2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Two-Stage Convolutional Network for Image Super-Resolution

Zheng Hui, Xiumei Wang, and Xinbo Gao
School of Electrical Engineering, Xidian University, Xi'an, China
Email: zheng_hui@aliyun.com, wangxm@xidian.edu.cn, xbgao@mail.xidian.edu.cn

***Abstract*—Deep convolutional neural networks (DCNN) have recently advanced the state-of-the-art on the issue of single image super-resolution (SR). In this work, we propose a two-stage convolutional network (TSCN) to estimate the desired high-resolution (HR) image from the corresponding low-resolution (LR) image. Specifically, we propose the multi-path information fusion (MIF) module that collects abundant information from feature maps of the input, output and intermediary in a module and distills primary information therein. Several cascaded MIF modules are used to progressively extract features desired by reconstruction and the output of each module is gathered for rebuilding the HR image. In addition, we introduce a refinement network with local residual topology architecture as the second stage so as to further restore the high-frequency details of HR image produced by the first stage. Due to less number of filters, the compact model achieves fast inference time and brings about state-of-the-art SR results on four benchmark datasets simultaneously. Code is available at https://github.com/Zheng222/TSCN.**

I. Introduction

Single image super-resolution (SR) is a classical computer vision problem, which aims to recover the visually pleasing high-resolution (HR) image from a low resolution (LR) image. Single image SR is widely used in computer vision applications such as medical image enhancement, remote sensing imaging, video surveillance, where important details and critical information are of great significance. Existing SR methods can be roughly classified into three categories: interpolation-based, reconstruction-based, and learning-based methods. Interpolation-based approaches, such as bilinear, bicubic [1] and nearest neighbor interpolation [2], typically employ interpolation kernels to estimate the unknown pixels. Although this type of methods runs very fast due to the low complexity of algorithm, it generates overly smoothed regions. Reconstruction-based SR methods usually utilize explicit prior information and introduce constraints to reconstruct HR images. Although this kind of methods is particularly effective to preserve geometric structure and to suppress ringing artifacts, it faces the shortcoming of losing high-frequency detail information especially in the case of large scaling factor. Learning-based or example-based methods try to learn the correspondence between LR feature space and HR feature space through a large number of representative example pairs, such as neighbor embedding [3], sparse coding [4] and random forest [5].

Recently, due to the strength of deep convolutional neural network (CNN), many CNN-based SR methods achieve prominent performance. Dong *et al.* [6] first exploit a three layer convolutional neural network, named SRCNN, to approximate the complex nonlinear mapping between the LR image and the HR counterpart. To reduce computational complexity, the authors propose a fast SRCNN (FSRCNN) [7], which adopts the transposed convolution to execute upscaling operation at the output layer. Kim *et al.* [8] present a very deep super-resolution network (VDSR) with residual architecture to achieve eminent SR performance, which utilizes broader contextual information with larger model capacity. Another model designed by Kim *et al.* [9] has a very deep recursive layer with skip-connection to ease the difficulty of training the model and improve the SR performance simultaneously. Tai *et al.* present a deep recursive residual network (DRRN) [11], which employs parameters sharing strategy to alleviate the requirement of enormous parameters of the very deep network.

Although the above CNN-based SR methods have achieved impressive performance, they still have deficiency to be addressed. The main limitation for the existing CNN-based methods is always trying to deepen the network with cascaded style to increase the capacity of model and then can improve the reconstruction accuracy, which leads to large computational cost and memory consumption. Therefore, to resolve this issue, in this paper, we propose a novel CNN-based single image SR approach, which introduces the multipath information fusion (MIF) module and two-stage reconstruction strategy to gradually reconstruct the desired HR images from the original LR input images. Since the number of filters in each convolutional layer is relatively few and the whole network is not particularly deep, the proposed model has advantage of less computational cost without sacrificing performance. As illustrated in Fig. 1, the MIF module fuses three types of features, which are original input feature, short-path feature and long-path feature respectively. In addition, to further enhance the quality of reconstruction, we introduce the refinement stage network that acts on the HR image generated by the first stage as shown in Fig. 2.

The main contributions of this paper can be summarized as follows:

- A two-stage convolutional network for single image SR is proposed to mitigate the problems of high computational cost and huge memory consumption. The proposed MIF module incorporates manifold information, which enhances the diversity of network topology structure and representational power.

978-1-5386-3787-6/18/$31.00 ©2018 IEEE

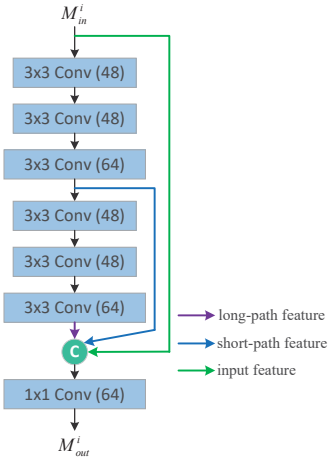

Fig. 1. Architecture of the MIF module. Here, green circle represents concatenation operation in channel dimension.

- We investigate two-stage reconstruction strategy, which is composed of rough SR stage and refinement stage. The latter experimental results suggest this approach is efficient in promoting the SR performance, especially in Urban100 [13] dataset.

II. Related work

Numerous methods have been proposed to tackle single image SR problem. In this section, we focus on recent CNN-based methods.

A. Convolutional network for image SR

Recently, many works have addressed the task of single image SR based on CNN. Dong *et al.* firstly propose the SRCNN, as depicted in Fig. 3(a). It implicitly learns a mapping between LR and HR images using a fully-convolutional network. Another network proposed by Dong *et al.* is fast SRCNN, namely FSRCNN (see Fig. 3(b)), which builds a compact hourglass-shape CNN structure for faster and better SR. Inspired by the success of very deep networks in recognition tasks [14], [15], Kim *et al.* proposed VDSR [8] with global residual architecture to achieve superior performance, which utilizes contextual information over larger image regions by increasing the network depth to 20 layers. The authors also design DRCN [9] that has recursive convolution with skip connection to avoid introducing additional parameters when the depth is increasing. Mao *et al.* [16] tackle the general image restoration problem through a very deep convolutional auto-encoder network and symmetric skip connections. Lai *et al.* propose LapSRN [10] to address the speed and accuracy of single image SR problem, which takes the original LR images as input and progressively reconstructs the sub-band residuals of HR images. Tai *et al.* [11] propose the deep recursive residual network to effectively build a very deep network structure for SR, which weighs the model parameters against the accuracy.

III. Proposed method

In this section, we describe the design methodology of the proposed convolutional network for single image SR.

A. Network architecture

As shown in Fig. 2, the proposed method adopts two-stage architecture. The first stage is a deep convolutional network which takes the original LR image as input and generates a relatively excellent HR image. The second stage is a small fully convolutional network that refines the prediction from the first network with more accurate results.

In the first stage, two 3×3 convolution layers with 64 filters are utilized to extract features from the LR image. Then we send these preliminary features to cascaded MIF modules with skip connections. In this way, intermediate features are aggregated by a concatenation operation. Considering the large number of feature maps generated by concatenation layer will significantly increase the computational cost, we introduce a convolution layer with 1×1 kernel that plays a role in reducing the number of input feature maps of the following transposed convolution layer. For both effectiveness and efficiency, the kernel size of transposed convolution are set to 4×4, 5×5, and 8×8 for upscaling factors of 2, 3, and 4, respectively. In the end of this stage, we apply a 3×3 convolution layer to produce the presentable HR image. Let's denote $\mathbf{y}$ and $\hat{\mathbf{y}}$ as the input and output of the first stage. Thus, the procedure of this subnetwork can be expressed as

$$\hat{\mathbf{y}} = S_1(\mathbf{y}), \tag{1}$$

where S_1 represents the first subnetwork.

The input to the second stage is the prediction from the first stage. This SR refinement stage is mainly composed of residual blocks. Each block contains two 3×3 convolutional layers, in which the former convolution has 8 groups. The purpose of doing so is to reduce the model parameters and computational cost, while enhancing the representational ability as in [27]. This stage can be simply formulated as

$$\hat{\mathbf{x}} = S_2(\hat{\mathbf{y}}), \tag{2}$$

where $\hat{\mathbf{x}}$ indicates the desired HR image, and S_2 denotes refinement subnetwork.

B. MIF module

We now present the details of our MIF module. The MIF module contains two 3-layer convolutional networks and a 1×1 convolutional layer. Each 3-layer network can be regarded as a basic unit, and the 1×1 convolutional layer is denoted as distillation unit that can be deemed to distill useful information from the preceding features. Specifically, the basic unit has three convolutions in which the second one is the grouped convolution with 4 groups. Each convolution has an activation function that is omitted in Fig. 1 to simplify the diagrammatic presentation. In addition, to slim the model size, the number of the convolutional filters is set to 48, 48, and 64, respectively. Suppose F^i, G^i represent the first three layers and sequential

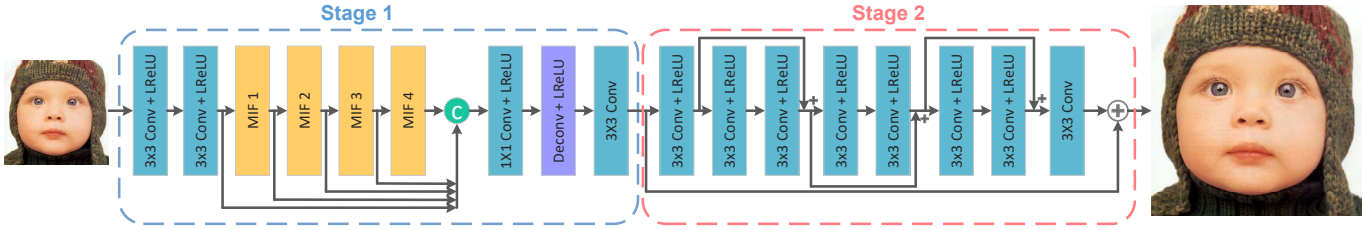

Fig. 2. Architecture of the proposed network.

(a) SRCNN (b) FSRCNN

Fig. 3. Network structures of SRCNN [6] and FSRCNN [7].

three layers of i-th MIF module respectively. Accordingly, the MIF module can be expressed as

$$M_{out}^{i} = D^{i}\left(\left[M_{in}^{i}, F^{i}\left(M_{in}^{i}\right), G^{i}\left(F^{i}\left(M_{in}^{i}\right)\right)\right]\right), \quad (3)$$

where D^i indicates the distillation unit of i-th MIF module, M_{in}^{i} and M_{out}^{i} are the input and output of the i-th MIF module, and $[\ldots]$ denotes the concatenation operation.

C. Loss function

Mean squared error (MSE) is the most widely used loss function for general image restoration. However, Lim *et al.* [17] apply mean absolute error (MAE) loss and obtain the better SR performance than MSE loss. We experimentally find that our model first employs MAE loss and then utilizes MSE loss can improve performance in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [18]. Therefore, we first train the network with MAE loss and then fine tune it using MSE loss.The MAE and the MSE are defined in Eq. 4 and Eq. 5.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - \hat{x}_i\right\|_1 \quad (4)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - \hat{x}_i\right\|_2^2 \quad (5)$$

Here N represents mini-batch size, x_i and $\hat{x}_i$ are the ground truth and the estimated value respectively for the i-th training sample.

IV. Experiments

A. Training datasets

For fair comparison with the other state-of-the-art methods, we also use 91 images from Yang *et al.* [19] and 200 images from BSD [20] as the training HR images. To make full use of this images, data augmentation is adopted in the proposed network training. Specifically, we augment these images in three ways: (1) Rotate each image with the degree of 90, 180, and 270. (2) Flip each image with horizontal. (3) Downscale each HR image by bicubic interpolation with the scaling factor of 0.9, 0.8, 0.7 and 0.6. Thus, we will obtain 39 additional augmented versions. For preparing the training samples, we first downsample the original HR training images with scaling factor $m\,(m = 2, 3, 4)$ by bicubic interpolation to generate the corresponding LR images and then crop the obtained LR images into a set of $s \times s$ size sub-images. In like manner, the HR images are cropped into $ms \times ms$ size sub-images. For $\times 2$, $\times 3$, and $\times 4$ models, the size of LR/HR image patches are set as $35^2/70^2$, $25^2/75^2$ and $19^2/76^2$, respectively.

B. Implementation details

According to the experiment, the leaky ReLU (LReLU) with negative scope 0.05 acts as the activation function. With regard to the initialization of network, we adopt the method proposed in He *et al.* [21]. The proposed SR method is trained in *Caffe* [22] package with Adam [23] optimizer. The parameters of mini-batch size and weight decay are set to be 64 and $1e-4$ respectively. Learning rate is set to be $1e-4$ and decreases by the factor of 10 after 2×10^5 iterations. We first update the first stage network without the refinement network. Through roughly 5×10^5 iterations, our first stage model is converged, we freeze its parameters and then update the refinement network. This process uses the same training method as the first stage.

C. Comparisons with state-of-the-arts

We compare the proposed method with other SR methods, including bicubic, FSRCNN [7], VDSR [8], DRCN [9], LapSRN [10] and DRRN [11] for different scaling factors ($\times 2$, $\times 3$ and $\times 4$) on four representative datasets Set5 [24], Set14 [25], BSD100 [20] and Urban100 [13]. Tab. I shows the average PSNR and SSIM values on four benchmark datasets with different scaling factors. From this table, we can see that the proposed method performs favorably against

TABLE I

AVERAGE PSNR/SSIMS FOR SCALE $\times 2$, $\times 3$ AND $\times 4$. **RED** COLOR INDICATES THE BEST AND BLUE COLOR INDICATES THE SECOND BEST PERFORMANCE.

Dataset	Scale	Bicubic	FSRCNN [7]	VDSR [8]	DRCN [9]	LapSRN [10]	DRRN [11]	TSCN (Ours)
Set5	$\times 2$	33.66 / 0.9299	37.00 / 0.9558	37.53 / 0.9587	37.63 / 0.9588	37.52 / 0.9591	37.74 / 0.9591	**37.88 / 0.9602**
	$\times 3$	30.39 / 0.8682	33.16 / 0.9140	33.66 / 0.9213	33.82 / 0.9226	33.81 / 0.9220	34.03 / 0.9244	**34.18 / 0.9256**
	$\times 4$	28.42 / 0.8104	30.71 / 0.8657	31.35 / 0.8838	31.53 / 0.8854	31.54 / 0.8852	31.68 / 0.8888	**31.82 / 0.8907**
Set14	$\times 2$	30.24 / 0.8688	32.63 / 0.9088	33.03 / 0.9124	33.04 / 0.9118	32.99 / 0.9124	33.23 / 0.9136	**33.28 / 0.9147**
	$\times 3$	27.55 / 0.7742	29.43 / 0.8242	29.77 / 0.8314	29.76 / 0.8311	29.79 / 0.8325	29.96 / 0.8349	**29.99 / 0.8351**
	$\times 4$	26.00 / 0.7027	27.59 / 0.7535	28.01 / 0.7674	28.02 / 0.7670	28.09 / 0.7700	28.21 / 0.7721	**28.28 / 0.7734**
BSD100	$\times 2$	29.56 / 0.8431	31.50 / 0.8906	31.90 / 0.8960	31.85 / 0.8942	31.80 / 0.8952	32.05 / 0.8973	**32.09 / 0.8985**
	$\times 3$	27.21 / 0.7385	28.52 / 0.7893	28.82 / 0.7976	28.80 / 0.7963	28.82 / 0.7980	**28.95** / 0.8004	**28.95 / 0.8012**
	$\times 4$	25.96 / 0.6675	26.96 / 0.7128	27.29 / 0.7251	27.23 / 0.7233	27.32 / 0.7275	27.38 / 0.7284	**27.42 / 0.7301**
Urban100	$\times 2$	26.88 / 0.8403	29.85 / 0.9009	30.76 / 0.9140	30.75 / 0.9133	30.41 / 0.9103	31.23 / 0.9188	**31.29 / 0.9198**
	$\times 3$	24.46 / 0.7349	26.42 / 0.8064	27.14 / 0.8279	27.15 / 0.8276	27.07 / 0.8275	**27.53 / 0.8378**	27.46 / 0.8362
	$\times 4$	23.14 / 0.6577	24.60 / 0.7258	25.18 / 0.7524	25.14 / 0.7510	25.21 / 0.7562	**25.44** / 0.7638	**25.44 / 0.7644**

TABLE II

AVERAGE IFCS FOR SCALE $\times 2$, $\times 3$ AND $\times 4$. **RED** TEXT INDICATES THE BEST AND BLUE TEXT INDICATES THE SECOND BEST PERFORMANCE.

Dataset	Scale	Bicubic	FSRCNN [7]	VDSR [8]	DRCN [9]	LapSRN [10]	DRRN [11]	TSCN (Ours)
Set5	$\times 2$	6.083	8.047	8.580	8.783	9.010	8.670	**9.175**
	$\times 3$	3.580	4.964	5.203	5.336	5.194	5.394	**5.544**
	$\times 4$	2.329	2.986	3.542	3.543	3.559	3.700	**3.766**
Set14	$\times 2$	6.105	7.731	8.159	8.370	8.501	8.280	**8.729**
	$\times 3$	3.473	4.549	4.691	4.782	4.662	4.870	**4.970**
	$\times 4$	2.237	2.707	3.106	3.098	3.145	3.249	**3.286**
BSD100	$\times 2$	5.619	7.082	7.494	7.577	7.715	7.513	**7.871**
	$\times 3$	3.138	4.030	4.151	4.184	4.057	4.235	**4.350**
	$\times 4$	1.978	2.359	2.679	2.633	2.677	2.746	**2.792**
Urban100	$\times 2$	6.245	8.026	8.629	8.959	8.907	8.889	**9.442**
	$\times 3$	3.620	4.842	5.159	5.314	5.156	5.440	**5.559**
	$\times 4$	2.361	2.895	3.462	3.465	3.530	3.669	**3.715**

TABLE III

COMPARISON OF RESULTS IN TERMS OF PSNR / TIME ON FOUR BENCHMARK DATASETS USING ONE-STAGE AND TWO-STAGE NETWORKS.

Methods	Scale	Set5	Set14	BSD100	Urban100
		PSNR / TIME	PSNR / TIME	PSNR / TIME	PSNR / TIME
TSCN_I	$\times 2$	37.87 / 0.017	33.26 / 0.028	32.08 / 0.017	31.23 / 0.071
TSCN	$\times 2$	37.88 / 0.028	33.28 / 0.046	32.09 / 0.028	31.29 / 0.130
TSCN_I	$\times 3$	34.14 / 0.013	29.96 / 0.018	28.93 / 0.012	27.38 / 0.043
TSCN	$\times 3$	34.18 / 0.025	29.99 / 0.037	28.95 / 0.023	27.46 / 0.104
TSCN_I	$\times 4$	31.78 / 0.011	28.25 / 0.015	27.41 / 0.010	25.40 / 0.034
TSCN	$\times 4$	31.82 / 0.023	28.28 / 0.034	27.42 / 0.021	25.44 / 0.094

state-of-the-art performance on most datasets. In addition, we utilize information fidelity criterion (IFC) metric to measure all methods, as shown in Tab. II, our TSCN achieves the excellent performance on the all test dataset. To evaluate the visual quality of the proposed approach, we show generated HR images by the proposed and compared methods. From Fig. 4, the reconstructed HR image by TSCN has clearer edges than results produced by other SR methods. In Fig. 5, we can obviously see that the proposed TSCN gains correct contour while other methods have different degrees of fake information.

As for inference time, we utilize the public codes of compared methods to assess the runtime on the same platform with Ubuntu 16.04 operating system, Matlab 2017a, 4.2GHz Intel i7 CPU, 64 GB memory and Nvidia Titan X (Pascal) GPU. Fig. 6 shows trade-offs between the execution time and PSNR values by the proposed method and the compared methods on Set5 dataset for the scaling factor of 3. Moreover, we also measure the inference times and PSNR indices of the first stage network as depicted in Tab. III. Here, this network is denoted as TSCN_I. From this table, we can see that the SR results of TSCN are slightly higher than that of TSCN_I on Set5, Set14 and BSD100. On Urban100 dataset, the performance improvement is comparatively obvious. The first stage network is already to gain commendable results with fast inference speed. In order to promote its generalization ability, we add the reinforcement net that takes the good quality HR images as input and then approaches the real images as far as possible.

(a) GT (PSNR/SSIM/IFC)

(b) Bicubic (26.46dB/0.8318/2.271)

(c) FSRCNN (29.34dB/0.8886/3.025)

(d) VDSR (29.81dB/0.9047/3.663)

(e) DRCN (30.10dB/0.9064/3.668)

(f) LapSRN (30.32dB/0.9090/3.710)

(g) DRRN (30.36dB/0.9111/3.860)

(h) TSCN (**30.73dB/0.9147/3.957**)

Fig. 4. The "woman" image from Set5 dataset with an upscaling factor 4.

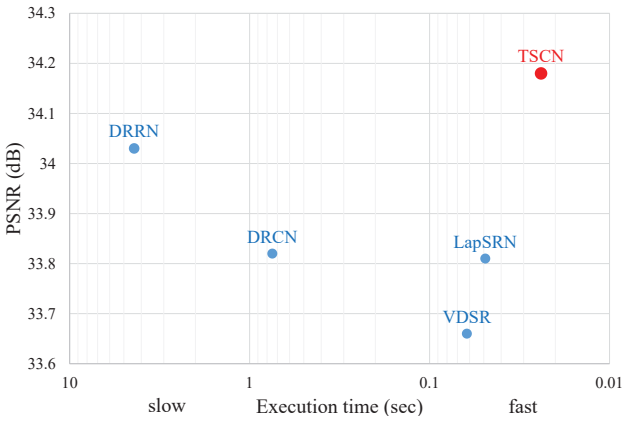

Fig. 6. Speed and accuracy trade-off. The average PSNR and the corresponding average time for upscaling $\times 3$ on Set5. The TSCN is faster than other methods and achieves the best performance at the same time.

V. Conclusion

In this paper, we propose a two-stage convolutional network for fast and accurate single image super-resolution. In the first stage, the vital portion is the proposed MIF module, which makes full use of multi-path information and enrich the representation of the network. As for the second stage that mainly further refines the HR image generated in the first stage to get better results on various testing datasets. The experiments show that our proposed model achieves state-of-the-art results in terms of PSNR and IFC. In the further, this effective approach for the image SR will be explored to other image restoration issues such as compression artifacts reduction.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61472304, 61432014 and U1605252.

(a) GT (PSNR/SSIM/IFC) (b) Bicubic (28.50dB/0.8285/2.638) (c) FSRCNN (29.81dB/0.8591/3.026) (d) VDSR (29.54dB/0.8651/3.388)

(e) DRCN (30.29dB/0.8653/3.323) (f) LapSRN (30.10dB/0.8710/3.432) (g) DRRN (29.77dB/0.8671/3.509) (h) TSCN (**30.56dB/0.8720/3.634**)

Fig. 5. The "8023" image from BSD100 dataset with an upscaling factor 4.

REFERENCES

[1] R. Keys, "Cubic convolution interpolating for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[2] R. Fattal, "Image up-sampling via imposed edge statistics," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 95–103, 2007.

[3] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceddings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[4] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation". *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

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

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution", in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 184–199.

[7] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network", in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 391–407.

[8] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks", in *roceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[9] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.

[10] 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*, 2017, pp. 624–632.

[11] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3147–3155.

[12] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration", in *Proceddings of the IEEE Conference on Computer Vision*, 2017, pp. 3147–3155.

[13] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5197–5206.

[14] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale recognition", in *Proceedings of the International Conference on Learning Representations*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[16] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections", in *Proceedings of the Conference on Neural Information Processing Systems*, 2016.

[17] B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, "Enhanced deep residual networks for single image super-resolution", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2017, pp. 136–144.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, vol. 13, no. 4, 2004, pp. 600–612.

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

[20] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001,pp. 416–423.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification", in *Proceddings of the IEEE Conference on Computer Vision*, 2015, pp. 1026–1034.

[22] Y. Jia, E. SHELHAMER, J. Donahue, S. Karayev, J. Long, R. Grishick, S. Guadarrama, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding", in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675–678.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *Proceedings of the International Conference on Learning Representations*, 2014.

[24] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding", in *Proceedings of the British Machine Vision Conference*, 2012.

[25] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations", in *Proceedings of the International Conference on Curves and Surfaces*, 2010, pp. 711–730.

[26] Z. Hui, X. Wang, and X. Gao, "Deep networks for single image super-resolution with mulit-context fusion", in *Proceedings of the International Conference on Image and Graphics*, 2017, pp. 397–407.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.