Real-Time Deep Image Super-Resolution via Global Context Aggregation and Local Queue Jumping

Yueyu Hu, Jiaying Liu*, Wenhan Yang, Shihong Deng, Luyao Zhang and Zongming Guo
Institute of Computer Science and Technology, Peking University, Beijing, P. R. China, 100871

***Abstract*—Deep learning-based image super-resolution has provided very impressive reconstruction quality. However, their running time still sets barriers for real-time applications. In this paper, we propose a Global context aggregation and Local queue jumping Network (GLNet) which provides the more effective image SR given a certain number of model parameters. In our GLNet, we reconsider the model design of the real-time image SR paradigm. Then, we construct a deep network with fewer channels but a deeper structure to effectively aggregate the global context. The dilated convolutions are used as parts of basic units of our GLNet, which further enlarges the receptive field. Besides, an additional local queue jumping path is employed to connect the first-layer feature map and the last-layer feature map to better model the local signal structure. Extensive experiments demonstrate the superiority of our GLNet which offers new state-of-the-art performance considering both reconstruction quality and time consumption.**

***Index Terms*—Real-Time Image Super-Resolution, Global Context Aggregation, Local Queue Jumping, Deep Learning**

I. Introduction

Nowadays, it is a common demand to embrace the high-quality video display. Limited by current hardware conditions, the presence of image and video super-resolution (SR) is promising. SR enlarges a low-resolution (LR) video to a high-resolution (HR) one solely from the perspective of software. In past decades, many models are proposed to build the mapping between LR and HR space, *i.e.* Markov random field [1], neighbor embedding [2], sparse coding [3], and anchor regression [4], *etc.* These efforts bring the impressive reconstruction visual quality to us, however, most of these works are still far away from the practical use because of the low time efficiency. The presence of deep learning and artificial neural networks is changing the reality.

In recent years, deep neural networks show great potential in high-level computer vision tasks like object detection [5], [6], face recognition [7], and pedestrian detection[8]. CNN is also applied to address image restoration problems like denoising, dehazing, and super-resolution. Super-Resolution methods based on deep learning framework were first proposed in [9], [10]. With convolutional operations modeling transformation from LR images to HR images, these methods are superior to sparse-coding-based methods in performance. To be the pioneer of deep super-resolution, the network only consists of three convolutional layers, and later works [11], [12] reach state-of-the-art performance by deepening the network, which adds more nonlinearities and expands the receptive field of one neuron. These characteristics enable the deeper architectures to take richer context into consideration and to model non-linear transformation of more complexity. To successfully train such a deep network, the residual network is proposed [13] to accelerate the training process. With the residual connection, gradients can be back-propagated to shallow layers in a very deep model.

Metrics for image super-resolution include both reconstruction quality and time-efficiency. One challenging problem for image super-resolution is to reduce the computational complexity of the algorithm to facilitate real-time applications like video surveillance and real-time video streaming. To accelerate the SR process, in [14], the feature extraction and transformation are performed in LR space. Faster super-resolution neutral network (FSRCNN) [15] provides the observations that, decreasing the number of channels will effectively reduce the number of the parameters of the network, and thus accelerate the SR process. Therefore, FSRCNN embeds the network shrinkage and expanding steps, to save a large part of model parameters and running time. Besides, it simplifies the structure of the network. These shrinkage and simplification results in a degradation of the quality of the reconstruction framework.

In this paper, we first revisit the real-time image SR. Then, to improve the modeling capacity, we reconsider the model design of the real-time image SR paradigms. Then, we construct a deep network to enlarge the receptive field and enhance the non-linear modeling capacity of the network. This is achieved by using a deeper architecture with dilated convolutions as its parts of basic units. Besides, an additional local queue jumping path is employed to connect the first-layer feature map and the last-layer feature map to better model the local signal structure. GLNet also has better scalability. By enlarging the number of parameters of the network, the quality can be further improved accordingly. Experimental results prove that with equal reconstruction quality, our model runs with lower latency, compared with state-of-the-art methods.

II. GLNet for Real-Time Image SR

A. Real-Time Image SR

As illustrated in [9], deep-learning-based image SR usually consists of three steps: 1) feature extraction; 2) nonlinear mapping; 3) reconstruction. This plain paradigm does not computationally efficient because its design does not consider the effect of the number of channels and spatial resolutions on the time complexity of deep networks.

To fasten the SR process, in [15], two improvements are employed. First, the feature extraction and non-linear mapping are performed in the LR space instead of the HR space, thus the feature map size to be processed in SR is largely reduced. Second, the number of channels plays an important role in the total number of parameters, thus the extracted features are projected into a low-dimensional space first before the non-linear mapping. However, in [15], two issues are neglected in its network design. First, the parameter allocation is not well optimized. For example, the kernel size of the convolutional layer for feature extraction is 5×5, which uses almost three times parameters than a 3×3 convolution. Second, when expanding the receptive field of a network by stacking the convolutional layers,

* Corresponding author. This work was supported by National Natural Science Foundation of China under contract No. 61472011.

978-1-5386-0462-5/17/$31.00 ©2017 IEEE.

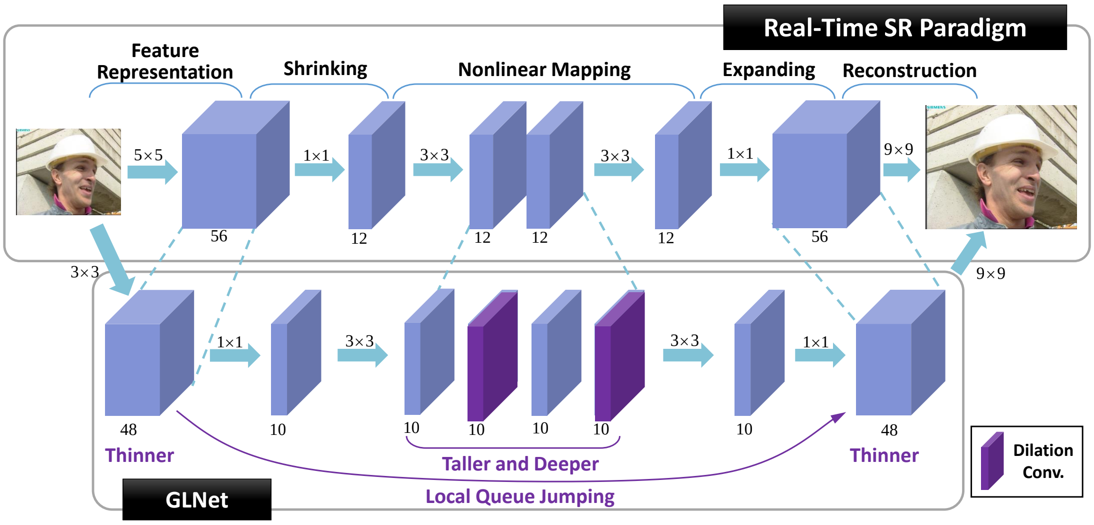


Fig. 1. GLNet adopts a thiner and deeper network structure for *global context aggregation* with an additional *local queue jumping* connection.

the capacity of the network to describe the local signal structures is weakened.

B. Time Complexity Analysis

To address these two issues, we first analyze the time complexity of the real-time image SR. The effort to make super-resolution methods time-efficient is critical to real-world applications. Traditional upscaling algorithms based on interpolation include a large number of matrix manipulation in estimating the lost information, which brings in high algorithm complexity. Example-based super-resolution algorithms solve this problem by separating the training and reconstruction phases. As the training process can be done based on an external dataset offline, we only need to care about the reconstruction time in practice. Deep learning based super-resolution inherits the characteristics of example-based methods, thus in practice, we only need to estimate the running time of the reconstruction phrase as follows,

$$O\left\{\left(f_1^2 n_1 + n_1 f_2^2 n_2 + n_2 f_3^2\right) S_{\mathrm{HR}}\right\}, \qquad (1)$$

where S_{HR} is the resolution of the HR image and n_i stands for the dimension of the feature vector for layer i. For the first layer, n_1 stands for the number of the channels of the input image. From Eqn. (1), it can be observed that, given a certain number of parameters, the width and depth of the network need to be balanced to achieve good performance. In our GLNet, we design a narrower and deeper network than FSRCNN to obtain superior performance.

C. Overall Architecture of GLNet

Fig. 1 shows the architecture of our proposed GLNet model. Following real-time SR paradigm, it goes through five steps for image and video SRs: feature extraction, shrinkage, nonlinear mapping, expanding and reconstruction. Comparing with existing real-time SR methods, GLNet has two distinguishing characteristics:

- A thinner and deeper network structure for *global context aggregation*. Each layer has fewer channels. Given a certain parameter number, the whole network is deeper. With dilated convolutions as parts of its units, GLNet has a very large receptive field.
- An additional *local queue jumping* connection between the first and penultimate layers enables the network to better describe the local signal structures.

Note that, functionally *global context aggregation* is achieved via the three parts of GLNet – the dimension shrinkage, non-linear mapping and dimension expanding phrases jointly.

D. Detailed Illustrations for GLNet

In the following, we discuss each part separately. Their basic settings are provided in Table I and II. For feature extraction, shrinkage, expanding and reconstruction phrases, $(\#1, \#2, \#3, \#4)$ denotes the number of input channels, height and width of the convolution kernel, and the number of output channels, respectively. For non-linear mapping, the additional $\#5$ denotes the layer number of the stacked convolutions.

*a) **Feature Extraction**:* In feature extraction phrase, GLNet learns a transformation in the LR space, and the spatial size of the output features is the same to the LR image. GLNet uses a 3×3 convolution as this layer, which saves almost 2/3 parameters than the 5×5 convolution in FSRCNN.

*b) **Dimension Shrinkage**:* From Eqn. (1), it is observed that, time complexity is largely determined by the dimension of the output feature vector, which is n_i in our formulation. In our GLNet, compared with FSRCNN, we use a narrower network with fewer channels. Thus, a single basic convolution layer of our GLNet saves more parameters than that of FSRCNN.

*c) **Nonlinear Mapping**:* Increasing the size of the filter kernels, the depth of the convolution layers and the dimension of the feature maps can result in increasing both the quality of the output image and the time complexity. Thus, we face a trade-off between quality and time complexity. Jointly with dimension shrinkage, we use fewer channels in each convolution layer but stack more ones. Comparatively, for the non-linear mapping, the number of channels of GLNet and FSRCNN is 10 and 12, respectively. And the depth of them is 4 and 8, respectively. As a result, GLNet is deeper than FSRCNN. For GLNet, half of the convolutions used in non-linear mapping are dilated convolutions with the dilation factor as 2 as shown in Fig. 2. Thus, the receptive field of GLNet can reach 29×29 comparing with 15×15 of FSRCNN.

d) Dimension Expanding: The dimension expanding is the inverse manipulation of previous dimension shrinkage. This operation greatly improves the quality of the constructed image.

e) Sub-Pixel Reconstruction: The up-scaling operations projecting the LR features to the HR ones are performed via sub-pixel

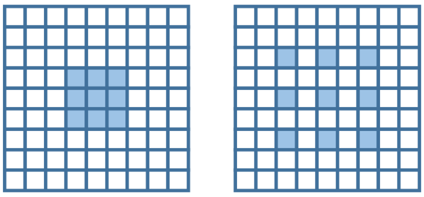
978-1-5386-0462-5/17/$31.00 ©2017 IEEE.

TABLE I
PARAMETER SETTINGS OF FSRCNN WITH RESIZE FACTOR $s = 3$

Network	Model	Feature Extraction	Shrinkage	Non-Linear Mapping	Expanding	Reconstruction	Receptive Field
FSRCNN	$(56, 12, 4)$	$(1 \times 5 \times 5 \times 56)$ 1400	$(56 \times 1 \times 1 \times 12)$ 672	$(12 \times 3 \times 3 \times 12) \times 4$ 5184	$(12 \times 1 \times 1 \times 56)$ 672	$(56 \times 9 \times 9 \times 1)$ 4536	15×15
FSRCNN-s	$(32, 5, 1)$	$(1 \times 5 \times 5 \times 32)$ 800	$(32 \times 1 \times 1 \times 5)$ 160	$(5 \times 3 \times 3 \times 5) \times 1$ 225	$(5 \times 1 \times 1 \times 32)$ 160	$(32 \times 9 \times 9 \times 1)$ 2592	9×9

TABLE II
PARAMETER SETTINGS OF GLNET WITH RESIZE FACTOR $s = 3$

Network	Model	Feature Extraction	Shrinkage	Non-Linear Mapping	Expanding	Reconstruction	Receptive Field
GLNet	$(48, 10, 48, 8)$	$(1 \times 3 \times 3 \times 48)$ 432	$(48 \times 1 \times 1 \times 10)$ 480	$(10 \times 3 \times 3 \times 10) \times 8$ 7200	$(10 \times 1 \times 1 \times 48)$ 480	$(48 \times 9 \times 9 \times 1)$ 3888	29×29
GLNet-s	$(32, 5, 32, 3)$	$(1 \times 3 \times 3 \times 32)$ 288	$(32 \times 1 \times 1 \times 5)$ 160	$(5 \times 3 \times 3 \times 5) \times 3$ 675	$(5 \times 1 \times 1 \times 32)$ 160	$(32 \times 9 \times 9 \times 1)$ 2592	17×17



(a) Dilated factor: 1 (b) Dilated factor: 2

Fig. 2. (a) Normal convolution. (b) Dilated convolution. The dilated convolution skips some pixels to weight farther pixels. Dilated factor denotes the number to skip in the convolution.

convolution proposed in [14].

*f) **Local Queue Jumping Path**:* Stacking convolutions enlarge the receptive field of the network effectively, however, the modeling capacity to describe the local signals may be reduced. In our GLNet, to better model local signal structure, we add a local queue jumping path that forwards the first feature map to the last one (before reconstruction), as shown in the bottom part of Fig. 1.

g) Residual Learning: Following [11], our GLNet predicts the difference value between the HR and LR images instead of directly predicting the HR image.

E. Implementation Details

For input RGB image, we first perform a color-space transformation to transform the image to YCbCr color space, where Y contains luminance information of the image, Cb and Cr are the blue-difference and red-difference chroma components, respectively. In practice, we only exert super-resolution reconstruction on Y component. For Cb and Cr we simply use bicubic interpolation to resize. For this reason, we have $c_1 = 1$ which is the number of channels of the input images. To determine f_i and n_i for $i = 1, 2, 3$ as well as m, we can first refer to the setting of FSRCNN. It has two common settings of $(56, 12, 4)$ and $(32, 5, 1)$ for f_i, n_i and m. We later denote the setting of $(32, 5, 1)$ as FSRCNN-s. Suppose we have a resize factor of $s = 3$. Table I and II show the configuration and number of parameters to optimize for each component of the network.

For a given training set $\{\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\}_{i=1}^{N}$, we adopt the Mean Squared Error (MSE) as the loss function:

$$\mathcal{L}(\{W\}, \{B\}) = \frac{1}{N} \sum_{i=1}^{N} \|F(\mathbf{X}^{(i)}; \{W\}, \{B\}) - \mathbf{Y}^{(i)}\|_2^2, \quad (2)$$

where $\{W\}, \{B\}$ stand for the filter kernels and bias in the convolution layers, respectively. During the training process, $\{W\}, \{B\}$ are parameters to be optimized.

TABLE III
COMPARISON OF PSNR (DB) AND TIME (S) ON $Set5$ AND $Set14$. SF DENOTES THE SCALING FACTOR. THE BEST AND SECOND BEST RESULTS ARE DENOTED IN BOLD AND WITH UNDERLINE, RESPECTIVELY.

	Factor	PSNR/Time	PSNR/Time	PSNR/Time
Dataset	SF	Bicubic	RAISR	FSRCNN-s
$Set5$	×2	33.66/-	36.15/0.018	36.53/0.057
	×3	30.39/-	32.21/0.015	32.55/0.025
	×4	28.43/-	29.84/0.017	30.04/0.016
$Set14$	×2	30.23/-	32.13/0.034	32.22/0.102
	×3	27.54/-	28.86/0/029	29.08/0.042
	×4	26.00/-	27.00/0.030	27.12/0.023
Dataset	SF	FSRCNN	GLNet-s	GLNet
$Set5$	×2	<u>36.94</u>/0.117	36.79/0.068	**37.25**/0.182
	×3	<u>33.06</u>/0.053	32.82/0.031	**33.21**/0.070
	×4	<u>30.55</u>/0.036	30.40/0.019	**30.71**/0.044
$Set14$	×2	<u>32.54</u>/0.230	32.46/0.127	**32.76**/0.345
	×3	<u>29.37</u>/0.098	29.27/0.056	**29.47**/0.144
	×4	<u>27.50</u>/0.064	27.40/0.029	**27.58**/0.090

III. EXPERIMENTAL RESULTS

A. Dataset

Following the experimental setting in [9] and [16], we compare the proposed method with recent SR methods on three popular benchmark datasets: Set5 [17], Set14 [18] and BSD100 [19] with scaling factors of 2, 3 and 4. The three datasets contain 5, 14 and 100 images respectively. Among them, the Set5 and Set14 datasets are commonly used for evaluating traditional image processing methods, and the BSD100 dataset contains 100 images with diverse natural scenes. We train our model using a training set including 91 images in [20] and additional 100 images in FSRCNN [15] used for training.

B. Data Synthesis

To achieve better performance, we augment the training data in two steps. We first resize each image in the training set with factor in $[0.6, 0.7, 0.8, 0.9]$ respectively which generate 5 times the number of images from the original dataset. We rotate each image by degrees of 90, 180 and 270. With the two steps, we now have 20 times the number of images, the augmented data will be used as training data in the practice of training.

To generate the actual training data from the dataset described above, we first downsample every image in the dataset with factor s, and we crop an image block $\mathbf{X}$ with size $f_{sub} \times f_{sub}$ from LR image and respectively $\mathbf{Y}$ with size $sf_{sub} \times sf_{sub}$ from HR image. The pairs $\{\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\}_{i=1}^{N}$ form the training set. f_{sub} is determined by the size of the receptive field of the model. For GLNet$(32, 5, 32, 3)$ we have receptive field of size 29×29, so we set $f_{sub} = 29$ accordingly. We set $k = \lfloor f_{sub}/2 \rfloor$ so that most image will be sampled 4 times on average.

C. Implementation Settings

To make comparisons with existing methods fairly, we first train the model on the 91-image set and fine-tune the model on General-

TABLE IV

COMPARISON OF PSNR AND SSIM ON $Set5$, $Set14$ AND $B100$. SF DENOTES THE SCALING FACTOR. THE BEST AND SECOND BEST RESULTS ARE DENOTED IN BOLD AND WITH UNDERLINE, RESPECTIVELY. GLNET-L DENOTES GLNET$(96, 36, 36, 8)$.

	Factor	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Dataset	SF	Bicubic	A+	SRCNN
$Set5$	×2	33.66/0.930	36.54/0.954	36.66/0.954
	×3	30.39/0.930	32.58/0.909	32.75/0.909
	×4	28.42/0.810	30.28/0.860	30.48/0.863
$Set14$	×2	30.23/0.869	32.81/0.906	32.42/0.906
	×3	27.54/0.774	29.13/0.819	29.28/0.821
	×4	26.00/0.702	27.32/0.749	27.49/0.750
$B100$	×2	29.56/0.668	31.21/0.886	31.36/0.888
	×3	27.21/0.739	28.29/0.783	28.41/0.786
	×4	25.96/0.668	26.82/0.718	26.90/0.710
Dataset	SF	VDSR	GLNet	GLNet-l
$Set5$	×2	**37.53**/0.959	37.25/0.957	37.40/0.958
	×3	**33.66**/0.921	33.30/0.917	33.54/0.920
	×4	**31.35**/0.884	30.85/0.871	31.08/0.879
$Set14$	×2	**33.03**/0.912	32.81/0.910	32.95/0.911
	×3	**29.77**/0.831	29.51/0.827	29.65/0.830
	×4	**28.02**/0.767	27.67/0.758	27.92/0.762
$B100$	×2	**31.90**/0.896	31.58/0.891	31.76/0.893
	×3	**28.82**/0.798	28.54/0.791	28.73/0.796
	×4	**27.29**/0.725	26.98/0.715	27.14/0.721

100 dataset. We set the batch size as 64. For the initial training, we use Adam [21] with the learning rate of 0.0001, the momentum of 0.9 and weight decay of 0.0001. For the fine-tuning, we use Stochastic Gradient Descent [22] with learning rate of 0.0005. For both processes, we use Parametric Rectify Linear Units [23] as activation function. Hardware and software environment: Ubuntu 14.04, Intel i7-5930K@3.5GHz, GTX 1080 GPU.

D. Objective and Subjective Comparisons

We compare our proposed GLNet with RAISR [24] and FSRCNN [15], which take time consumption into consideration. All the models are trained on the 91 images. Testing phase runs on pure CPU. Results can be found in Tab. III. Our model can provide impressive reconstruction quality while we well control the time consumption, which achieves real-time level performance.

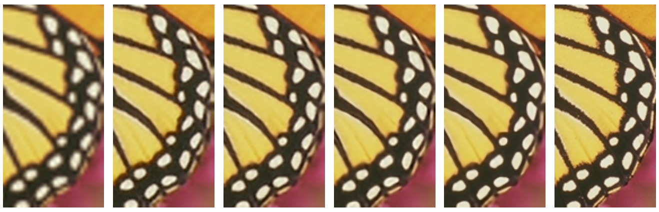


Fig. 3. Visual comparison between different algorithms. From left to right: Bicubic, FSRCNN-s, Ours$((48, 10, 48, 8))$, Ours$((96, 36, 36, 8))$, VDSR, Ground truth.

We also test our model on several different datasets for reconstruction quality, as is shown in Tab. IV. Our method achieves superior performance to most other SR methods. We also prove the scalability of our model by training a GLNet$(96, 36, 36, 8)$ which has a greater number of parameters and compare the quality with state-of-the-art methods. It provides comparable results with state-of-the-art methods and consumes less time for reconstruction. The visual quality illustration is presented in Fig. 3. We can see that GLNet achieves superior visual quality to state-of-the-art method VDSR.

IV. CONCLUSIONS

This paper introduces an efficient and effective GLNet for real-time image super-resolution. The dilated convolutions and additional local queue jumping path are employed to achieve satisfactory SR reconstruction quality while keeping the time consumption low. Our model is also highly scalable so that it can handle different quality/latency requirements. Experiments show that our model runs faster to achieve the equal quality as other state-of-the-art methods and outperforms other models under the equal latency constraints.

REFERENCES

[1] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, 2002.

[2] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *CVPR*, vol. 1, 2004, pp. I–I.

[3] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[4] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *ACCV*. Springer, 2014, pp. 111–126.

[5] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy *et al.*, “Deepid-net: Deformable deep convolutional neural networks for object detection,” in *CVPR*, 2015, pp. 2403–2412.

[6] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, “Part-based r-cnns for fine-grained category detection,” in *ECCV*. Springer, 2014, pp. 834–849.

[7] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in *CVPR*, 2014, pp. 1891–1898.

[8] W. Ouyang and X. Wang, “Joint deep learning for pedestrian detection,” in *ICCV*, 2013, pp. 2056–2063.

[9] C. Dong, C. Chen, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *ECCV*, 2014, pp. 184–199.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[11] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *CVPR*, 2016, pp. 1646–1654.

[12] J. K. L. J. Kim and K. M. Lee, “Deeply-recursive convolutional network for image super-resolution,” in *CVPR*, 2016, pp. 1637–1645.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.

[14] W. Shi, J. Caballero, F. Huszr, 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 *CVPR*, 2016, pp. 1874–1883.

[15] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” in *ECCV*, 2016, pp. 391–407.

[16] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” in *ICCV*, 2015, pp. 370–378.

[17] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, “Low complexity single-image super-resolution based on nonnegative neighbor embedding,” in *BMVC*, 2012, pp. 1–10.

[18] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Proc. IEEE Int’l Conf. Curves and Surfaces*. Springer, 2010, pp. 711–730.

[19] 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 *ICCV*, vol. 2. IEEE, 2001, pp. 416–423.

[20] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

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

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *ICCV*, 2015, pp. 1026–1034.

[24] Y. Romano, J. Isidoro, and P. Milanfar, “Raisr: rapid and accurate image super resolution,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 110–125, 2017.