2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)

HIERARCHICAL RECURSIVE NETWORK FOR SINGLE IMAGE SUPER RESOLUTION

Minglan Su[1], *Shenqi Lai*[2], *Zhenhua Chai*[2], *Xiaoming Wei*[2], *Yong Liu*[1],

[1]Beijing University of Posts and Telecommunications, [2]Vision and Image Center of Meituan

ABSTRACT

Super Resolution (SR) technique aims to reconstruct the high resolution (HR) image from the observed low resolution (LR) one, which is a significant application in our daily life. In this paper, we propose a novel structure named hierarchical recursive network (HRN), which consists of several sub networks and will reconstruct the HR progressively. In each sub network, the LR feature map will be used as input, the contextual information will be explored and the predicted residuals together with the transposed convolutional outputs will be fused to the finer one. Besides, our network can generate multi-scale HR images with a single model and thus is potentially useful in practical applications. Extensive experimental results show that our proposed method can achieve the state-of-the-art performance.

Index Terms— Single image super resolution, progressive reconstruction, hierarchical recursive network

1. INTRODUCTION

Single image super resolution (SISR) has wide applications in different fields, such as aerial imaging, medical image processing, texture analysis and biometrics recognition [1], which is an important research topic over the past two decades.

The core of SISR is to learn the linear or nonlinear mapping from the LR to HR, and lots of works have been proposed in the literature [2]. For instance, the traditional methods use low- and high-resolution exemplar pairs to learn the mapping function, while the deep learning based methods which can learn the mapping in an end to end fashion can exhibit even better performance. Super resolution convolutional neural network (SRCNN) [3] is one of the early work proposed in this way. However, SRCNN needs a preprocessing step bicubic interpolation before the input, which will bring some unnecessary computational cost. Then, Dong et al. [4] propose a fast version based on SRCNN, which uses the transposed convolution in the final layer. In this way, feature maps will be enlarged so that LR images can be directly fed to the network. The depth of the CNN will also affect the performance. Kim et al. [5] propose to increase number of the convolutional layers from 3 to 20 with residual structure and the proposed very deep super resolution (VDSR) network

Fig. 1: LR (left) and HR (right) reconstructed by the proposed method.

has made significant improvement over the original SRCNN. There are also other ways to further explore the network structure. For example, in order to control the parameter volume of the deeper network structure, recursive neural network (RNN) is proposed to use by Kim et al. [6]. In order to make use of the information from multiple levels, Zhang et al. [7] propose a novel block named residual dense network (RDN), which can extract richer local features via dense connected convolutional layers.

All the methods mentioned above can reconstruct HR images only in a single scale factor, which could limit its use in practice. Lai et al. [8] propose the laplacian pyramid super resolution network (LapSRN), which can progressively reconstruct HR images with different scaling factors. In this way, multiple HR images with several scales can be obtained within a single forward. Although LapSRN has brought great efficiency, there are still some issues needed to be addressed. Firstly, the feature extraction branch only consists of the vanilla convolutional layers, which could neglect the mining of the context information. Secondly, in image reconstruction branch the raw LR image can be further exploited in deeper cascades, while in LapSRN it is only explored in the first cascade. Finally, the performance improvement of LapSRN is relatively limited in comparison with other state-of-the-art SR methods. In order to solve these problems, in this paper we propose a novel hierarchical structure to better explore the context information. The finer texture can be better reconstructed, which is shown in Fig.1. The details will be elaborated in next section. In Section 3 the experiments will conducted, and the conclusion will be made in the Section 4.

978-1-5386-9214-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMEW.2019.00108

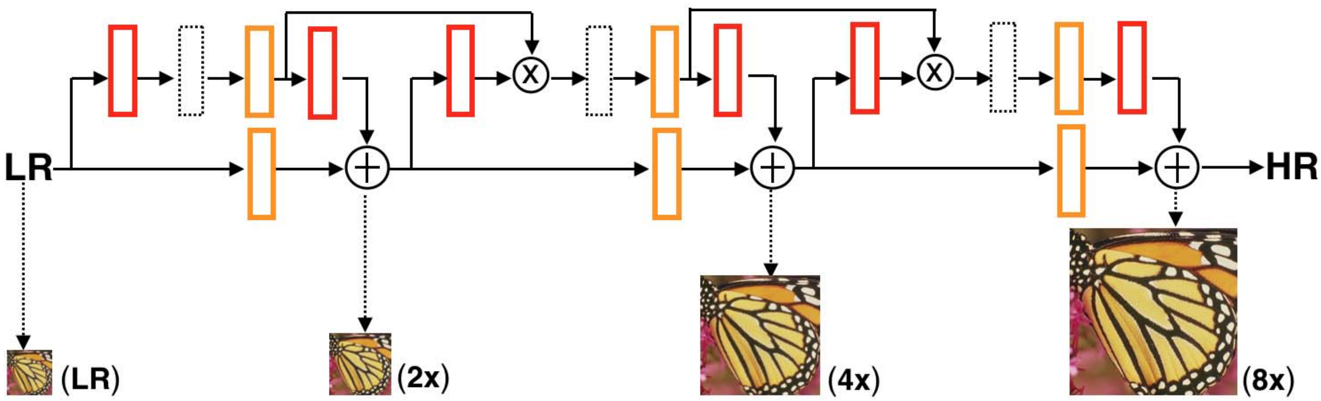


Fig. 2: The proposed architecture hierarchical recursive network (HRN). Red boxes indicate convolutions. Orange boxes indicate transposed convolutions (for upsampling). Black boxes indicate hierarchical recursive blocks, the details can be found in Fig.3. $\oplus$ means element-wise addition. $\otimes$ stands for element-wise multiplication.

2. PROPOSED METHOD

In order to inherit the advantages of lapSRN which can produce HR images in multiple scales in a single forward, we use an improved network structure by exploring the context information. The details can be found in the following.

2.1. Network Structure

As is shown in Fig.2, there are mainly two differences in comparison with the original lapSRN: (1) the design of the backbone network; (2) the connection way between two adjacent subnetworks.

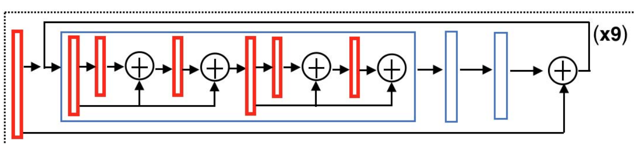


Fig. 3: The network structure of HRB.

The design of the backbone network: The context information could be very important to reconstruction of the texture details. In our design we propose to use the hierarchical residual block (HRB) in each sub network. In HRB (Fig.3), we use hierarchical convolutions instead of traditional convolution, which can be viewed as a special case of dense block with less parameters. In this way, context information from different scales will be utilized without increasing too much computation. In order to explore the non-linearity with a deeper network and at the same time keep a good balance with model size, we use recursive operator to share the convolution weights within the structure. We find that better performance can be achieved even under the same model size.

The connection way between two adjacent subnetworks: both residual features and the medium output image are useful for reconstruction in the finer stage, so in our design the attention module is used to get the fused input with element-wise multiplication in feature space. The medium output image from each subnetwork and the corresponding ground truth HR will be used for the computation in the final loss. In order to have a fair comparison with LapSRN, we use the same loss function, which is defined as follows:

$$\begin{aligned} L(\hat{\mathrm{y}}_s, \mathrm{y}; \theta) &= \frac{1}{N}\sum_{i=1}^{N}\sum_{s=1}^{L}\rho(\hat{y}_s{}^{(i)} - y_s{}^{(i)}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{s=1}^{L}\rho((\hat{y}_s{}^{(i)} - x_s{}^{(i)}) - r_s{}^{(i)}) \end{aligned} \quad (1)$$

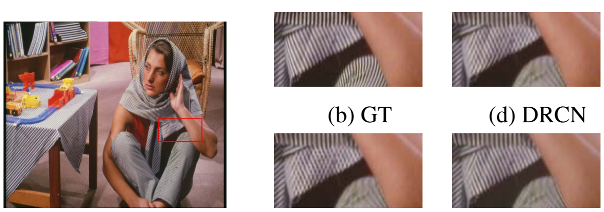
where $\rho(x) = \sqrt{x^2 + \xi^2}$ is the Charbonnier penalty function, N is the number of training samples in each batch, and L is the number of our sub-network. In subnetwork s, $\hat{y}_s$ and y_s are denoted as HR reconstruction result and ground truth respectively. The input and residual image are denoted as x_s and r_s.

2.2. Implementation details

In the proposed HRN, 3 x 3 is set as the filter kernel size and 64 filters are used for all convolutional layers except that in the fusion layers whose kernel size is 1 x 1. Besides, zero padding is used in order to keep the output size fixed, and the size of the transposed convolutional filters is 4 *x* 4. For the non-linear activation function, we use the ReLu in hierarchical block and in the rest layers the leaky rectified linear unit (LReLU) with a negative slope of 0.2 is used.

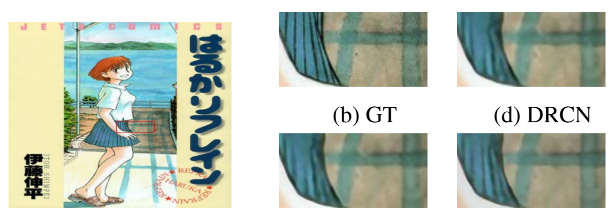
Table 1: Comparisons with the State-of-the-arts

Algorithm	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSDS100 PSNR/SSIM	Urban100 PSNR/SSIM	MANGA109 PSNR/SSIM
Bicubic	2	33.65 / 0.937	30.34 / 0.881	29.56 / 0.858	26.88 / 0.851	30.84 / 0.937
SRCNN [3]	2	36.65 / 0.954	32.29 / 0.903	31.36 / 0.888	29.52 / 0.895	35.72 / 0.968
FSRCNN [4]	2	36.99 / 0.955	32.73 / 0.909	31.51 / 0.891	29.87 / 0.901	36.62 / 0.971
SelfExSR [9]	2	36.49 / 0.954	32.44 /0.906	31.18 / 0.886	29.54 /0.897	35.78 / 0.968
SCN [10]	2	36.52 / 0.953	32.42 / 0.904	31.24 / 0.884	29.50 / 0.896	35.47 / 0.966
VDSR [5]	2	37.53 / 0.958	32.97 / 0.913	31.90 / 0.896	30.77 / 0.914	37.16 / 0.974
DRCN [6]	2	37.63 / 0.959	32.98 / 0.913	31.85 / 0.894	30.76 / 0.913	37.57 / 0.973
LapSRN [8]	2	37.52 / 0.959	33.08 / 0.913	31.80 / 0.895	30.41 / 0.910	37.27 / 0.974
Ours	2	37.81/ 0.968	33.36 / 0.927	32.08 /0.912	31.30 / 0.931	37.79 / 0.978
Bicubic	4	28.42 / 0.822	26.10 / 0.721	25.96 / 0.687	23.15 / 0.674	24.92 / 0.794
SRCNN [3]	4	30.49 / 0.862	27.61 / 0.754	26.91 / 0.712	24.53 / 0.724	27.66 / 0.858
FSRCNN [4]	4	30.71 / 0.865	27.70 / 0.756	26.97 / 0.714	24.61 / 0.727	27.89 / 0.859
SelfExSR [9]	4	30.33 / 0.861	27.54 / 0.756	26.84 / 0.712	24.82 / 0.740	27.82 / 0.865
SCN [10]	4	30.39 / 0.862	27.48 / 0.751	26.87 / 0.710	24.52 / 0.725	27.39 / 0.856
VDSR [5]	4	31.35 / 0.882	28.03 / 0.770	27.29 / 0.726	25.18 / 0.753	28.82 / 0.886
DRCN [6]	4	31.53 / 0.884	28.04 / 0.770	27.24 / 0.724	25.14 / 0.752	28.97 / 0.886
LapSRN [8]	4	31.54 / 0.885	28.19 / 0.772	27.32 / 0.728	25.21 / 0.756	29.09 / 0.890
Ours	4	31.70 / 0.899	28.46 / 0.801	27.43 / 0.752	25.45 / 0.781	29.39 / 0.902
Bicubic	8	24.39 / 0.647	23.19 / 0.561	23.67 / 0.542	20.74 / 0.509	21.47 / 0.636
SRCNN [3]	8	25.33 / 0.689	23.85 / 0.593	24.13 / 0.565	21.29 / 0.543	22.37 / 0.682
FSRCNN [4]	8	25.41 / 0.682	23.93 / 0.592	24.21 / 0.567	21.32 / 0.537	22.39 / 0.672
SelfExSR [9]	8	25.52 / 0.704	24.02 / 0.603	24.18 / 0.568	21.81 / 0.576	22.99 / 0.718
SCN [10]	8	25.59 / 0.705	24.11 / 0.605	24.30 / 0.573	21.52 / 0.559	22.68 / 0.700
VDSR [5]	8	25.72 / 0.711	24.21 / 0.609	24.37 / 0.576	21.54 / 0.560	22.83 / 0.707
LapSRN [8]	8	26.14 / 0.738	24.44 / 0.623	24.54 / 0.586	21.81 / 0.581	23.39 / 0.735
Ours	8	26.38 / 0.747	24.65 / 0.635	24.66 / 0.592	22.07 / 0.598	23.83 / 0.752



(a) GT (b) GT (c) LapSRN (d) DRCN (e) Ours

Fig. 4: Visual comparison for 2× SR on Set14.



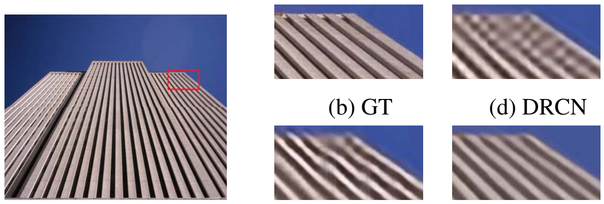
(a) GT (b) GT (c) LapSRN (d) DRCN (e) Ours

Fig. 5: Visual comparison for 4× SR on MANGA109.

3. EXPERIMENTS

3.1. Datasets and training details

We use the same training set as the other existing methods, which includes 91 images from [11] and 200 images from the Berkeley Segmentation Dataset [12]. We use pytorch and train our model with the ADAM solver. We set the size of mini-batch and weight decay to 24 and 0 respectively. We crop the HR into patches sized in 128 *x* 128, and we do data augmentation in three ways: (1) randomly downscaling between [0.5, 1.0]; (2) randomly image rotating by 90°, 180°, or 270°; (3) image flip horizontally with probability 0.5. We strictly follow the training protocol of existing methods and generate the LR training patches using the bicubic downsampling. We add warming-up strategy in the first 10 epochs, and the number of total epochs is set to 200. The learning rate is initialized with 3e - 4 for all layers and decreased by a factor of 10 for every 100 epochs. Training the HRN roughly takes 10 hours in average with a PASCAL V100 GPU.



(b) GT (d) DRCN

(a) GT (c) LapSRN (e) Ours

Fig. 6: Visual comparison for 8× SR on Urban100.

3.2. Comparison with the State-of-the-art Methods

More experiments are carried out on 5 popular datasets: Set5 [13], Set14 [14], BSDS100 [12], Urban100 [9], MANGA109 [15], which contain natural images, urban scenes and Japanese manga, and the detailed results can be found in Tab.1. All the methods are evaluated under commonly used quality metric (e.g. PSNR and SSIM), and our proposed method achieves the best performance. Some visual results for different upscaling factors are shown in Fig.4 (2x), Fig.5 (4x) and Fig.6 (8x) respectively. The details reconstructed by our proposal exhibit better than the rest methods especially in some texture areas.

4. CONCLUSIONS

In this paper, we have proposed a structure named hierarchical recursive network (HRN) to explore the contextual information for single image super resolution. The experimental results show the superiority of the proposed methods. Besides, our model size is only 1.2M and the algorithm can run in 60fps on a modern GPU without any explicit optimization. In our future work, we will further compress the model and transplant HRN to the mobile platform.

5. REFERENCES

[1] Kamal Nasrollahi and Thomas B. Moeslund, "Super-resolution: a comprehensive survey," *Mach. Vis. Appl.*, vol. 25, no. 6, pp. 1423–1468, 2014.

[2] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang, "Single-image super-resolution: A benchmark," pp. 372–386, 2014.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2016.

[4] Chao Dong, Chen Change Loy, and Xiaoou Tang, "Accelerating the super-resolution convolutional neural network," pp. 391–407, 2016.

[5] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," pp. 1646–1654, 2016.

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Deeply-recursive convolutional network for image super-resolution," pp. 1637–1645, 2016.

[7] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu, "Residual dense network for image restoration," *CoRR*, vol. abs/1812.10477, 2018.

[8] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," pp. 5835–5843, 2017.

[9] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single image super-resolution from transformed self-exemplars," pp. 5197–5206, 2015.

[10] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas S. Huang, "Deep networks for image super-resolution with sparse prior," pp. 370–378, 2015.

[11] Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[12] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[13] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," pp. 1–10, 2012.

[14] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," pp. 711–730, 2010.

[15] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21811–21838, 2017.