Single Image Super-Resolution Network Based on Multiple-Scale Self-Similarity Module

Jiaxin Xiong1,a, Guoyou Wang2,b*
1 School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
2 School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
*aemail: m201972575@hust.edu.cn
*bmail: gywang@mail.hust.edu.cn

Abstract. Recently, single image super-resolution (SISR) based on deep convolutional neural networks (CNNs) has been widely studied and achieved remarkable results. However, most of these methods mainly widen or deepen the network to get better results, ignoring the self-similarity in natural images. Some works have successfully leveraged this self-similarity by exploring non-local attention modules, but they only use non-local attention modules at one scale or two scales. And they need to calculate the self-similarity in each residual module, which makes the network more complex and takes a long time to train. We propose a multi-scale self-similarity module (MSSS) that can be flexibly inserted into existing various super-resolution networks. MSSS uses convolution with different kernels and the trailing dilated convolution to obtain feature maps at different scales, and then calculates the correlation between multiple feature maps at different scales through two-head region-level recurrent criss-cross attention modules (TRRCCA), finally gets fusion output through channel attention. So MSSS can mine multi-scale self-similarities. We only insert a MSSS module into the basic residual SISR network, and the result is significantly improved.

1. Introduction

Single image super-resolution (SISR) is the recovery of high-resolution (HR) images from a low-resolution (LR) image[1]. SISR is an important part of computer vision and image processing. SR has various applications in real life, such as in medical imaging, in security and surveillance images, etc. It can also help improve the performance of other computer vision tasks, such as small target detection. Traditional methods based on interpolation, statistics, and so on have been used for the super-resolution problem. With the development of deep learning techniques in recent years, more and more deep learning-based SR models have been proposed and have made great progress.

In 2005, SRCNN[2] was the first to use deep learning for SR networks. Compared with traditional methods, it has better results and faster inference speed. From then on, SISR based on deep learning has developed rapidly. SRResNet[3] first introduced the residual structure into SISR networks to obtain more high-level information in 2007. In the same year, EDSR[4] optimized SRResNet by removing batch normalization layers and activation layers. It also used residual scaling technique to stably train large models. In 2018 RDN[5] proposed the densely connected residual block, RCAN[6] used channel attention modules instead of residual scaling factors and proposed the RIR structure.
All of the above methods get better results by deepening the network through various connection or attention mechanisms, without using the image self-similarity. In 2019, SAN[7] proposed a non-local attention mechanism at the image block level to improve the SR results. In 2020, CSNLN[8] calculates the correlation between two different scale feature maps by downsampling the image. Both of them do not take full advantage of the self-similarity of images and seriously limit the results of SISR. Self-similarity of images is a strong property of natural images, meaning that small image patches will appear repeatedly in different scales of a single image.

To solve the above problems, we propose MSSS module, MSSS module obtains different scales feature maps by convolution with different kernels and the trailing dilated convolution, which is more consistent with the human visual system than simple downsampling. And MSSS module uses two-head region-level recurrent criss-cross attention instead of non-local attention, which is more friendly for GPU memory.

2. Related Works

2.1. Feature maps at different scales
Both San and CSNLN use bicubic downsampling to obtain different scale feature maps. In bicubic downsampling, the original image is smoothed by Gaussian kernel, and then all even rows and columns in the result are deleted to get 0.5 times of the original image. Analogously to the receptive field, each pixel in this result represents a 4×4 receptive field in the original image. Image pyramid is obtained by repeated downsampling operation. From the bottom to the top, the receptive field represented by each pixel in each layer changes from 4×4 to 16×16, 64×64 and so on, so as to get different scale images. So we know that we can get different scale feature maps by changing the size of receptive field. Multi-scale receptive field technology is often used in target detection. The most commonly used ones are the Inception family[9], ASPP[10], Deformable CNN[11], and RFB[12]. Inception block adopts multiple branches with different kernel sizes to capture multi-scale information, but all the kernels have the same center so loses some crucial details. ASPP uses dilated convolution to change the distance from the kernel to the center but treats the clues at all positions equally. Deformable CNN has the same drawbacks as ASPP. RFB highlights the relationship between receptive field size and eccentricity, where bigger weights are assigned to the positions nearer to the center by small kernels, claiming that they are more important than the farther ones. Therefore, MSSS adopts an RFB-like approach to obtain a multi-scale receptive field.

2.2. Non-local operation
The non-local operation in CNN is designed to allow the network to capture long-range dependencies. Non-local Neural Networks[13] first proposed non-local attention networks to allow neural networks to focus more on location with rich information by assigning weights to features at all locations in high-level vision tasks such as image recognition and target detection. Subsequently, because of the self-similarity of images, the non-local attention block was used in SR networks by SAN, CSNL, etc., and achieved significant improvement in SR results. Unlike the semantic-level features in high-level vision tasks that do not differ significantly across scales, the feature maps at lower scales in SISR tasks contain richer detail and texture information. Therefore, it is necessary to exploit multiple-scale self-similarity on the SISR network. However, non-local operation can put a lot of pressure on the GPU in the training of SISR. So both SAN and CSNLN divide the feature maps into small patches with non-local operation on each patch. In the meanwhile non-local operations at a proper neighborhood size are preferable for low-level tasks. So we propose two-head region-level recurrent criss-cross attention module (TRRCCA). TRRCCA also divides feature maps into small blocks, and implements recurrent criss-cross attention[14] (RCCA) instead of non-local attention on each patch. One criss-cross attention can capture contextual information in horizontal and vertical directions, RCCA by serially stacking two criss-cross attention modules can collect contextual information from all pixels. Because
TRRCCA uses both the method of dividing feature maps into blocks and RCCA, it greatly reduces the complexity in time and space.

3. Multiple-Scale Self-Similarity Network

In this section, we describe in detail the structure and the parts of MSSS module. And the validity of MSSS is demonstrated by inserting it into the SISR network that constitutes MSSSNet.

3.1. Multiple-Scale Self-Similarity module (MSSS)

The overall structure of MSSS is shown in Figure 1. The whole MSSS can be divided into three parts: the first part is a multi-branch convolutional layer with different kernel sizes followed by the corresponding dilated convolution. The second part is two-head region-level recurrent criss-cross attention module (TRRCCA). The three part is fusion layer.

In the first part, we draw lessons from RFB using convolution with different kernels and the corresponding dilated convolution. We obtain maps at different scales by changing the size of receptive field of each branch, which losses less image information and is more closer to human visual systems than the traditional bicubic downsampling. The receptive fields of each branch from left to right are 3×3, 9×9, 15×15 and 21×21 respectively. In the fusion layer, unlike RFB which directly contacts the outputs of each branch together and then passes a 1×1 convolution, we use channel attention to better fuse the outputs and focus on more important branches.

3.2. Two-head Region-level Recurrent Criss-Cross Attention module (TRRCCA)

In the second part, we use TRRCCA to perform non-local operation on feature maps from different branches at different scales from the first part. The output of the branch with 3×3 receptive field is selected as the basis to form a group of TRRCCA inputs with other branches.

Generic non-local operation in CNN as:

$$y_j = \frac{1}{C(x)} \sum_{j} f(x_i, x_j) g(x_j)$$

(1)

Here i is the index of an output position whose response is to be computed and j is the index that enumerates all possible positions. x is the input, y is the output. f computes the relationship between i and j. g computes a representation of the input y at the position j. y is normalized by C(x).

The equation (1) is for the single input, we can use it after the first branch for a single scale non-local operation. Considering there are two different inputs on the remaining branches, we need to modify equation (1) to equation(2).
Here $x_1$ represents the output of the first branch and $x_s$ represents the set of the outputs of the remaining branches. When $s$ equals 1, equation (2) is equal to equation (1), and the cross-scale self-similarity calculation is performed when $s$ represents other branches. If we directly perform non-local operation on the full map, the computation will be very big. So we also divide the map into $2 \times 2$ regions and performs non-local operation on each region. In order to further reduce the computational burden, we adopt RCCA instead of the non-local attention. So the final structure of TRRCCA is shown in the Figure 2 below.

3.3. Multiple-Scale Self-Similarity Network (MSSSNet)

In SISR network, EDSR, RCAN, SAN, CSNLN, etc. all adopt this mainstream framework as shown in the Figure 3 below. Image input into the network will first through a layer of convolution, then into the body part, and finally upsampling. The body part often consists of various basic module (BM) and connections.

To demonstrate the effectiveness of the proposed MSSS module. In this paper, we choose the most representative and simplest EDSR as the backbone. The MSSS is inserted into the middle of a series of BM to form the MSSSNet. As shown in Figure 4.

4. Experiments

4.1. Setting

Following EDSR, RCAN, we used 800 training images from DIV2K[15] dataset as the training set. For testing, we used four standard benchmark datasets: Set5[16], Set14 [17], B100[18], Urban100[19].
Degradation models are Bicubic (BI). The SISR results were evaluated with PSNR and SSIM on the Y channel of transformed YCbCr space. Data augmentation was performed on the 800 training images, which were randomly rotated by 90°, 180°, 270° and flipped horizontally. In each training batch, 16 LR color patches with the size of 48×48 were extracted as inputs. Our model was trained by ADAM optimizer with β1=0.9, β2=0.999, and ε=10^{-8}. The initial learning rate was set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We used PyTorch to implement our models. MSSSNet had 32 BM and the number of internal channels was 256. MSSS was inserted in the middle of the 16th and 17th BM.

4.2. Ablation Experiments
To demonstrate the effectiveness of MSSS, RFB and TRRCCA were inserted into EDSR in series with the same location as MSSS, forming RT-EDSR as shown in Figure 5. Here RFB only expanded the receptive field.

![Figure 5. RT-EDSR](image)

The results of MSSSNet and RT-EDSR for ×2 SR were compared as shown in Table 1.

| Method     | Set5 PSNR | Set5 SSIM | Set14 PSNR | Set14 SSIM |
|------------|-----------|-----------|------------|------------|
| EDSR       | 38.11     | 0.9602    | 33.92      | 0.9195     |
| RT-EDSR    | 38.19     | 0.9613    | 33.96      | 0.9205     |
| MSSSNet    | 38.24     | 0.9613    | 34.01      | 0.9209     |

MSSSNet and RT-EDSR do not differ in any way other than the blocks inserted into them. Table 1 showed that the MSSS with the multiple-scale self-similarity does outperform the simple superposition of RFB and TRRCCA. Because RT-EDSR only used single-scale self-similarity.

4.3. Results with Bicubic (BI) Degradation Model
We compared our method with 12 state-of-the-art methods: SRCNN, FSRCNN[20], VDSR[21], LapSRN[22], MemNet[23], EDSR, NLRN[24], DBPN[25], RDN, RCAN, SAN, CSNLN. All the quantitative results were reported in Table 2.

It can be seen that MSSS improved significantly over EDSR, while MSSS also surpassed the vast majority of methods, but it was slightly less than the three methods of RCAN, SAN, CSNLN. Because RCAN added channel attention into every basic module. SAN not only added a region-level non-local module before and after the body, but also added a second-order channel attention in each basic module. CSNLN also inserted CS-NL attention module into every basic module. MSSS just inserted a MSSS module in the middle of the EDSR without making changes to each basic module, so MSSSNet was slightly below the three methods in the evaluation metrics. The advantage was that the training time of MSSSNet is only about half of this three methods.

We also showed the zoomed results of representative methods in Figure 6, the enlarged areas were marked with red boxes. It can be seen from Figure 6 that some SR models produced mis-oriented textures, but MSSSNet did not.
Table 2. Quantitative results with BI degradation model.

| scale | Method  | Set5   | Set14   | B100    | Urban100 |
|-------|---------|--------|---------|---------|----------|
|       | PSNR    | SSIM   | PSNR    | SSIM    | PSNR     | SSIM     |
| ×2    | Bicubic | 33.68  | 0.9304  | 30.24   | 0.8691   | 29.56    | 0.8435   | 26.88    | 0.8405   |
|       | SRCNN   | 36.66  | 0.9542  | 32.45   | 0.9067   | 31.36    | 0.8879   | 29.51    | 0.8946   |
|       | FSRCNN  | 37.05  | 0.9560  | 32.66   | 0.9090   | 31.53    | 0.8920   | 29.88    | 0.9020   |
|       | VDSR    | 37.53  | 0.9590  | 33.05   | 0.9130   | 31.90    | 0.8960   | 30.77    | 0.9140   |
|       | LapSRN  | 37.52  | 0.9591  | 33.08   | 0.9130   | 31.08    | 0.8950   | 30.41    | 0.9101   |
|       | MemNet  | 37.78  | 0.9597  | 33.28   | 0.9142   | 32.08    | 0.8978   | 31.31    | 0.9195   |
|       | EDSR    | 38.11  | 0.9602  | 33.92   | 0.9195   | 32.32    | 0.9013   | 32.93    | 0.9351   |
|       | NLRN    | 38.00  | 0.9603  | 33.46   | 0.9159   | 32.19    | 0.8992   | 31.81    | 0.9246   |
|       | DBPN    | 38.09  | 0.9600  | 33.85   | 0.9190   | 32.27    | 0.9000   | 32.55    | 0.9324   |
|       | RDN     | 38.24  | 0.9614  | 34.01   | 0.9212   | 32.34    | 0.9017   | 32.89    | 0.9353   |
|       | RCAN    | 38.27  | 0.9614  | 34.11   | 0.9216   | 32.41    | 0.9026   | 33.34    | 0.9384   |
|       | SAN     | 38.31  | 0.9620  | 34.07   | 0.9213   | 32.42    | 0.9028   | 33.10    | 0.9370   |
|       | CSNCLN  | 38.28  | 0.9616  | 34.12   | 0.9223   | 32.40    | 0.9024   | 33.25    | 0.9386   |
|       | MSSSSNet| 38.24  | 0.9613  | 34.01   | 0.9209   | 32.35    | 0.9017   | 32.99    | 0.9360   |

Figure 6. Visual comparison for ×2 SR model on Urban100 datasets.

5. Conclusion
We propose a MSSSSNet for SISR, MSSS module can be inserted into other similar SISR networks to bring promotion besides EDSR. The biggest innovation of MSSS module is that unlike previous methods that only use self-similarity on one or two scales, MSSS uses self-similarity on multiple scales. In the MSSS module we use a combination of different kernels and dilated convolutions to obtain feature maps at different scales. We also propose TRRCCA, TRRCCA greatly reduces the
calculation burden and greatly increases the non-local operation speed by dividing the feature maps into small regions and performing RCCA in each small region. In the meantime, we make some changes to RCCA in the TRRCCA to make it more suitable for multi-scale feature maps by changing it from a single input port to two inputs ports. We experimentally demonstrate the effectiveness of the MSSS module. Although it doesn't surpass some of the best methods, it is very close to them with a simpler structure. In the future, we will study how to better integrate the MSSS into each basic module.

References

[1] Glasner, D., Bagon, S., & Irani, M. (2009, September). Super-resolution from a single image. In 2009 IEEE 12th international conference on computer vision (pp. 349-356). IEEE.

[2] Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell, 38(2), 295-307.

[3] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4681-4690).

[4] Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 136-144).

[5] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2472-2481).

[6] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In Proceedings of the European conference on computer vision (ECCV) (pp. 286-301).

[7] Dai, T., Cai, J., Zhang, Y., Xia, S. T., & Zhang, L. (2019). Second-order attention network for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11065-11074).

[8] Mei, Y., Fan, Y., Zhou, Y., Huang, L., Huang, T. S., & Shi, H. (2020). Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5690-5699).

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[10] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834-848.

[11] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 764-773).

[12] Liu, S., & Huang, D. (2018). Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 385-400).

[13] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7794-7803).

[14] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019). Cnet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 603-612).
[15] Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 126-135).

[16] Bevilacqua, M., Roumy, A., Guillemot, C., & Alberi-Morel, M. L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding.

[17] Zeyde, R., Elad, M., & Protter, M. (2010, June). On single image scale-up using sparse-representations. In International conference on curves and surfaces (pp. 711-730). Springer, Berlin, Heidelberg.

[18] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001, July). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001 (Vol. 2, pp. 416-423). IEEE.

[19] Huang, J. B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5197-5206).

[20] Dong, C., Loy, C. C., & Tang, X. (2016, October). Accelerating the super-resolution convolutional neural network. In European conference on computer vision (pp. 391-407). Springer, Cham.

[21] Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1646-1654).

[22] Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 624-632).

[23] Tai, Y., Yang, J., Liu, X., & Xu, C. (2017). Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE international conference on computer vision (pp. 4539-4547).

[24] Liu, D., Wen, B., Fan, Y., Loy, C. C., & Huang, T. S. (2018). Non-local recurrent network for image restoration. arXiv preprint arXiv:1806.02919.

[25] Haris, M., Shakhnarovich, G., & Ukita, N. (2018). Deep back-projection networks for super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1664-1673).