

Early methods [1], [2], [3], [4], [5], besides bicubic and bilinear interpolation, learned direct mapping from LR to HR pairs by sacrificing certain accuracy or speed. improvement.Dong et al. [6] proposed the Super-Resolution Convolutional Neural Network (SRCNN) in 2014, which is the first successful model adopting CNN structure to solve SISR problem and obtaining great performance improvement. In SRCNN, convolutional neural network was used to learn non-linear mapping from each LR vector to a set of HR vector.

Because of the novel effect SRCNN had gotten, several deeper and more complicated models were proposed to extend it. Kim el at. [7] proposed a network called VDSR, even it achieves high performance in accuracy, its speed remain slow speed as it use a very deep residual convolutional network and an upscale image preprocess.

To avoid the complexities of feature extraction network and upscale preprocess, Shi el at. [8] replaced upscale preprocess with sub-pixel convolution layers. The sub-pixel layers could produce HR image from feature maps directly with a set of up-scaling filters. This architecture greatly improved the speed of networks. Therefore, following the strategy of up-sampling layer, Ledig el at. [9] further proposed the SRResNet with a very deep ResNet [10] architecture. Lai el at. [11] proposed the LapSRN, which use learned kernel as up-sampling unit to direct produced SR images.

In spite of great success achieved in the above architectures, the main issue that how to better model mapping from LR to HR image in a fast and flexible way remained unsolved.

In this paper, we proposed a Super-Resolution Squeeze-and-Excitation Network (SrSENet) for SISR. The concept of SEBlock [12] is employed to better modeling interdependencies between channels. Short connections from input to each SEBlock are used to remedy information lost. And different deconvolution layers are used for different scales under the same feature extraction architecture. The proposed method is evaluated on some popular publicly available benchmarks. The experiment results show that our proposed network can achieve competitive accuracy in a more accurate and flexible way. It can greatly reduce models complexity by using less layers and allow designing more flexible applications.

The contributions of this paper are two folds:

- we have introduced an effective super-resolution network with SEBlock. The proposed network can be efficiently trained. It performs dynamic channel-wise feature recalibration to provide a new powerful architecture to improve the representational ability of information extraction part from low-resolution images.
- We have set up a new state-of-the-art super-resolution method with fast running speed and accurate result in the measurement of PSNR and SSIM, without increasing networks complexity, especially in case of large upscale rate.

II. Related Work

A. Single-Image Super-Resolution

A typical network for single-image super-resolution could be approximately divided into two parts. The first part could be seen as a feature extraction block, which is composed of many stacked convolutional layers. The second part recorded up-scaling information from LR images to HR images. Recent works on SISR primarily concentrated on improving the first part by changing the way of skip connections between inputs of each layer. In other words, focus on changing the proportion of information captured by initial layers.

* Corresponding author

We group mainstream SISR models into four categories, as shown in Figuree 1. The (a) category only contains feature extraction, such as network in [6]. The (b) category like [7] introduces short connection as residual-learning. The (c) category like [9], [13], [14] accepts input in each feature extraction layer. Our proposed model could be categorized into the last category (d). The difference from the other three categories is that each extraction layer block receives input before channel-wise modeling. In this way, network could better learn mapping between LR-HR images.

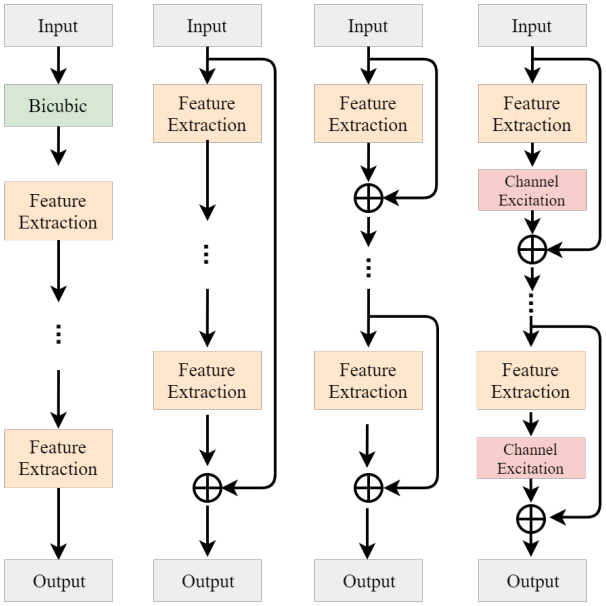

Fig. 1. Brief comparison for four SISR architecture category.

B. Squeeze-and-Excitation Channel

Even works on enhancing spatial encoding had got gread result, SENet [12] was proposed to fully capture channel-wise dependencies through adaptive recalibration. The SENet was separated into two steps, squeeze and excitation, to explicitly model channel interdependencies.

After initial images were input into the first convolution layer, the output feature $U \in \mathbb{R}^{W \times H \times C}$ will be passed to an architectural named SEBlock to do squeeze and excitation operator. The first squeeze operator were used to embeds the information from gloabl receptive field into a channel descriptor in each layer. Then sigmoid activation function and FC layer were later used to gain nonlinear interaction between each layers. The squeeze operator produces a sequence S in $1 \times 1 \times C$ which represented the correlations of each layer. The excitation operator later used to perform feature recalibration through reweighting the original feature mappings:

$$\tilde{U} = F_{scale}(U, S) = u_c \times s_c$$

Where u_c refers to the parameters of the c-th filter and s_c refers to the element of c-th channel descriptor. This architecture can help feature extaction parts better caputre the information from input to output. In our work, we combined SEBlock with ResNet for feature extraction. Without deeper network used, it can greatly decrease the complexity of our model.

C. Transposed Convolutional Layer

In order to obtain super-resolution image, a simple idea is to upscale original image first, then final HR image is directly generated from the resulted scaled image. It is not difficult to find that this kind of method wastes most time on preprocessing without any obvious advantage.

Shi el at. [8] first proposed to use sub-pixel convolution layer to produced HR images directly. It upscale a LR image by periodic shuffling the elements of a $W \times H \times C \cdot r^2$ tensor to a tensor of shape $rH \times rW \times rC$, but it didn't make full use of the correspondence information from LR to HR.

LapSRN [11] proposed by Lai el al. used a multiple transposed convolutional layer to do different upscale rate in a progressive way. Without any preprocessing step like upscale, LapSRN networks got more accurate information between LR and HR in fast way.

We use transposed convolutional layer with different parameters for different upscale rate, which can keep network simple and improve the power of networks to record reconstruction information.

III. Proposed Method

The proposed method aims at extracting information from LR images I_L and learn mapping function F from I_L feature maps to HR images I_H. We describe I_L with C channels in size of $W \times H$. With upscale rate r, I_H is in size of $rW \times rH$. Our ultimate goal is to minimize the loss between the reconstructed images and the corresponding ground truth HR images. In the following, we will describe the details of the proposed method.

A. Network Architecture

Our proposed method is inspired from SRResNet [9] and LapSRN [11]. Following LapSRN, our model contains two parts: residual learning stage and image reconstruction stage, as shown in Figure 2

Unlike SRResNet and LapSRN, in the residual learning stage, we introduced SrSEBlock to extract features from LR images. The SrSEBlock structure integrates ResNet and SENet, which can better capture information from inputs and better modeling interdependencies between channels.

As VDSR [7] suggested, in the SR ill-posed problem, surrounding pixels were useful to correctly infer center pixel. With larger receptive field a SR model has, it could use more contextual information from LR to better learn correspondences from LR to HR. In our proposed network, the filters of SrSEBlock is in size of $3 \times 3 \times 64$. Therefore, in case of depth D layer, its receptive field could be seen as $(2 \times D + 1) \times (2 \times D + 1)$ in the original image space. The

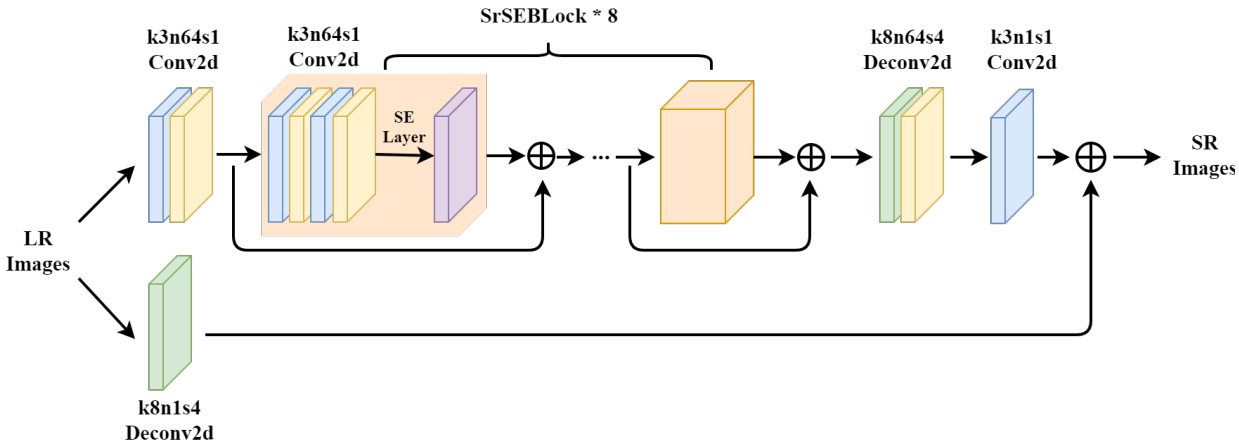

Fig. 2. Our proposed Network architectures of SrSENet in upscale of 4x. Blue blocks represent a Convolutional layer. Yellow blocks represent a LeakRelu layer. Green Blocks represent a Transposed Convolutional layer.

bigger receptive field means our network can use more context to reconstruct images.

As we know, with the increase of network depth, gradient disappearance or explosion will occur during training and the high-frequency information will also disappear. So, we introduce a short connection between SrSEBlocks which can receive input information before channel-wise modeling.

In the proposed network, we employ 8 SrSEBlocks to generate a feature mapping, and then we employ a transposed layer to transform the resulted mapping directly into a residual image by applying a deconvolutional layer. On different upscale rates, we dont increase the number of deconvolution layers, just directly change parameters such as the kernel size, stride and padding steps, to obtain corresponding residual image. In image reconstruction stage, the up-sampled LR image feature mappings and the learned residual feature mappings are added together to reconstruct HR image. By using residual image learning, network converges efficiently. The final feature mapping is output directly as the SR image.

B. Channels Excitation in SrSEBlock

Different from recent work that focus on enhancing spatial encoding, we use SrSEBlock to model correlations between channels. In this section, we will describe how the SrSEBlock work in our network.

In details, feature maps are input into a SELayer as Figure 3 shows. The corresponding excitations to each channel are output to scale original feature map. Taking a feature maps U in size of $W \times H \times C$ as input, we first do a global average pooling to generate channel-wise statistics z in size of $1 \times 1 \times C$., as show in below:

$$z_c = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i,j)$$

.

In order to learn nonlinear interaction between each channels, we use two FC layers with non-linear activations to form a bottleneck, as done in He el at. [10]. This architecture

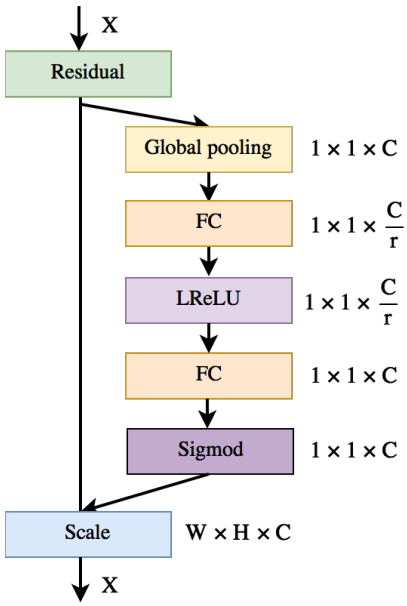

Fig. 3. The architecture of SELayer

could limit model complexity and benefit for generalization. The reduction ratio r at 16 is accepted to do dimensionality reduction. The final output s of SELayer is use to scale corresponding channels of residual feature mappings.

In this way, noise information in previous feature mappings could be reduced. And channels that contain useful information will be highly activated, helping to boost features discriminatitivities.

IV. Experiments

We compare our SrSENet with several state-of-the-art methods on popular benchmark datasets. The architecture of feature extraction part is kept the same in all cases, and the transposed convolutional layer size is changed according to different upscale rate.

Algorithm	Scale	Layer	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSDS100 PSNR / SSIM	Urban100 PSNR / SSIM	Manga109 PSNR / SSIM
Bicubic	2x	-	33.65 / 0.930	30.34 / 0.870	29.56 / 0.844	26.88 / 0.841	30.84 / 0.935
SelfExSR [15]	2x	-	36.49 / 0.954	32.44 / 0.906	31.18 / 0.886	29.54 / 0.897	35.78 / 0.968
SRCNN [6]	2x	3	36.65 / 0.954	32.29 / 0.903	31.36 / 0.888	29.52 / 0.895	35.72 / 0.968
FSRCNN [16]	2x	8	36.99 / 0.955	32.73 / 0.909	31.51 / 0.891	29.87 / 0.901	36.62 / 0.971
VDSR [7]	2x	20	37.53 / *0.958*	32.97 / **0.913**	**31.90** / **0.896**	**30.77** / *0.914*	37.16 / **0.974**
DRCN [17]	2x	5 (recursive)	**37.63** / **0.959**	32.98 / **0.913**	*31.85* / 0.894	*30.76* / 0.913	**37.57** / *0.973*
LapSRN [11]	2x	14	37.52 / **0.959**	*33.08* / **0.913**	31.80 / *0.895*	30.41 / 0.910	37.27 / **0.974**
SrSENet (ours)	2x	20	*37.56* / *0.958*	**33.14** / *0.911*	31.84 / **0.896**	30.73 / **0.917**	*37.43* / **0.974**
Bicubic	4x	-	28.42 / 0.810	26.10 / 0.704	25.96 / 0.669	23.15 / 0.659	24.92 / 0.789
SelfExSR [15]	4x	-	30.33 / 0.861	27.54 / 0.756	26.84 / 0.712	24.82 / 0.740	27.82 / 0.865
SRCNN [6]	4x	3	30.49 / 0.862	27.61 / 0.754	26.91 / 0.712	24.53 / 0.724	27.66 / 0.858
FSRCNN [16]	4x	8	30.71 / 0.865	27.70 / 0.756	26.97 / 0.714	24.61 / 0.727	27.89 / 0.859
VDSR [7]	4x	20	31.35 / 0.882	28.03 / *0.770*	*27.29* / *0.726*	*25.18* / 0.753	28.82 / 0.886
DRCN [7]	4x	5 (recursive)	*31.53* / *0.884*	28.04 / *0.770*	27.24 / 0.724	25.14 / 0.752	28.97 / 0.886
LapSRN [11]	4x	27	**31.54** / **0.885**	**28.19** / **0.772**	**27.32** / **0.728**	**25.21** / *0.756*	**29.09** / **0.890**
SrSENet (ours)	4x	20	31.40 / 0.881	*28.10* / 0.766	*27.29* / 0.720	**25.21** / **0.762**	*29.08* / *0.888*
Bicubic	8x	-	24.40 / 0.657	23.19 / 0.568	23.67 / 0.547	20.74 / 0.515	21.47 / 0.649
SelfExSR [15]	8x	-	25.52 / 0.704	24.02 / 0.603	24.18 / 0.568	*21.81* / *0.576*	22.99 / 0.718
SRCNN [6]	8x	3	25.33 / 0.689	23.85 / 0.593	24.13 / 0.565	21.29 / 0.543	22.37 / 0.682
FSRCNN [16]	8x	8	25.41 / 0.682	23.93 / 0.592	24.21 / 0.567	21.32 / 0.537	22.39 / 0.672
VDSR [7]	8x	20	25.72 / *0.711*	24.21 / *0.609*	24.37 / *0.576*	21.54 / 0.560	22.83 / 0.707
LapSRN [11]	8x	25	**26.14** / **0.738**	**24.44** / *0.623*	*24.54* / **0.586**	*21.81* / *0.581*	*23.39* / **0.735**
SrSENet (ours)	8x	20	*26.10* / 0.703	*24.38* / 0.586	**24.59** / 0.539	**21.88** / 0.571	**23.54** / *0.722*

TABLE I

QUANTITATIVE COMPARISONS OF STATE-OF-THE-ART METHODS. RED BOLD TEXT INDICATES THE BEST PERFORMANCE AND BLUE ITALICS TEXT INDICATES THE SECOND BEST PERFORMANCE. WE USE RESULTS FROM LAPSRN TO DO COMPARATION, AND ATTENTION THAT LAYERS IN THE TABLE INCLUDE CONVOLUTION AND DECONVOLUTION

In our experiment settings, given a set of HR images $\{Y_i\}$ and the corresponding down-sampled LR images $\{X_i\}$ through bicubic, our goal is to minimize the Charbonnier Penalty Function [18] defined as below, which is a differentiable variant of L_1 norm:

$$\rho(z) = \sqrt{z^2 + \varepsilon^2}$$

The loss is minimized using stochastic gradient descent with the standard backpropagation. We solve:

$$G^* = arg \min_G \frac{1}{n} \sum_{i=1}^{n} \rho(Y_i - G(X_i))$$

Where G represents our SR image networks.

A. Datasets for Training and Testing

Different from previous work, we use DIV2K [19] to train our model for more realistic modeling. DIV2K is a new high quality image datastes for image super resolution. Its training data has 800 high definition, high resolution images. In our experiments, we find different image processing framework will produce different bicubic downscale results. So for fair comparison, we all use the bicubic downsampling algorithm in Matlab image processing tool to generate LR-HR image pairs for our network training. For each pair, we crop HR sub image in 96×96 size and downscale it to LR images by different downscale factors. We export the pairs as MAT variable in HDF5 type.

We compare our proposed method with SRCNN [6], FSRCNN [16], SelfExSR [15], VDSR [7], DRCN [7] and LapSRN [11] on five common used benchmark datastes Set5[20], Set14[21], BSDS100[22], Urban100[23] and Manga109[24], we evaluate difference between the resulted SR and ground truth HR in PSNR and SSIM[25]. Our code is available on GitHub[1]

B. Training Details

We use 8 SrSEBlocks to do feature extraction, for each upscale super-reslution we use [4,2,1], [8,4,2], [16,8,4] for 2x, 4x, 8x rate up-scaled super-resolution respectively. Here in the format [*,*,*], the first represents kernel size, the second represents stride steps, and the last is padding size in transposed layer. About odd multiples of magnification, we can achieve an odd magnification by modifying the kernel size(the third parameter) of the convolutional network to an odd number.(e.g., [3,*,*]) During the training, we set the initial learning rate at 1e-4 and decrease after each 150 epochs. We use Adma optimizer [26] with $\beta_1 = 0.9$ to let network convergence and the training batches is 64. It roughly takes half day on a machine using 4 Titan X GPU for a single upscale training.

The performance comparisons are shown on Table I. From the experiment results, we can easily find that our proposed method obtains competitive performance in all datasets in different upscale rates. Especially in larger scale case, the advantages of our method are more obvious. Our method can achieve top performance with less network depth.

[1]Our code: https://github.com/MKFMIKU/SrSENet.git

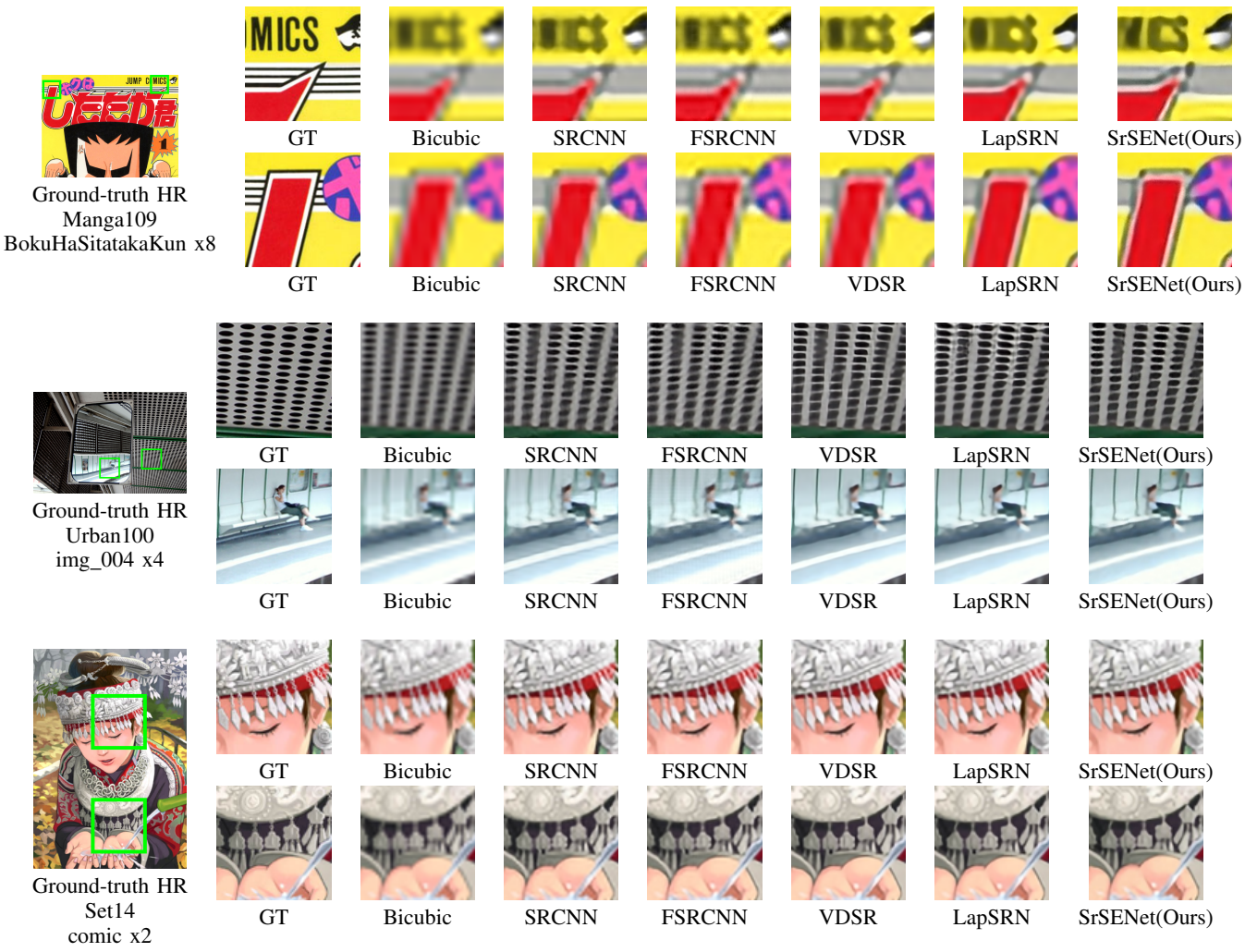

Fig. 4. Visual comparation on Bicubic, SRCNN, FSRCNN, VDSR, LapSRN and SrSENet. in upscale rate of 8x, 4x, 2x

In Figure 4, we further show some realistic results for visual comparison. We can find that the fine texture of Image in our method are recovered better.

V. Conclusions

In this paper, we have proposed a new effective super-resolution model by using a deep residual network with SrSE-Block. Our method focuses on modeling channels correlations between feature mappings from LR images. By modeling channel wise, we have confirmed that our method could produce more realistic texture on real world images. We set a new state-of-the-art super resolution method without increasing complexities of network. We believe that our approach could be applied to other real world computer vision problems.

Acknowledgment

This work was supported by National Natural Science Foundation of China under Grant Nos. 61365002 and 61462045.

References

[1] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[2] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 561–568.

[3] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[4] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[5] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199.

[7] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[8] 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*, 2016, pp. 1874–1883.

[9] 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," *arXiv preprint arXiv:1609.04802*, 2016.

[10] 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*, 2016, pp. 770–778.

[11] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[13] 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*, 2017.

[14] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[15] 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*, 2015, pp. 5197–5206.

[16] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision*. Springer, 2016, pp. 391–407.

[17] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.

[18] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[19] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[20] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[21] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[22] 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 *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.

[23] 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*, 2015, pp. 5197–5206.

[24] 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*, vol. 76, no. 20, pp. 21 811–21 838, 2017.

[25] 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*, vol. 13, no. 4, pp. 600–612, 2004.

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