Computer Methods in Medicine and Health Care
D. Bhattacharjee (Ed.)

doi:10.3233/ATDE210239

Dual Features Extraction Network for Image Super-Resolution

Guosheng ZHAO* and Kun WANG*[1]
School of Electronic Information, Wuhan University, Wuhan 430072, China

Abstract. With the development of deep convolutional neural network, recent research on single image super-resolution (SISR) has achieved great achievements. In particular, the networks, which fully utilize features, achieve a better performance. In this paper, we propose an image super-resolution dual features extraction network (SRDFN). Our method uses the dual features extraction blocks (DFBs) to extract and combine low-resolution features, with less noise but less detail, and high-resolution features, with more detail but more noise. The output of DFB contains the advantages of low- and high-resolution features, with more detail and less noise. Moreover, due to that the number of DFB and channels can be set by weighting accuracy against size of model, SRDFN can be designed according to actual situation. The experimental results demonstrate that the proposed SRDFN performs well in comparison with the state-of-the-art methods.

Keywords. Super-resolution, dual features, convolutional network, deep learning

1. Introduction

SISR aims to recover a high-resolution (HR) image from its low-resolution (LR) image. It has a wide range of applications in real life, such as monitoring [1], medical image processing [2], as well as in the image compression and transmission [3].

Since Dong et al. [4] firstly introduced the Convolutional Neural Network (CNN) to the super-resolution (SR), its superior performance highlights that leaning-based methods have more advantages than traditional methods such as interpolation-based methods [5-7] and reconstruction-based methods [8, 9]. Then the learning-based method [10-12] generally turns into mainstream on SISR. Current deep learning methods for SR can be divided into two main frameworks based on employed upsampling operations and their locations in the model: pre-upsampling and post-upsampling framework [13]. They are shown in figure 1.

LR
Upsample
CNN
HR
LR
CNN
Upsample
HR

(a) pre-upsampling framework (b) post-upsampling framework

Figure 1. Pre-upsampling and post-upsamping framework.

[1] Corresponding Author, Kun WANG, School of Electronic Information, Wuhan University, Wuhan 430072, China; Email: Chinawhuwangkun@163.com.

*The authors of * notes are the first authors.

Pre-upsampling framework was first adopted by [4]. For this framework, the operation of upsample is at the front of the CNN. Therefore, the network can not directly extract features from LR images. References [14-16], which adopt the pre-upsampling framework, utilize an interpolated image as the input. Therefore, these networks need to extract HR features from the interpolated image. In fact, these networks learn a nonlinear mapping between interpolated image and HR image. This framework reduces the difficulty of learning by avoiding learning the upsampling operation. However, the operation of interpolating will inevitably amplify the noise and introduce blurring. To solve this problem, the post-upsampling framework was proposed. In contrast, references [17-20] need to learn the upsampling operation at the end of their networks and the input of these networks is a LR image. These networks can directly extract LR features from the input image for image super-resolution. Due to the main calculation cost of this framework is on low-dimensional space, the computation complexity is greatly reduced.

To utilize the advantages of both, we propose a novel network for image SR, namely the super-resolution dual features extraction network (SRDFN). The proposed SRDFN utilizes the dual features extraction block (DFB) to extract the LR and HR features and combine the advantages of both. The experimental results indicate the advantages of SRDFN against other state-of-the-art methods.

In summary, the main contributions of our work are as follows:

- Proposed DFB not only enriches the feature maps via up-and-down sampling layers, but also combines the LR and HR features. The DFB utilizes low-resolution feature extraction path (LRFP) to extract LR features and high-resolution feature extraction path (HRFP) to extract HR features. Then using a 1 × 1 convolutional layer [21] to combine LR and HR features. The output of DFB contains the advantages of both features.
- The size of our network SRDFN is changeable, which means the network can be a trade-off between size and accuracy. Therefore, the network can be customized by increasing or decreasing the amount of DFB and the number of channels according to the actual situation.

2. Related Works

Deep learning has shown its superior performance in SISR. Dong et al. [4] firstly introduced the CNN in SISR to learn a nonlinear mapping between LR and HR. Kim et al. [22] adopted the residual architecture to get the better SR results. However, these two networks use the interpolated images as the input of networks, which leads to heavier computation time and memory. Moreover, the interpolated images amplify the noise from the LR images and bring the noise to the SR results. Rather than using an interpolated image as an input, EDSR [17] utilizes a LR image as an input. This network improve the computational efficiency by making the most cost of computation on the low-dimension space. Using the LR image as the input, there is no noise introduced to the networks. Therefore, their networks get the better SR results. However, they only use the LR or HR features to implement SR. Ke et al. [23] and Nakarmi et al. [24] make use of multi-scale feature extraction from LR and HR images to improve the quality of reconstructed images. To utilize the relationships of LR and HR features, DBPN [25] and SRFBN [26] use the iterative up-and-down sampling to fully address the mutual dependencies of LR and HR images. These networks utilize deep concatenation of the HR features from all up-sampling stages to reconstruct the HR images. DBPN [25] and

SRFBN [26] improve the SR results by exploiting the sufficient features and the relationships of LR and HR features. They focus on how to extract HR features and project them back to the LR spaces. Unlike [25, 26], we pay more attention to combine the LR and HR features. We propose SRDFN to mine the deep relationships of LR and HR features and combine the LR and HR features.

3. Image Super-resolution Based on Dual Features Extraction Network

3.1. Image Super-resolution Dual Features Extraction Network

As shown in figure 2, SRDFN includes an LR feature extraction block (LRFB), N dual features extraction blocks (DFBs), and a reconstruction block (RB). And there is a global residual skip connection from an upsampled image to the output of the RB. Thus, the output of the RB should be a residual image. In order to read this paper clearly, we summarize some acronyms in table 1.

Firstly, the LRFB uses a 3 × 3 convolutional layer with 4 × m filters and a 1 × 1 convolutional layer with m filters to extend the LR features from a LR image. Suppose an LR input I_{LR} for the LRFB, then the input of the first DFB D_{in}^{1}, also the output of the LRFB, can be obtained by:

$$D_{in}^{1} = f_{LRFB}(I_{LR}) \tag{1}$$

where f_{LRFB} refers to extract features from I_{LR}.

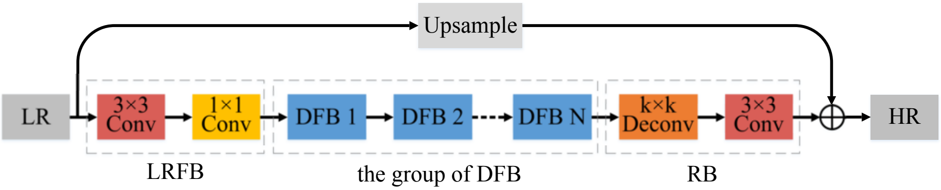

Figure 2. Overall architecture of SRDFN.

Table 1. The summary of acronyms.

Acronym	Description
LRFB	low-resolution feature extraction block
DFB	dual features extraction block
RB	reconstruction block
LRFP	low-resolution feature extraction path
HRFP	high-resolution feature extraction path

Then the DFB extracts and combines the LR and HR features. The input of the n-th DFB can be denoted as D_{in}^{1}, the output of the n-th DFB can be denoted as D_{out}^{n}. Then the mathematical formulation of DFB is:

$$D_{out}^{n} = f_{DFB}(D_{in}^{n}) \tag{2}$$

where f_{DFB} represents to the operations of the DFB. The specific details are in Section 3.2.

After that, the RB uses the features extracted and combined by the DFB to reconstruct a residual image. The RB consists of a $k \times k$ deconvolutional layer that has m filters and a 3 × 3 convolutional layer that has 3 filters. Then the output of RB R_{out} is obtained by:

$$R_{out} = f_{RB}(D_{out}^{N}) \tag{3}$$

where D_{out}^{N} refers to the output of the N-th DFB and f_{RB} denotes the operations of RB.

In the end, we can obtain the output image I_{SR}:

$$I_{SR} = R_{out} + f_{UP}(I_{LR}) \tag{4}$$

where f_{UP} represents to the operations of upsampling.

3.2. Dual Features Extraction Block

As shown in figure 3, DFB is constructed by LR feature extraction path (LRFP) and HR feature extraction path (HRFP). The DFB receives the information D_{in}^{n} to extract the LR features and HR features, and then combine two kinds of features as the input of the next block to extract more powerful information for reconstruction. Due to the input of LRFP and HRFP is both the D_{in}^{n}, we can consider the LRFP and HRFP are parallel.

The LRFP uses a 3 × 3 convolutional layer with m filters. The LR features of the n-th DFB L_n can be obtained by

$$L_n = f_{LRFP}(D_{in}^{n}) \tag{5}$$

where f_{LRFP} denotes the operations of the LRFP.

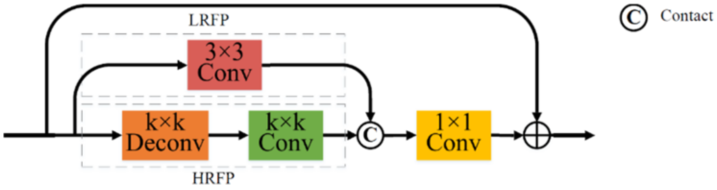

Figure 3. DFB.

Similar to LRFP, the HRFP consists of a $k \times k$ deconvolutional layer that has m filters and a 3 × 3 convolutional layer that has m filters. The HR features of the n-th DFB H_n can be obtained by:

$$H_n = f_{\text{HRFP}}(D_{in}^{n}) \tag{6}$$

where f_{HRFP} denotes the operations of the HRFP.

The L_n and H_n are contacted as the input of a 1 × 1 convolutional layer with m filters. The operation of contact is to place H_n after L_n. Then the output of n-th DFB D_{out}^{n} is obtained by

$$D_{out}^{n} = f_C(|L_n, H_n|) \tag{7}$$

where f_{C} denotes the operations of combination using a 1 × 1 convolutional layer with m filters.

4. Experimental Results

4.1. Training Details

DIV2K [27] is selected as our training dataset. To fully exploit the training dataset, we use the method as [17] does. We set the patch size according to the scale factor. The definite values of them are listed in table 2. We used two metrics, peak signal-to-noise ratio (PSNR) and structure similarity index (SSIM) [28], to evaluate the performance of our networks. As for the testing dataset, we used four widely used benchmark datasets: Set5 [29], Set14 [30], B100 [31] and Manga109 [32]. Due to that people are more sensitive to the luminance component, we only evaluate them on the luminance channel.

We adopt the method in [33] to initial the parameters of our networks and use ADAM optimizer [34] to update the parameters with an initial learning rate 0.0001. Our experiments are on NVIDIA TITAN XP GPU with Pytorch [35] framework.

Table 2. The setting of patch size.

Scale	**×2**	**×3**	**×4**
Patch Size	60	50	40

4.2. Implementation Details

We select PReLU [33] as the activation function after all convolutional layers and deconvolutional layers apart from the convolutional layer of the RB.We use L1 loss as our loss function and bicubic as upsampling method. The value of m mentioned in Section 3 is set 32 for SRDFN-S, a lightweight network, and 64 for SRDFN. About the value of *N*, it is set 8 for SRDFN-S and 16 for SRDFN. As for the value of *k*, we use diverse *k* for different scale factors. For ×2 scale factor, we set *k* = 6. Then to satisfy the scale factor, the striding and padding are both 2. As for ×3 scale factor, we set *k* = 7. To enlarge 120 the image three times, the striding should be 3 and padding should be 2. And for ×4 scale factor, we set *k* = 8. Similarly, the striding is set 4 and the padding is set 2.

4.3. Comparison with State-of-the-Art Methods

We select two networks, SRDFN-S, a lightweight network, and SRDFN, a network with a large number of parameters, to compare with the state-of-the-art methods. To improve the performance of SRDFN, we determined to use the self-ensemble [17] method on SRDFN (denoted as SRDFN+). We consider the state-of-the-art methods in this experiment including SRCNN [4], VDSR [22], DRRN [14], MemNet [15], EDSR [17] and SRFBN [26].

As shown in figure 4, SRDFN-S attains the best results among the lightweight networks. For the large networks, SRDFN and SRDFN+ can achieve great results with fewer parameters. The parameters of our proposed SRDFN is only 12.5% of EDSR [17] for the scale factor ×2. And the quantitative valuation results of our proposed models (SRDFN-S, SRDFN, SRDFN+) are listed in table 3.

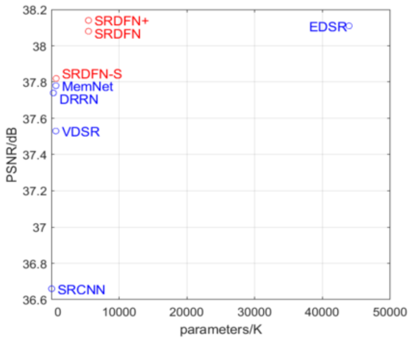

Figure 4. Performance on Set5 and number of parameters. Reds are our proposed network.

Table 3. Quantitative evaluation results comparing with the state-of-the-art methods on the standard benchmark datasets (PSNR(dB)/SSIM). The best performance is shown in red and the second-best performance is shown in blue.

Model	Scale	Set5	Set14	B100	Manga109
Bicubic	2	33.66/0.9299	30.24/0.8688	29.56/0.8431	30.30/0.9339
SRCNN [4]	2	36.66/0.9542	32.45/0.9067	31.36/0.8879	35.60/0.9663
VDSR [22]	2	37.53/0.9590	33.05/0.9130	31.90/0.8960	37.22/0.9750
DRRN [14]	2	37.74/0.9591	33.23/0.9136	32.05/0.8973	37.60/0.9736
MemNet [15]	2	37.78/0.9597	33.28/0.9142	32.08/0.8978	37.72/0.9740
SRDFN-S (ours)	2	37.82/0.9599	33.45/0.9163	32.04/0.8979	38.09/0.9759
EDSR [17]	2	38.11/0.9602	33.92/0.9195	32.32/0.9013	39.10/0.9773
SRFBN [26]	2	38.11/0.9609	33.82/0.9196	32.29/0.9010	39.08/0.9779
SRDFN (ours)	2	38.08/0.9607	33.79/0.9195	32.24/0.9004	38.66/0.9771
SRDFN+ (ours)	2	38.14/0.9610	33.89/0.9203	32.28/0.9009	38.87/0.9777
Bicubic	3	30.39/0.8682	27.55/0.7742	27.21/0.7385	26.95/0.8556
SRCNN [4]	3	32.75/0.9090	29.30/0.8215	28.41/0.7863	30.48/0.9117
VDSR [22]	3	33.67/0.9210	29.78/0.8320	28.83/0.7990	32.01/0.9340
DRRN [14]	3	34.03/0.9244	29.96/0.8349	28.95/0.8004	32.42/0.9359
MemNet [15]	3	34.09/0.9248	30.00/0.8350	28.96/0.8001	32.51/0.9369
SRDFN-S (ours)	3	34.28/0.9260	30.24/0.8405	29.01/0.8031	33.21/0.9422
EDSR [17]	3	34.65/0.9280	30.52/0.8462	29.25/0.8093	34.17/0.9476
SRFBN [26]	3	34.70/0.9292	30.51/0.8461	29.24/0.8084	34.18/0.9481
SRDFN (ours)	3	34.63/0.9289	30.51/0.8459	29.23/0.8087	33.81/0.9464
SRDFN+ (ours)	3	34.73/0.9296	30.59/0.8469	29.30/0.8090	34.09/0.9482
Bicubic	4	28.42/0.8104	26.00/0.7027	25.96/0.6675	24.89/0.7866
SRCNN [4]	4	30.48/0.8628	27.50/0.7513	26.90/0.7101	27.58/0.8555
VDSR [22]	4	31.35/0.8830	28.02/0.7680	27.29/0.7260	28.83/0.8870
DRRN [14]	4	31.68/0.8888	28.21/0.7721	27.38/0.7284	29.18/0.8914
MemNet [15]	4	31.74/0.8893	28.26/0.7723	27.40/0.7281	29.42/0.8942
SRDFN-S (ours)	4	32.06/0.8937	28.51/0.7802	27.51/0.7340	30.21/0.9052
EDSR [17]	4	32.46/0.8968	28.80/0.7876	27.71/0.7420	31.02/0.9148
SRFBN [26]	4	32.47/0.8983	28.81/0.7868	27.72/0.7409	31.15/0.9160
SRDFN (ours)	4	32.32/0.8969	28.76/0.7857	27.68/0.7397	30.82/0.9128
SRDFN+ (ours)	4	32.44/0.8981	28.84/0.7872	27.74/0.7410	31.06/0.9150

We show SR results in figure 5. We can see that our proposed network can produce clear images. Particularly in the SR results of the ‘ppt3’ from Set14, the SR results of SRDFN is very close to the ground truth. For the SR results of the ‘butterfly’ from Set5, SRCNN [4], VDSR [22], DRRN [14], and MemNet [15] fail to recover the shape of the image. However, the result of our SRDFN is more close to the ground truth. For the results of ‘302008’ from B100, the results of SRCNN [4], VDSR [22], and DRRN [14] is too blurry in the central part. The results of SRDFN and SRDFN-S are clearer than the result of MemNet [15]. As for the ‘GarakutayaManta’ from Manga109, SRDFN recovered the first word, however, other models failed to recover it.

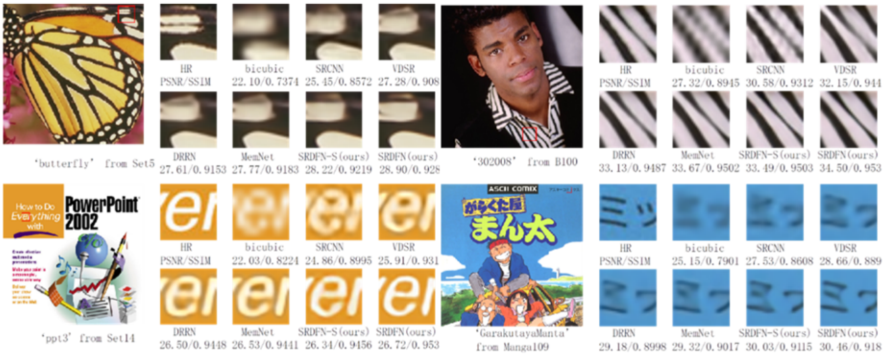

Figure 5. Visual results of our models with other works for scale factor ×4.

4.4. Network Analysis

4.4.1. The Efficiency of DFB

To highlight the effectiveness of DFB, we design two other networks to compare with SRDFN-S. One utilizes only LRFP to extract LR features and the other uses only HRFP to extract HR features. However, the channel factor m of both networks are set 32 and the factor N of both networks are set 8. From table 4, we can see that the model, using the DFB, gets the best PSNR. The results demonstrate that our proposed DFB can indeed combine the two kinds of features to improve the image quality.

Table 4. The PSNR/SSIM of SR results provided by model using LRFP, HRFP and DFB for scale ×2.

DataSet	LRFP	HRFP	DFB
Set5 [29]	37.38/0.9583	37.72/0.9596	37.82/0.9599
Set14 [30]	33.04/0.9124	33.37/0.9157	33.45/0.9163

We can also get this conclusion from figure 6. We can see that the SR result, provided by model, which use LRFP, is blurry at the high frequency part. This is caused by that LRFP only extracts LR features. In contrast, using HRFP can recover a sharper result than using LRFP. However, the result provided by the model, using HRFP, is still blurry, while DFB combines LR features and HR features to get a clearest result.

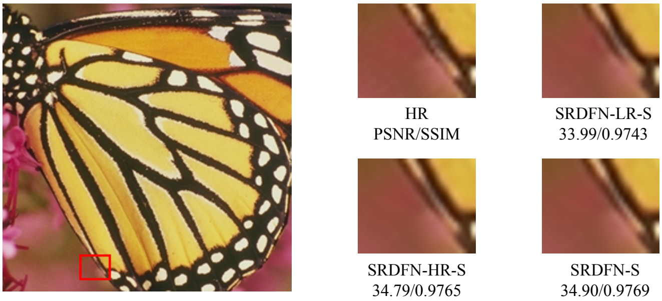

Figure 6. Visual results of LRFP, HRFP and DFB for scale ×2.

To further explain that how DFB to combine the two kinds of features, we save some features of DFB. LR_feature in figure 7 is the 1^{st} channel output of the LRFP in the 6^{th} DFB, HR_feature in figure 7 is the 1^{st} channel output of the HRFP in the 6^{th} DFB, and out_feature in figure 7 is the 1^{st} channel output of the 6^{th} DFB. We can observe from figure 7 that there are more details in HR_feature than in LR_feature, particularly in an eyelash. As we said in Section 2, there are some noises amplified in HR_feature, but in the meantime, the details are also amplified, which can improve the image quality. However, 158 LR_feature is smoother than HR_feature, because there is less noise in LR_feature. At last, we can see that there are more details than LR_feature and less noise than HR_feature in out_feature.

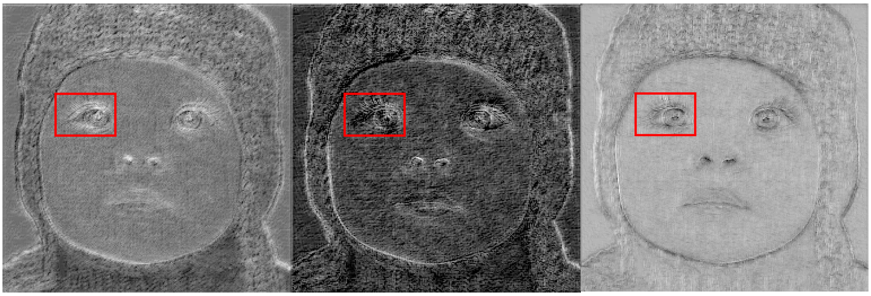

Figure 7. Features in the 6^{th} DFB of SRDFN-S.

4.4.2. The Trade-off of the Size and Accuracy

Our network can change the channel factor m or the factor N to determine the size of model. In this section, we set various values of them to generate 4 different networks. They are named according to SRDFN_N_m. For example, the channel factor m is 32 and the factor N is 8 for SRDFN_8_32. The convergence comparison of SRDFN_8_32, SRDFN_16_32, SRDFN_8_64, and SRDFN_16_64 is shown in figure 6. It can be observed from figure 8, that SRDFN_8_64 (the green one in figure 8) is higher than others in the early epoch. As the epoch increases, SRDFN_16_64 (the red one in figure 8) is generally catch up and pass the others. SRDFN_8_32 (the yellow one in figure 8) converges very quickly, but its result is lower than others. We can see that increasing m or N both improve the image quality, thus the DFB is efficient for image super-resolution.

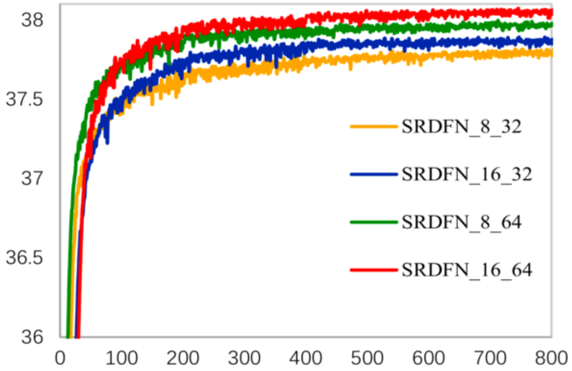

Figure 8. Convergence comparison with SRDFN_8_32, SRDFN_16_32, SRDFN_8_64, and SRDFN_16_64 on the Set5 for scale ×2.

It is obvious from table 5 that increasing channels is better than increasing the amount of DFB although both of them can improve performance. However, increasing channels will increase more parameters 172 than increasing the number of DFB. Thus, for different situations, we can design different networks by weighting the size of model and accuracy.

Table 5. Average PSNR values on Set5 for scale ×2.

Model	Parameters (M)	PSNR (dB)
SRDFN_8_32	0.69	37.82
SRDFN_16_32	1.34	37.89
SRDFN_8_64	2.76	38.00
SRDFN_16_64	5.36	38.08

5. Conclusion

In this paper, we propose a new method to realize super-resolution. By extract and combine the LR features and HR features, we get abundant features to reconstruct an HR image. HR features provide more details for reconstructing images, on the other hand, LR features are smoother than HR features. We utilize the advantages of both to reconstruct images and get a good performance. The DFB accomplishes the task excellently, enriching the feature maps and combining LR and HR features. Besides, our proposed network SRDFN can balance the size and accuracy according to the actual situation. Extensive experiments show that our proposed network could reconstruct images efficiently.

References

[1] Rasti, P.; Uiboupin, T.; Escalera, S.; Anbarjafari, G. Convolutional neural network super resolution for face recognition in surveillance monitoring. International conference on articulated motion and deformable objects. Springer, 2016, pp. 175–184.

[2] Greenspan, H. Super-resolution in medical imaging. The computer journal 2009, 52, 43–63.

[3] Park, S.C.; Park, M.K.; Kang, M.G. Super-resolution image reconstruction: a technical overview. IEEE signal processing magazine 2003, 20, 21–36.

[4] Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. European conference on computer vision. Springer, 2014, pp. 184–199.

[5] Zhang, L.; Wu, X. An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE transactions on Image Processing 2006, 15, 2226–2238.

[6] Keys, R. Cubic convolution interpolation for digital image processing. IEEE transactions on acoustics, speech, and signal processing 1981, 29, 1153–1160.

[7] Duchon, C.E. Lanczos filtering in one and two dimensions. Journal of applied meteorology 1979, 18, 1016–1022.

[8] Zhang, K.; Gao, X.; Tao, D.; Li, X. Single image super-resolution with non-local means and steering kernel regression. IEEE Transactions on Image Processing 2012, 21, 4544–4556.

[9] Yan, Q.; Xu, Y.; Yang, X.; Nguyen, T.Q. Single image superresolution based on gradient profile sharpness.IEEE Transactions on Image Processing 2015, 24, 3187–3202.

[10] Timofte, R.; De Smet, V.; Van Gool, L. Anchored neighborhood regression for fast example-based super-resolution. Proceedings of the IEEE international conference on computer vision, 2013, pp. 1920–1927.

[11] Peleg, T.; Elad, M. A statistical prediction model based on sparse representations for single image super-resolution. IEEE transactions on image processing 2014, 23, 2569–2582.

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

[13] Wang, Z.; Chen, J.; Hoi, S.C. Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020.

[14] Tai, Y.; Yang, J.; Liu, X. Image super-resolution via deep recursive residual network. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3147–3155.

[15] Tai, Y.; 220 Yang, J.; Liu, X.; Xu, C. Memnet: A persistent memory network for image restoration. Proceedings of the IEEE international conference on computer vision, 2017, pp. 4539–4547.

[16] Kim, J.; Kwon Lee, J.; Mu Lee, K. Deeply-recursive convolutional network for image super-resolution. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1637–1645.

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

[18] Tong, T.; Li, G.; Liu, X.; Gao, Q. Image super-resolution using dense skip connections. Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4799–4807.

[19] Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; others. Photo-realistic single image super-resolution using a generative adversarial network. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[20] Han, W.; Chang, S.; Liu, D.; Yu, M.; Witbrock, M.; Huang, T.S. Image super-resolution via dual-state recurrent networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1654–1663.

[21] Lin, M.; Chen, Q.; Yan, S. Network in network. arXiv preprint arXiv:1312.4400 2013.

[22] Kim, J.; Kwon Lee, J.; Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1646–1654.

[23] Ke, T.W.; Maire, M.; Yu, S.X. Neural multigrid. arXiv preprint arXiv:1611.07661 2016.

[24] Nakarmi, U.; Cheng, J.Y.; Rios, E.P.; Mardani, M.; Pauly, J.M.; Ying, L.; Vasanawala, S.S. Multi-scale Unrolled Deep Learning Framework for Accelerated Magnetic Resonance Imaging. 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, 2020, pp. 1056–1059.

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

[26] Li, Z.; Yang, J.; Liu, Z.; Yang, X.; Jeon, G.;Wu,W. Feedback network for image super-resolution. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3867–3876.

[27] Agustsson, E.; Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. Proceedings of the IEEE Conference on Computer Vision and Pattern RecognitionWorkshops, 2017, pp. 248 126–135.

[28] Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 2004, 13, 600–612.

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

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

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

[32] Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications 2017, 76, 21811–21838.

[33] He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision, 2015, pp. 262 1026–1034.

[34] Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 2014.

[35] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; others. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 2019, pp. 8026–8037.