Persistent Memory Residual Network for Single Image Super Resolution

Rong Chen[1], Yanyun Qu* [1], Kun Zeng[2], Jinkang Guo[1], Cuihua Li[1], Yuan Xie[3]
[1]School of Information Science and Engineering, Xiamen University, Xiamen, China
[2]College of Electronic Science and Technology, Xiamen University, Xiamen, China
[3]Research Center of Precision Sensing and Control, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
chenrong_mail@qq.com, yyqu@xmu.edu.cn, zengkun301@aliyun.com,
jinkang.guo@gmail.com, chli@xmu.edu.cn, yuan.xie@ia.ac.cn

Abstract

Progresses has been witnessed in single image super-resolution in which the low-resolution images are simulated by bicubic downsampling. However, for the complex image degradation in the wild such as downsampling, blurring, noises, and geometric deformation, the existing super-resolution methods do not work well. Inspired by a persistent memory network which has been proven to be effective in image restoration, we implement the core idea of human memory on the deep residual convolutional neural network. Two types of memory blocks are designed for the NTIRE2018 challenge. We embed the two types of memory blocks in the framework of enhanced super resolution network (EDSR), which is the NTIRE2017 champion method. The residual blocks of EDSR is replaced by two types of memory blocks. The first type of memory block is a residual module, and one memory block contains four residual modules with four residual blocks followed by a gate unit, which adaptively selects the features needed to store. The second type of memory block is a residual dilated convolutional block, which contains seven dilated convolution layers linked to a gate unit. The two proposed models not only improve the super-resolution performance but also mitigate the image degradation of noises and blurring. Experimental results on the DIV2K dataset demonstrate our models achieve better performance than EDSR.

1. Introduction

Image super-resolution aims at restoring rich details of a high-resolution (HR) image from an LR image or a sequence of low-resolution images without additional hardware support. Moreover, super-resolution (SR) is an ill-posed problem because an LR image can be generated by a large subspace of high-resolution images. Until now, SR is still a challenging task due to the complex degradation in the wild, such as image noises, blurring, downsampling and so on.

Deep learning methods have been successfully applied to SR, but most of the deep learning-based SR methods are usually trained on the traditional training dataset, such as bicubic downsampling which simply simulates the realistic image degradation process. However, LR images are generated by more complex degradation than bicubic downsampling in the realistic world. In the NTIRE2018 challenge on single image super-resolution [16], DIV2K is a more challenging training dataset than the traditional SR training dataset. The degradations on the DIV2K dataset include noises, blurring, displacement or combination of the above-mentioned degradation with unknown downsampling. Previous SR methods may not achieve good performance on the real LR images. Further studies are required for the S-R problem of LR images, which are generated from more complex degradations.

In this paper, we focus on improving the SR problem of complexity degradation LR images. Motivated by human long-term memory, we implement persistent memory block [14] by using deep residual convolution neural network in which the long-path mimics human long-term memory. We use SR architecture similar to EDSR [10], which is the N-TIRE2017 champion method in the task of SR. We design two novel models: the Memory Enhanced Deep Super Resolution (MemEDSR) and Image Restoration with Memory network (IRMem). Both the proposed models overcome not only the degradation of downsampling but also the degradation of blurring and noises. In the MemEDSR model, we replace the residual blocks of EDSR with a memory block. The memory block is composed of four residual modules with four residual blocks, and the output of each module links to a gate unit, in order to realize long-path links. In the IRMem model, we design a residual dilated convolu-

[*] Corresponding author

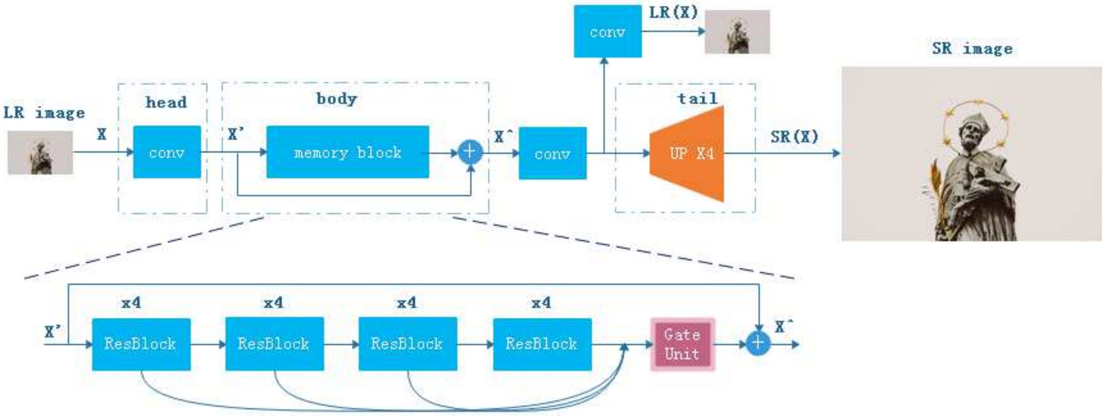

Figure 1. The architecture of the proposed MemEDSR network.

tional module as a memory block. Several memory blocks are used for constructing fine details, and each module is composed of seven dilated convolutional layers. In the end, the output of each memory block links to a gate unit, which adaptively selects the features needed to store.

The contributions are summarized as follows:

(1) Two types of memory blocks are designed for solving the SR problem, which are caused by complex degradation such as downsampling, blurring and noises and so on.

(2) Two deep network models (i.e., MemEDSR and IRMem) are proposed, and each model contains different persistent memory blocks and residual short cut links in order to tackle the four tracks in the NTIRE2018 challenge on single image super-resolution.

(3) Extensive experiments are conducted on the DIV2K dataset. We discuss three important factors in the proposed models: the persistent memory block, the loss function, and the batch normalization. Qualitative and quantitative results show that the proposed two models can achieve good results on Track 2,3,4 in the NTIRE2018 challenge on single image super-resolution.

2. Related Work

2.1. Image SR via Deep Learning

Great progress has made in deep learning-based SR methods. SRCNN [2], which first implemented the convolutional neural network on SR, achieved the milestone SR performance. After that, many CNN-based SR methods rose up [4, 6, 7, 8, 9, 10, 12, 17, 19, 22]. In VDSR [6] and DRCN [7], convolutional networks contain deeper convolutional layers so that the receptive field in the original image is enlarged. They made a great improvement in the SR performance. Different from the previous SR methods which takes the magnified image by bicubic interpolation as the input of SR network, ESPCN [12] directly extracts image features in the LR space. The sub-pixel convolutional layer learns an array of upscaling filters to upscale the final LR feature maps into the HR output. ESPCN is optimal and reduces the computational complexity. Similar to ESPCN [12], FSRCNN [3] also does features extraction in the LR space and deconvolution is used for image reconstruction. Some variants of ESPCN [12], such as SRResNet [9], DRRN [13] and EDSR [10] use the deeper network to improve performance. In the NTIRE2017 challenge on single image super-resolution [15], EDSR [10] has achieved impressive results and won this competition. It removes batch normalization (BN) layer in its neural network architecture in order to effectively reduce memory consumption and uses a residual scaling factor to stabilize the training of the model.

However, the above-mentioned methods do not solve well the SR problem caused by the complex degradation. These methods are good at dealing with the degradation caused by bicubic downsampling but neglect deblurring and denoising.

2.2. Image Restoration via Deep Learning

Image restoration is a classical problem in the field of computer vision. BM3D [1] is one of the famous deblurring methods in the traditional model-based optimization algorithms. Inspired by the great success of deep learning in image classification and speech recognition, the deep convolutional neural network is widely used to solve the problem of image restoration [20, 11, 18, 14, 21]. MemNet [14] and IRCNN [21] are two latest image restoration methods. MemNet [14] is proposed by the inspiration of human persistent memory. A persistent memory network is designed for image restoration, to explicitly mine persistent memory through the adaptive learning process. Similar to DenseNet [5], the network connects each layer to every other layer in a feed-forward fashion, thus, the feature map learned by this layer is also directly passed to all subsequent layers as the

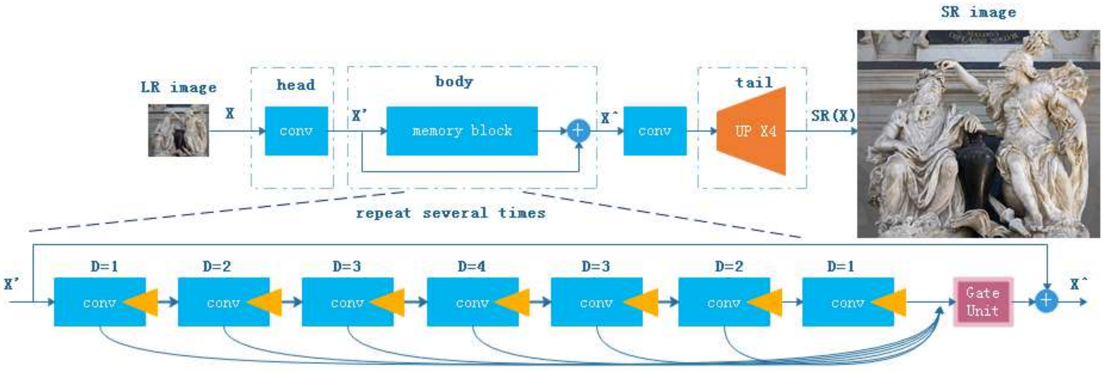

Figure 2. The architecture of the proposed IRMem network.

input (thoughts). Finally, the results of the reconstruction of each module are added according to a certain weight ratio to generate the final SR result (the idea of DRCN [7]). It achieves excellent results in image restoration. Zhang et al. proposed IRCNN [21], which combines the model-based optimization methods with deep learning based discriminative denoising. IRCNN uses the dilated convolution layers to enhance the receptive field in the training, and contains seven layers with different dilated filter parameters. IRCNN has a good effect on image denoising, deblurring, and SR.

3. Proposed Methods

It is recognized in neuroscience that there are many recursive connections ubiquitously existing in the neocortex. Motivated by the recursive connections, MemNet [14] proposes the persistent memory block, which contains a recursive unit and a gate unit. The recursive unit can mimic the short-term memory, and a gate unit can mimic human long-term memory. Each recursive unit links to the gate unit in a memory block. Inspired by the effectiveness of MemNet on deblurring and denoising, we embedded the persistent memory blocks in our models. We propose two models (i.e., MemEDSR and IRMem) to tackle the SR tracks in the NTIRE2018. In the following, we introduce two deep networks in detail.

3.1. MemEDSR

We adopt similar network architecture to EDSR [10], which contains three parts: the head part, the body part, the tail part. The head part contains one convolutional layer, and the body part contains several residual blocks, and the tail part tackles the upsampling reconstruction like ESPCN [12]. A residual network structure contains 64 filter kernels and the residual scale is set at 0.1.

In MemEDSR, we use a memory block which contains four residual modules in the body part. Each residual module contains four residual blocks and is regarded as a recursive unit. Each recursive unit links to a gate unit. Similar to EDSR, BN layers are removed in MemEDSR. The framework of the proposed MemEDSR is shown in Figure 1.

There are two branches at the back of the memory block. One branch is used for upsampling the output of the memory block so as to generate the SR image. The other branch is used to reconstruct the LR image. For the training set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $x^{(i)}$ is the LR patch, and $y^{(i)}$ is the ground truth patch. We denote the reconstruction of an LR patch by $LR(x^{(i)})$ and denote the bicubic downsampling of the SR result generated by the network by $D(SR(x^{(i)}))$. The loss function is formulated as Eq.(1),

$$\begin{aligned} L(\Theta) = & \frac{1}{2N}\sum_{i=1}^{N}\left(\left\|y^{(i)} - SR(x^{(i)})\right\|_1\right) + \\ & \frac{1}{2N}\sum_{i=1}^{N}\left(\left\|D(SR(x^{(i)})) - LR(x^{(i)})\right\|_1\right). \end{aligned} \tag{1}$$

The loss function contains two terms. The first loss term is the fidelity of the SR result to the ground truth image, and the second loss term is the fidelity of the reconstruction of the LR image to the downsampling result of the SR image by bicubic downsampling.

3.2. IRMem

As shown in Figure 2, we modify a dilated convolution block of IRCNN [21] into a memory block of the IRMem model. We implement the core idea of persistent memory on an IRCNN block. In an IRCNN memory block, there are seven dilated convolutional layers, and the dilation factors are $1, 2, 3, 4, 3, 2, 1$ is shown in Table 1. We link the output of each dilated convolution layer to the gate unit, and all the output features are concatenated as the input of the gate unit. Draw lessons from EDSR, we remove all the BN layers in IRCNN memory block due to its high memory-consuming. Furthermore, we add a short cut from the head to the output of the gate unit in each IRCNN memory block, and the residual scale is set to 0.1. We repeat the IRCNN memory block several times for enhancing the performance of the model. And then we embed all the memory blocks into the body part. The advantages of IRMem are three folds:1) IRMem enlarges the receptive field of the convolution layers; 2) IRMem adaptively controls how much of the previous features should be reserved; and 3) IRMem decides how much of the current features should be stored.

Layer	Filter	Dilation	Padding	Succedent layer
Conv1	3*3	1	1	Relu
Conv2	3*3	2	2	BN+Relu
Conv3	3*3	3	3	BN+Relu
Conv4	3*3	4	4	BN+Relu
Conv5	3*3	3	3	BN+Relu
Conv6	3*3	2	2	BN+Relu
Conv7	3*3	1	1	-

Table 1. A Module Composition of IRCNN[21].

4. Experimental Results

4.1. Experiment Setup

DIV2K 2018 dataset is a high-quality image dataset containing 1000 images ($\sim 2K$ resolution in width or height) for single image super-resolution tasks with four tracks. The images are degraded in different ways to form four corresponding LR datasets corresponding to four tracks: 1) Track 1: Classic bicubic - $\times 8$ in which the degradation of an image is generated by bicubic downsampling with the factor 8, 2) Track 2: Realistic mild - $\times 4$ in which an LR image is a downsampling version with the factor 4 together with noises, 3) Track 3: Realistic difficult - $\times 4$ in which an LR image is a downsampling version with the factor 4 together with noises and blurring, and 4) Track 4: Realistic wild - $\times 4$ in which an LR image is a downsampling version with the factor 4 with noises, blurring and displacement. Every track contains 800 training images, 100 validation images, and 100 test images. HR image patches from HR images with the size of $192 * 192$ are randomly sampled for training. An HR image patch and its corresponding LR image patch are treated as a training pair. In our experiments, we only use the datasets given in NTIRE2018 challenge on SR without using the external datasets for training.

We implement our proposed SR models on the four tracks. For each track, MemEDSR contains one memory block with 16 residual blocks, and 64 feature maps. IRMem contains 16 memory blocks, and 64 feature maps in each memory block. It is the trade-off between the accuracy and speed of the model.

Drawing on the experience of existing SR methods, we set our two models with the same parameters for training. Experimentally, we set the initial learning rate to $1e-4$, and we use ADAM learning optimization algorithm to update the parameters of our models, which makes the proposed network converge fast. The two parameters β_1 and β_2 are set to 0.9 and 0.999, respectively. 16000 training image patch pairs in each track are used for training. The batch size is 16. We use l_1 norm in the loss function, and use the validation set to verify the performance of the proposed models. We adapt PSNR and SSIM as the criteria which are computed according to the rules given by the organizer of NTIRE2018.

All the experiments are implemented in the platform Ubuntu 16.04 with GTX1080 GPU and 32G CPU Memory. The development environment is Pytorch $0.3.0$.

4.2. Model Analysis

In this subsection, we discuss three important components of our models to the SR performance: the BN layer, the loss function, and the number of convolution layers in IRMem. We first analyze the effect of the BN layer on SR. For the IRMem model, we give the implementation details in Table 1. We compare IRMem without BN layers and IRMem with BN layers (denoted by IRMem+BN). In Table 2, we show the comparison results between IRMem and IRMem+BN on the first and third track datasets. The results demonstrate that removing BN can improve the SR performance a little. It also demonstrates that BN influences the SR performance a little more in Difficult$\times 4$ than in Cubic$\times 8$. IRMem achieves the gain of PSNR and SSIM by $(0.1, 0.032)$ in Difficult$\times 4$ and by $(0.0039, 0.0008)$ in Cubic$\times 8$, compared with IRMem$+$BN.

Methods	Difficult×4	Cubic×8
IRMem	22.22/0.4811	25.3749/0.6888
IRMem+BN	22.12/0.4779	25.3710/0.6880

Table 2. Comparison results between IRMem and IRMem+BN on the Difficult$\times 4$ and Cubic$\times 8$ datasets.

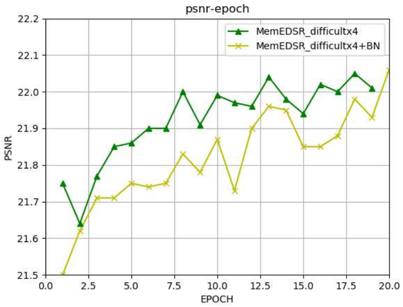

Figure 3. Experimental comparison between MemEDSR and MemEDSR+BN on the Difficult$\times 4$ dataset.

Moreover, we estimate the effect of BN layer to MemEDSR and make the comparison between MemEDSR and MemEDSR with BN layers (MemEDSR+BN). Figure 3 shows the comparison results between MemEDSR and MemEDSR+BN on the Difficult×4 dataset. Table 2 and Figure 3 both demonstrate that removing BN will improve the SR performance. The positive gain is about $0.1db$ in terms of PSNR.

We estimate the effect of different loss functions on SR and formulate the traditional loss function as Eq.(2), the fidelity of the SR result to the ground truth.

$$L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \left(\left\| y^{(i)} - SR(x^{(i)}) \right\|_1 \right) \tag{2}$$

We denote the model with the traditional loss function as Eq.(2) by MemEDSR-LF1. The comparison results between MemEDSR and MemEDSR-LF1 on the Cubic×8 and Mild×4 datasets are shown in Table 3. It demonstrates that the loss function with two loss terms as Eq.(1) is better than Eq.(2).

Methods	Mild ×4	Cubic ×8
MemEDSR	22.7511/0.5064	25.1439/0.6800
MemEDSR-LF1	22.7027/0.5026	25.1124/0.6783

Table 3. Comparison of MemEDSR with different loss functions in terms of PSNR and SSIM on the Mild×4 and Cubic×8 datasets.

Furthermore, we discuss how the number of convolutional layers affects the SR performance in the IRMem. We denote B as the number of memory blocks and F as the number of feature maps. In the implementation of IRMem, we set $B = 16$ and $F = 64$. We compare it with the IRMem model with $B = 24$, $F = 64$, denoted by IRMem*. Table 4 gives the comparison results, which demonstrates that the more the block number is, the better the SR performance is.

Methods	IRMem	IRMem*
	(B=16 F=64)	(B=24 F=64)
Mild ×4	22.7511/0.5064	22.9424/0.5107

Table 4. The effect of the number of the memory blocks in IRMem on the Mildx×4 dataset.

4.3. Comparison with the state-of-the-art SR methods

In this subsection, we compare the proposed models with the state-of-the-art SR methods. We set the bicubic interpolation method as the baseline. Six state-of-the-art SR methods are compared with our two proposed models (i.e., MemEDSR, IRMem) on DIV2K: Bicubic, SRCNN [2], VDSR [6], CDA [19], EDSR [10], IRCNN [21]. In our experiments, EDSR and IRCNN are retrained on DIV2K. As for EDSR, the published best SR results are achieved with 32 memory blocks, each of which contains 256 feature maps $(B = 32, F = 256)$. For the limitation of the computational resource, we cannot run our model with $B = 32$ and $F = 256$ but only run the model with $B = 16$ and $F = 64$. For the fairness of comparison, we compare our model with EDSR with $B = 16$ and $F = 64$, denoted by EDSR*. The code of EDSR used in our experiments is supplied by the third party with the development environment of Pytorch. As for IRCNN and IRMem, we set the same network structure with 16 dilated convolution blocks and 64 feature maps. For SRCNN, VDSR, and CDA, we use the models provided by their authors and we test the models on the NTIRE2018 validation datasets for three tracks. As for the proposed models, we firstly train MemEDSR and IRMem on the Mild×4 dataset, respectively. And then, based on the two models, we train the new models on Difficult×4 and Wild×4 datasets, respectively. During training on the Mild×4 dataset, 60 epochs are trained with the initial learning rate. After that, 120 epochs are trained with half of the previous learning rate. The same strategy is used in the Difficult×4 and Wild×4 datasets.

The comparison results of different methods on the last three tracks are shown in Table 5. Each item has two digitals in the results: the first is the averages of PSNRs, and the second is the average of SSIMs. We find that Bicubic achieves the best SR performance in terms of PSNR and SSIM on Mild×4 dataset. We further observe that the results of the models retrained on Mild×4 are inferior to Bicubic, SRCNN, and CDA. Together with the visual effect comparison in Figure 4, we guess that models retrained on the Mild×4 dataset make the SR result over-smoothed. Thus, their PSNRs are less than Bicubic, SRCNN, and CDA. On the Difficult×4 and Wild×4 datasets, the two proposed models achieve the best SR results among the 8 methods and IRMem is better than MemEDSR.

Methods	Mild×4	Difficult ×4	Wild ×4
Bicubic	**23.14/0.5166**	21.77/0.4590	22.45/0.4857
SRCNN [2]	22.96/0.5062	21.54/0.4431	22.16/0.4681
VDSR [6]	22.86/0.4984	21.50/0.4387	22.10/0.4621
CDA [17]	22.99/0.5065	21.57/0.4440	22.21/0.4694
EDSR* [10]	22.70/0.5029	22.13/0.4753	22.53/0.4943
IRCNN [21]	22.68/0.5014	22.10/0.4742	22.54/0.4921
MemEDSR	22.78/0.4993	22.13/0.4727	22.57/0.4942
IRMem	22.75/0.5064	**22.22/0.4801**	**22.70/0.4986**

Table 5. Comparison different methods on the last three datasets in the NTIRE2018 challenge.

We also compare our models with EDSR on the five datasets: $Set5$, $Set14$, BSD100, Urban100, and DIV2K validatation. We estimate our models for the SR problem with the simple degradation caused by bicubic downsam-

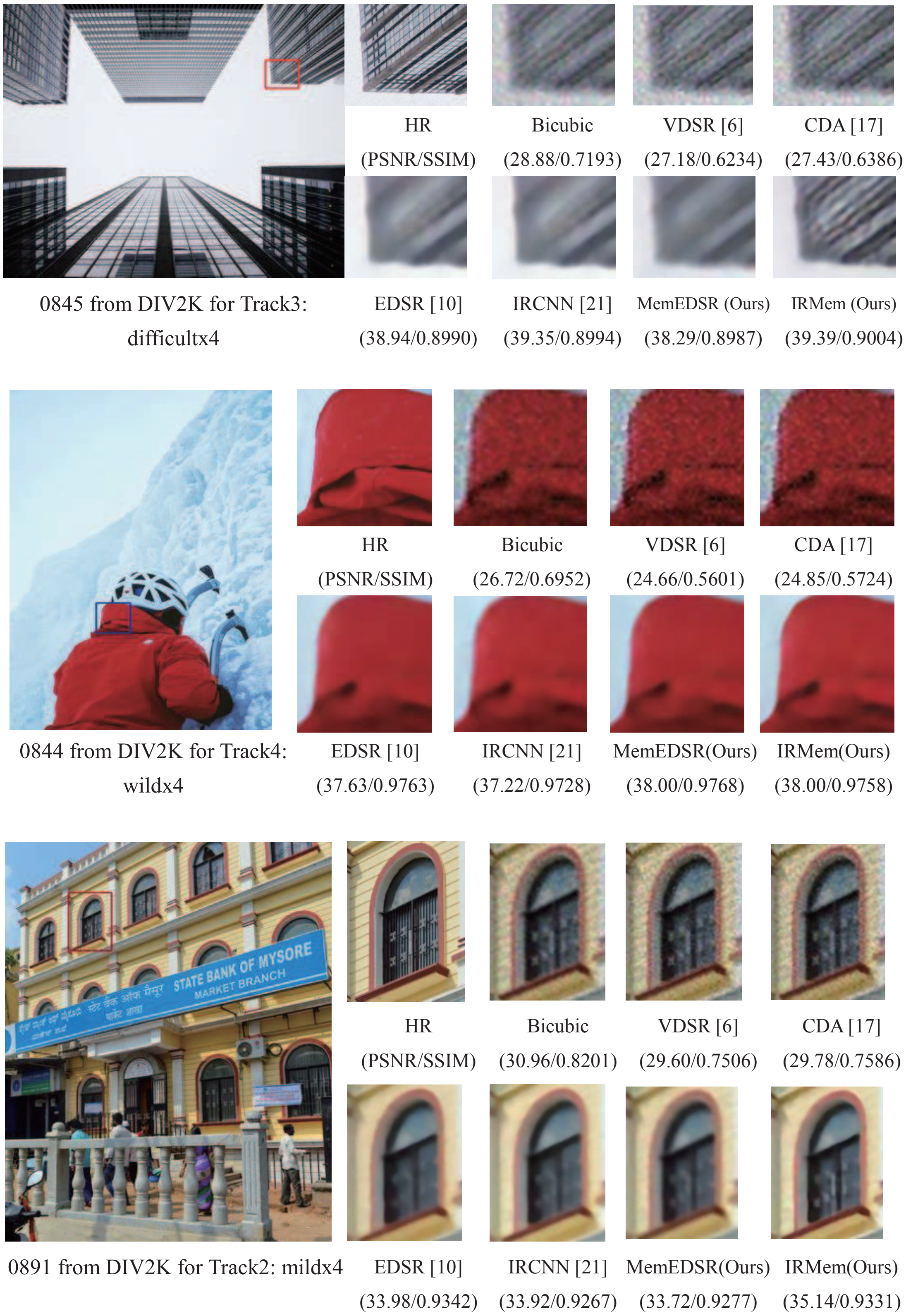

Figure 4. Visual effect comparison of the five SR methods on three Tracks of the NTIRE2018 challenge.

pling with the factor 4. For the comparison fairness, we use the same network structure configuration for EDSR and MemEDSR with 16 residual blocks, and IRMem has 16 memory blocks, each of which contains 64 feature maps $(B = 16, F = 64)$. Table 6 shows the comparison results. It demonstrates that our models outperform EDSR on all datasets. On $Set5$ and $Set14$, IRMem achieves the highest gain by about $3db/0.09$ in terms of PSNR and SSIM. On DIV2K, MemEDSR is superior to EDSR with the gain $(1.35, 0.0405)$ and IRMem is superior to EDSR with the gain $(2.03.0.0702)$ in terms of PSNR and SSIM. IRMem achieves better SR results than MemEDSR.

Methods	EDSR*	MemEDSR	IRMem
	(B=16 F=64)	(B=16 F=64)	(B=16 F=64)
Set5	27.34/0.7774	29.64/0.8524	**30.36/0.8655**
Set14	24.93/0.6786	26.35/0.7293	**26.88/0.7453**
BSD100	23.54/0.5697	**23.66**/0.5815	23.65/**0.5831**
Urban100	21.13/0.5701	21.47/0.6002	**21.52/0.6080**
DIV2K VAL	27.03/0.7528	28.38/0.8033	**29.06/0.8210**

Table 6. Comparison between EDSR and our models with magnification factor 4 on five datasets. In the experiments, EDSR* is the model with $B = 16$ and $F = 64$.

In the following, we give the visual effect comparison of the five compared methods. Figure 4 shows the comparison of the visual effect on the last three track datasets. From the visual effect, our proposed models can give more fine details than other SR methods. Because SRCNN, CDA, and VDSR are the original models proposed by their authors which are trained on a bicubic downsampled training dataset, they have little effect on noises. EDSR and IRCNN can mitigate noises but their results look blurring. IRMem is superior to other comparied SR methods, and it can not only denoise but also keep fine details.

5. Conclusion

In this paper, we focus on the SR problem in which LR images are degraded by complex degradation which contains downsampling, blurring, and noises. DIV2K is a more difficult SR dataset than the previous SR dataset. We implement the core idea of human memory on the deep neural network for SR. We design two models (i.e., MemEDSR and IRMem) with the different memory blocks. Similar to EDSR, which is the NTIRE2017 champion SR method, the proposed models contain three parts: the head part, the body part, and the tail part. In MemEDSR, we use a memory block to replace the body part of EDSR. The memory block contains four residual modules and each residual module contains four residual blocks. The output of each residual module links to a gate unit. In IRMem, we design a memory block which is composed by a residual dilated convolutional block. Each memory block contains seven dilated convolution layers and the each of them is linked to a gate unit. The memory block is repeated several times for constructing fine detail. The experimental results show that the two proposed models achieve better SR performance than EDSR in Track 2, 3 and 4. It demonstrates the effectiveness of our models.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 61373077, Grant 61772524, and Grant 61601389 and in part by the Beijing Natural Science Foundation under Grant 4182067.

References

[1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080, 2007.

[2] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.

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

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

[5] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.

[6] 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 (CVPR)*, pages 1646–1654, 2016.

[7] 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 (CVPR)*, pages 1637–1645, 2016.

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

[9] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690, 2017.

[10] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, volume 1, page 3, 2017.

[11] X. Mao, C. Shen, and Y. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *CoRR*, abs/1606.08921, 2016.

[12] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016.

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

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

[15] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1110–1121. IEEE, 2017.

[16] R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, et al. Ntire 2018 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018.

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

[18] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep image prior. *CoRR*, abs/1711.10925, 2017.

[19] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao. Coupled deep autoencoder for single image super-resolution. *IEEE transactions on cybernetics*, 47(1):27–37, 2017.

[20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[21] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3929–3938, 2017.

[22] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. *arXiv preprint arXiv:1802.08797*, 2018.