Collaborative Representation Cascade for Single-Image Super-Resolution

Yongbing Zhang, Yulun Zhang, *Student Member, IEEE*, Jian Zhang, *Member, IEEE*,
Dong Xu, *Senior Member, IEEE*, Yun Fu, *Senior Member, IEEE*,
Yisen Wang, Xiangyang Ji, *Member, IEEE*, and Qionghai Dai

***Abstract*—Most recent learning-based single-image super-resolution methods first interpolate the low-resolution (LR) input, from which overlapped LR features are then extracted to reconstruct their high-resolution (HR) counterparts and the final HR image. However, most of them neglect to take advantage of the intermediate recovered HR image to enhance image quality further. We conduct principal component analysis (PCA) to reduce LR feature dimension. Then we find that the number of principal components after conducting PCA in the LR feature space from the reconstructed images is larger than that from the interpolated images by using bicubic interpolation. Based on this observation, we present an unsophisticated yet effective framework named collaborative representation cascade (CRC) that learns multilayer mapping models between LR and HR feature pairs. In particular, we extract the features from the intermediate recovered image to upscale and enhance LR input progressively. In the learning phase, for each cascade layer, we use the intermediate recovered results and their original HR counterparts to learn single-layer mapping model. Then, we use this single-layer mapping model to super-resolve the original LR inputs. And the intermediate HR outputs are regarded as training inputs for the next cascade layer, until we obtain multilayer mapping models. In the reconstruction phase, we extract multiple sets of LR features from the LR image and intermediate recovered. Then, in each cascade layer, mapping model is utilized to pursue HR image. Our experiments on several commonly used image SR testing datasets show that our proposed CRC method achieves state-of-the-art image SR results, and CRC can also be served as a general image enhancement framework.**

***Index Terms*—Collaborative representation cascade (CRC), feature space, principal component analysis (PCA), super-resolution.**

I. Introduction

SINGLE-image super-resolution (SISR) plays a fundamental role in computer vision applications and tries to generates a visually pleasing high-resolution (HR) result from its degraded low-resolution (LR) measurement. Because, for any LR input, there exists a multitude of solutions [1], [2], image SR becomes to be an ill-posed procedure. In order to mitigate and tackle this inverse problem, researchers have proposed plenty of image SR algorithms. Among most of those SR algorithms, strong prior information is always used to constrain the image SR solution space. We can obtain such prior information by training from internal or external LR–HR image pairs, supposing a certain distribution for image feature space, and utilizing several regularization terms [3], [4]. As a result, plenty of image SR algorithms have been proposed and can be roughly classified into three categories: 1) interpolation-based methods [5]–[7]; 2) reconstruction-based methods [8]–[11]; and 3) example-based methods [12]–[33].

As a fundamental role in image SR family, interpolation-based algorithms can obtain zoom-in SR results very fast due to its simplicity. The image interpolation process often uses various types of kernels, such as fixed-function kernels [5] and adaptive-structured kernels [6], [7]. Despite being simple and fast in some real-time image SR cases, interpolation-based methods often over-smooth images, producing blurring artifacts and losing details. By assuming that the observed LR input can be generated with several degradation processes (e.g., blurring, adding noise, and down-scaling), reconstruction-based algorithms often introduce many types of reconstruction constraints. Along with a data-fidelity term, edge-directed priors [8]–[10] and similarity redundancy [11] are usually incorporated to model a regularized objective function. Nonetheless, reconstruction-based methods depend too much on the priors and often produce SR results with unnatural edges and obvious ringing artifacts.

Besides, by utilizing machine learning techniques, example-based image SR algorithms have achieved state-of-the-art

Manuscript received November 28, 2016; revised February 26, 2017; accepted May 9, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61571254, Grant U1301257, and Grant 61325003. This paper was recommended by Associate Editor Q. Wang. *(Corresponding author: Yulun Zhang.)*

Y. Zhang is with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: zhang.yongbing@sz.tsinghua.edu.cn).

Y. Zhang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zhangyl14@mails.tsinghua.edu.cn).

J. Zhang is with the Image and Video Understanding Laboratory, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia (e-mail: jian.zhang@kaust.edu.sa).

D. Xu is with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW 2006, Australia (e-mail: dong.xu@sydney.edu.au).

Y. Fu is with the Department of Electrical and Computer Engineering, College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA (e-mail: yunfu@ece.neu.edu).

Y. Wang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: wangys14@mails.tsinghua.edu.cn).

X. Ji and Q. Dai are with the TNLIST and the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: xyji@tsinghua.edu.cn; qhdai@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2705480

SR results. This type of methods would use internal likeness of the LR input [26], [29], or train the mapping relationships from LR to HR images with external LR and HR image pairs, including neighbor embedding (NE) [12], [13], [16], dictionary learning [14], [15], [19], [21], [31], regression [17], [18], [20], [22]–[24], [27], [28], [32], and deep learning [25], [33]. Chang *et al.* [12] presented the NE algorithm to generate HR patches with the usage of locally linear embedding (LLE) [34]. The LLE algorithm supposes that every image patch and its neighboring patches would lie on or near a locally linear sample of the manifold, if there has enough number of image patches. With the usage of nonnegative NE and a non-negativity constraint, Bevilacqua *et al.* [16] presented another NE image SR method by making a least squares approximation for LR inputs. Based on dictionary learning, Yang *et al.* [14], [35] introduced sparse coding for image SR. In the sparse coding-based method, an assumption was made that LR and HR features share the same sparse coefficients over their corresponding dictionaries. Similar method was proposed by Zeyde *et al.* [15], who used principal component analysis (PCA) to diminish the LR feature dimension and orthogonal matching pursuit [36] to obtain sparse coefficients. Similar mapping principle was used for high-quality image resizing [31]. In addition, more sophisticated dictionary learning strategies have been presented recently. He *et al.* [19] applied a beta process prior to train the LR and HR dictionaries. Based on sparse representation, Peleg and Elad [21] presented a statistical prediction image SR model. Fast image SR methods based on regression were also investigated. For each LR dictionary atom, anchored to which Timofte *et al.* [18], [20] learned regressors within an LR–HR dictionary pairs or the whole training features. Yang and Yang [37] clustered the training data spaces into numerous subspaces and learned simple functions for each subspace. Following a similar approach, Zhang *et al.* [38] proposed to learn manifold linear mapping in each LR–HR feature cluster pair. Dai *et al.* [27] jointly learned the separation into cells and regressors. Some methods were proposed to learn mappings between LR and HR patches with the usage of random forests [28], [39] or matrix-value operators [32]. By extending neighbor grouping to several clusters and using local geometry property, Zhang *et al.* [40] proposed cluster and collaborative representation (CCR) for fast image SR. In order to use reconstructed HR images from further image enhancement, Zhang *et al.* [41] also proposed iterative collaborative representation (ICR) for high-quality image SR. Similar technique was latterly included as one of seven ways to improve SR in Timofte *et al.*'s [42] work. Very recently, deep learning has shown its power in image SR by learning hierarchical representations of high-dimensional data. Under the notion of internal example-based method [43], Cui *et al.* [26] presented deep network cascade (DNC) to progressively upsample LR inputs layer by layer. Adopting external training data, Dong *et al.* [25] first presented a model named super-resolution convolutional neural network (SRCNN) to directly learn mappings from LR to its HR counterpart in an end-to-end way. Kim *et al.* [33] further increased the network depth by using residual learning and adjustable gradient clipping. Kim *et al.* [44] also used deeply recursive convolutional network for image SR with recursive-supervision and skip-connection. Although these types of algorithms have obtained wonderful SR results, most of them neglect to utilize the intermediate recovered results to enhance image quality further.

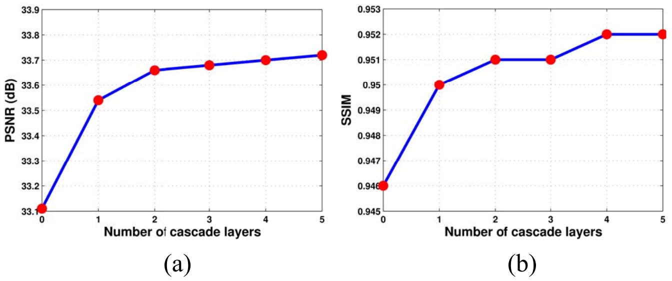


Fig. 1. PSNR (dB) and SSIM results of our CRC method using different number of cascade layers for the "*Starfish*" image ($s = 2$). The image SR result would be promoted by only one or two cascade layers. More details about the number of cascade layers in our CRC method will be discussed in Section IV-B. (a) PSNR (dB) outcomes using our CRC method. (b) SSIM outcomes using our CRC method.

More specifically speaking, most example-based algorithms interpolated the LR image to a desired size and used it as a start point of image SR, which can lose lots of details. By keeping 99.9% of the average energy, we applied PCA algorithm to reduce dimension largely in the LR feature space, whose dimension would be far less than that of HR feature. While, if the same type of LR features were extracted from the intermediate recovered or the real HR samples and the same PCA was applied, the dimension would become larger than that from the simply interpolated HR images. (More details will be discussed in Section II.) This observation encourage us to believe that the intermediate recovered results hold more helpful clues than the simply interpolated start point for further image SR and enhancement.

Motivated by the observation above and the fact that a few researches of utilizing the reconstructed results for more sophisticated image SR, in this paper, we present a simple but effective method named collaborative representation cascade (CRC). Figs. 1 and 2 show a comparison on an example quantitatively and visually. The main procedures of our CRC are illustrated in Fig. 4. We take advantage of the intermediate recovered results to progressively enhance image SR results. Our method is different from recent leading example-based approaches [12], [14]–[16], [18]–[21], [25]–[28]. We extract LR features from the intermediate recovered result to reconstruct a more sophisticated output, which can be used as input for the next cascade layer. In the learning stage, we learn multilayer mapping matrices between LR and HR feature spaces layer by layer. Consequently, when compared with the recent leading SR algorithms, our CRC obtains excellent SR results, while maintaining computation efficiency.

All in all, our main contributions of this paper can be summarized in three aspects.

1) We believe that the intermediate recovered SR result holds more helpful information than that of the simply interpolated image. To enhance image SR quality further, we try to build multilayer mapping relationships between

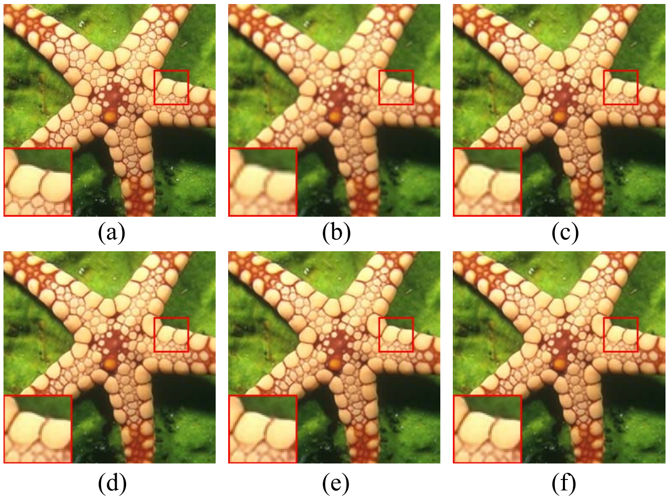


Fig. 2. Visual comparisons with different SR methods for the "*Starfish*" image ($s = 2$). "CRC-r" means r cascade layers are used in our CRC method. The proposed CRC outperforms the anchored neighborhood regression (ANR) [18] in terms of PSNR (zoom in for better view). (a) Original/PSNR. (b) Bicubic/30.28. (c) ANR [18]/32.31. (d) CRC-1/33.54. (e) CRC-3/33.68. (f) CRC-5/33.72.

the intermediate images and their corresponding original HR ones in each cascade layer.

2) We propose a CRC model for image SR. We extract LR features from the intermediate recovered SR results and train multilayer mapping models from these LR features to original HR features.
3) We conduct extensive experiments and show that CRC is a helpful and general image SR framework. With little pre/post-processing, CRC could obtain high-quality SR results fast-speed efficiently. This model can be used to further promote the performance of most recent main image SR methods.

A preceding conference version related to this paper was published in [41]. We extend [41] in the following aspects. First, in Section II, we provide the detailed motivations of our CRC image SR model, in which the analysis of PCA behavior for the extracted LR features is added in this version. Second, we conduct more experiments to show the effectiveness and robustness of our presented CRC framework. Specifically, we utilize Set10 [45]–[47], Set5 [16], Set14 [15], and SetSRF [28] to evaluate the performances when setting the upscaling factors as 2–4. More recently published state-of-the-art methods including DNC [26], JOR [27], and ARFL [28] are also included for the comparison. Third, we investigate the performance with respect to the number of iterations, as the running time can also increase with more number of iterations. Moreover, we extend bicubic, A+ [20], JOR [27], and our previous work [30] (here we refer it as ICR for simplicity) as Bicubic+, A++, JOR+, and ICR+, respectively.

We organize the rest of this paper as follows. The motivation of our method from the PCA in LR feature space is first shown in Section II. The details of our proposed CRC for image SR is presented in Section III. Abundant experiments and discussion are shown in Section IV. Finally, this paper is concluded in Section V.

TABLE I
DIMENSION OF LR FEATURE

Scale	Without PCA	With PCA based on LR features			
		Original HR images	Bicubic	A+ [20]	CRC-1
×2	144	71	28	54	57
×3	324	154	30	91	97
×4	576	264	31	113	134

II. MOTIVATIONS

In example-based SR methods, in order to save the running time and memory utilization, it is a common means to diminish the LR feature dimension by using PCA. For example, the majority of recent example-based SR algorithms [15], [18], [20], [27], [28], [30] first utilize Bibubic interpolation to interpolate LR image, whose resolution becomes the same as that of desired HR counterpart. Then PCA dimension reduction is conducted in the extracted LR features in the learning and reconstruction stages. This will reduce the LR feature dimension largely, which significantly speeds up the SR process while no obvious performance degradation can be observed. Actually, LR features can also be extracted from HR images generated by other methods, which may lead to varying performance. However, as far as we know, there has few works in image SR to investigate the effectiveness of PCA dimension reduction on the reconstructed images. To better describe the influence of extracted LR features from different images, we perform an analysis of PCA behavior over LR features extracted from reconstructed HR images by different methods in this section.

We use 91 images in [14] as our real HR images, from which we mimic LR images with the usage of bicubic interpolation. These LR images are up-scaled by different methods, bicubic interpolation, A+ [20], and the proposed CRC, to generate the HR outputs. Low-level feature was demonstrated to be effective in natural image processing [48], we extract LR features similar to Zeyde's methods in [15]. Then, on the basis of preserving 99.9% average energy, we conduct PCA dimension reduction on LR features extracted from these reconstructed and original HR images. We tabulate the LR feature dimension before and after PCA by different methods in Table I. We can learn that the original dimension (no PCA in Table I) of the LR features are very high (e.g., 144, 324, and 576 when scaling factor $s = 2$, 3, and 4, respectively). Taking $s = 3$ as an example (the patch size is 9×9), after conducting PCA, the dimension of LR features on the bicubic results drops to 30. While, if we extract the LR features from the real HR samples and conduct the same PCA to reduce feature dimension, which will drop to be 154, far larger than 30. In addition, we also find that the dimension of LR features extracted from the results by A+ [20] is 91, which can be further increased to 97 in the LR feature space from the results by CRC-1. Similar phenomena could be observed, when PCA procedure is conducted under $s = 2$ and 4. The reconstructed HR results contain more detailed textures and high frequency information, compared with HR images generated by bicubic interpolation. As a result, we may expect better performance if we extract LR features from intermediate recovered HR images with higher fidelity.

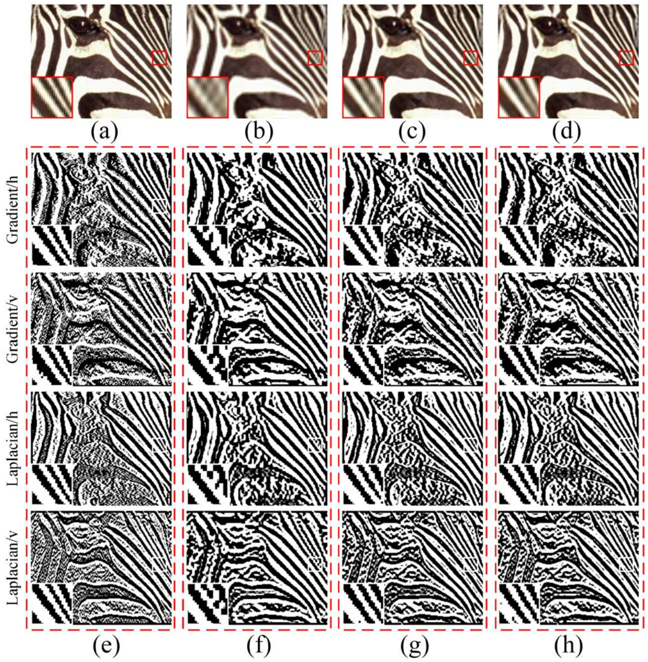


Fig. 3. Visual comparisons of original and different recovered SR results and their corresponding feature maps (scaling factor $s = 3$). Column 1: the original HR image and its feature maps. Column 2: HR image by bicubic interpolation and feature maps. Column 3: HR image by A+ [20] and feature maps. Column 4: HR image by CRC-1 and feature maps. (a) Original/PSNR. (b) Bicubic/22.70. (c) A+/28.70. (c) A+/28.01. (d) CRC-11/28.44. (e) Feature maps original. (f) Feature maps bicubic. (g) Feature maps A+. (h) Feature maps CRC-1.

At the same time, we select an HR image from the 91 images above and obtain its corresponding LR image. As presented in the first column of Fig. 3, LR feature maps are extracted from the original HR image. Several feature maps extracted from intermediate reconstructed images obtained by bicubic interpolation, A+ [20], and CRC-1, are shown from second to fourth columns of Fig. 3. From the original and reconstructed HR images, we can learn that extracting LR features from the reconstructed image is able to progressively improve the image SR results (e.g., PSNR value of CRC-1 is higher than that of A+ [20]). This can also be proved by the feature maps of intermediate recovered SR results by different algorithms. Feature maps from bicubic interpolated HR image appear to be crude and lose many details. By comparison, the feature maps from HR image by the recent leading SR method A+ [20] contain more details and structural information. However, the HR image by CRC-1 produce feature maps with more faithful details (e.g., the gradient feature maps in the horizontal direction in Fig. 3). These visual results are consistent with that in Table I.

The above observations indicate that the reconstructed HR images contain more useful information for image SR. And their corresponding feature maps possess more faithful details to recover more sophisticated HR features, which motivates us to conduct image SR in an iterative way, which is detailed in the following section.

III. Proposed Collaborative Representation Cascade for Super-Resolution

We first present image SR by using collaborative representation with unified form of mutual coherence. We then show how to learn multilayer mapping models in our CRC model. Finally, we provide details on how to conduct image SR via CRC.

In this paper, we denote $\boldsymbol{y}_{L/H}$, $\boldsymbol{p}_{L/H}$, and $\boldsymbol{I}_{L/H}$ as an LR/HR feature, patch, and image, respectively. Bold lowercase and uppercase letters mean vectors and matrices. Scalars are presented by plain lowercase letters.

A. Collaborative Representation for Image SR

For fast high-quality image SR, Timofte *et al.* [20] presented A+ by utilizing collaborative representation [49]. This paper was further improved in our previous work [30]. In the dictionary learning and neighbor grouping stages, we use mutual coherence as unified form of distance metrics. Mutual coherence is used to measure the distance between dictionary atoms and atoms/samples. On both training and reconstruction phases, we follow the same settings in our previous work [30] for each cascade layer in the CRC model. Detailed discussion would be given in Sections III-B and III-C.

First, we give a brief review of collaborative representation [49] based SR. From LR training images, we extract LR features, whose dimension are diminished with the usage of PCA projection matrix $\mathbf{P}$ to form LR feature set $\boldsymbol{Y}_{Ls} = \{\boldsymbol{y}_L^i\}_{i=1}^{N_s}$. Then we utilize LR features to train LR dictionary $\boldsymbol{D} = \{\boldsymbol{d}_k\}_{k=1}^{K}$ via

$$\min_{D,X} \|Y_{Ls} - DX\|_F^2 + \lambda_1 \sum_{i=1}^{K} \sum_{j=1,\, j\neq i}^{K} \left|d_i^{\mathrm{T}} d_j\right|$$
$$\text{s.t. } \forall k,\ \|d_k\|_2 = 1 \text{ and } \forall n,\ \|x_n\|_0 \leq L \tag{1}$$

where $\lambda_1 \sum_{i=1}^{K} \sum_{j=1,j\neq i}^{K} |\boldsymbol{d}_i^{\mathrm{T}} \boldsymbol{d}_j|$ is the mutual coherence between dictionary atoms to encourage large mutual incoherence of the learned dictionary, $\boldsymbol{X}$ is the coefficient matrix for $\boldsymbol{Y}_{Ls}$, and $\lambda_1 \geq 0$ is a weight parameter making a trade-off between mutual coherence and reconstruction error. We solve (1) via MI-KSVD proposed by Bo *et al.* [50].

Then, a larger LR–HR training feature set $\{\boldsymbol{y}_{Lw}^i, \boldsymbol{y}_{Hw}^i\}_{i=1}^{N_w}$ from the same training images is used for neighborhood sampling, similar as in A+ [20]. For every LR atom $\boldsymbol{d}_k$, we first compute the mutual coherence between each training LR feature and $\boldsymbol{d}_k$. Then, we search its neighbors $\boldsymbol{N}_{L,k}$ by selecting *max* nearest LR features. We use the their HR counterparts to form $\boldsymbol{N}_{H,k}$. For each LR–HR feature set $\{\boldsymbol{N}_{L,k}, \boldsymbol{N}_{H,k}\}_{k=1}^{K}$, we utilize collaborative representation [49] to compute mapping matrix. This l_2-norm regularized least squares regression problem can be stated as

$$x_i = \arg\ \min_{x_i} \left\|y_L^i - N_{L,k} x_i\right\|_2^2 + \lambda_2 \|x_i\|_2^2 \tag{2}$$

where $\boldsymbol{y}_L^i \in \mathbb{R}^n$ represents LR feature. $\boldsymbol{N}_{L,k}$ denotes the LR neighbors of LR atom $\boldsymbol{d}_k$. $\boldsymbol{x}_i$ is the coefficient of $\boldsymbol{y}_L^i$ over $\boldsymbol{N}_{L,k}$. $\lambda_2 > 0$ is a constant. A closed-form solution is available for the above problem as follows:

$$x_i = \left(N_{L,k}^T N_{L,k} + \lambda_2 I\right)^{-1} N_{L,k}^T y_L^i \tag{3}$$

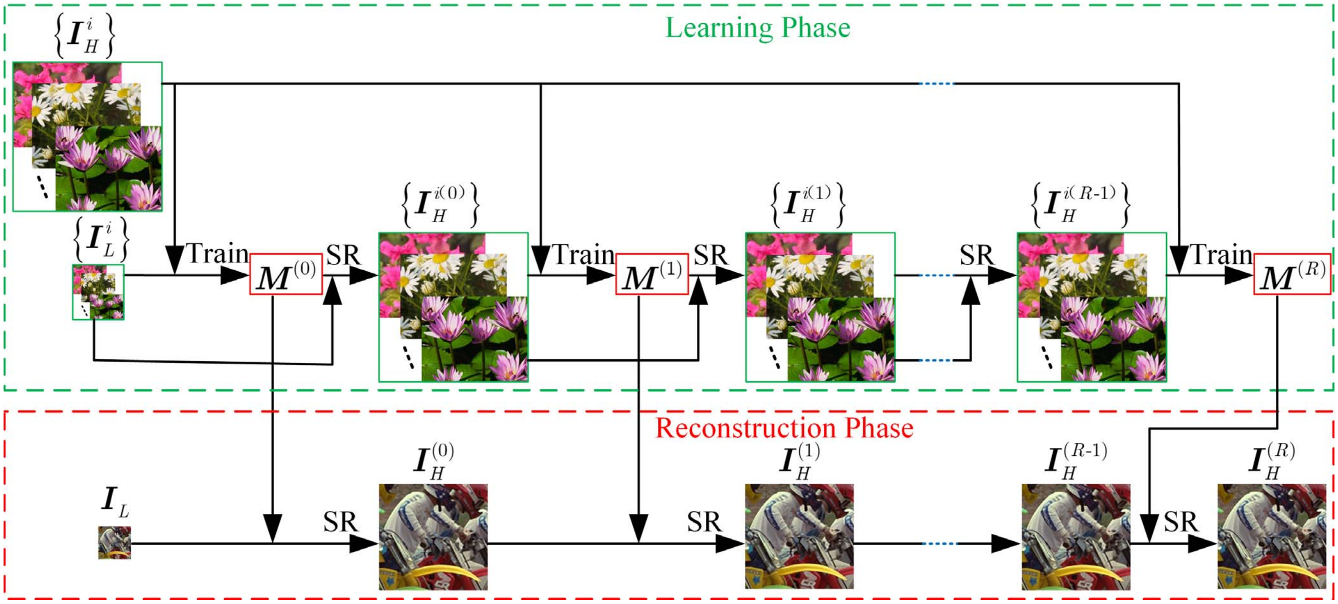


Fig. 4. Learning and reconstruction phases of our CRC pipeline.

where the projection matrix $\boldsymbol{F}_k$ can be obtained in the training stage by

$$F_k = N_{H,k}\left(N_{L,k}^T N_{L,k} + \lambda_2 I\right)^{-1} N_{L,k}^T, \quad k = 1, \ldots, K. \tag{4}$$

We can solve image SR problem by calculating the nearest dictionary atom $\boldsymbol{d}_k$ for each LR feature $\boldsymbol{y}_L^i$, followed by converting $\boldsymbol{y}_L^i$ to HR counterpart $\boldsymbol{y}_H^i$ by utilizing precomputed projection matrix $\boldsymbol{F}_k$, namely:

$$y_H^i = F_k y_L^i. \tag{5}$$

B. Multilayer Mapping Models Learning in CRC

As displayed in Fig. 4, we utilize original LR and HR training data to calculate PCA transform matrix $\mathbf{P}^{(0)}$, LR dictionary atoms $\{\boldsymbol{d}_k^{(0)}\}_{k=1}^K$, and their corresponding projection matrices $\{\boldsymbol{F}_k^{(0)}\}_{k=1}^K$ mapping the LR feature $\boldsymbol{y}_L$ to its corresponding HR feature $\boldsymbol{y}_H$. Here, we utilize the same process as in our previous work [30] for three considerations. First, we use gradient and Laplacian features in the horizontal and vertical directions as the LR features from the interpolated LR images. As we know, the gradient-based features would become sensitive to the area where gradients change apparently. However, most previous leading SR methods can produce SR results suffering from unpleasing artifacts along edges. Second, compared with most recent SR algorithms, the SR results from our previous work [30] usually restrain artifacts and have sharper edges. Third, we could evaluate the effectiveness of CRC by utilizing different existing SR algorithms as the initial methods in CRC. More details about different initializations in CRC are shown in Section IV-E.

In our first iteration, we utilize the method in our previous work [30] to super-resolve the original LR training images $\{\boldsymbol{I}_L^i\}_{i=1}^{N_I}$ to $\{\boldsymbol{I}_H^{i(0)}\}_{i=1}^{N_I}$ to have the same size of the original HR counterparts $\{\boldsymbol{I}_H^i\}_{i=1}^{N_I}$. For simplicity, we regard $\{\boldsymbol{I}_H^{i(r-1)}\}_{r=1}^R$ (R denotes the total number of cascade layer) as LR images. Specifically, we extract LR features from $\{\boldsymbol{I}_H^{i(0)}\}_{i=1}^{N_I}$, and conduct PCA dimension reduction to obtain $\mathbf{P}^{(1)}$. Using transformed LR features, we train the LR dictionary $\boldsymbol{D}^{(1)}$ via (1). For every atom $\boldsymbol{d}_k^{(1)}$, we organize its corresponding LR/HR neighborhoods from the training images $\{\boldsymbol{I}_H^{i(0)}, \boldsymbol{I}_H^i\}_{i=1}^{N_I}$. Then we obtain the projection matrix $\boldsymbol{F}^{(1)}$ via (4) and obtain the mapping model $\boldsymbol{M}^{(1)} = \{\mathbf{P}^{(1)}, \boldsymbol{D}^{(1)}, \boldsymbol{F}^{(1)}\}$.

When training mapping model for the rth iteration, we extract LR and HR feature pairs from $\{\boldsymbol{I}_H^{i(r-1)}, \boldsymbol{I}_H^i\}_{i=1}^{N_I}$ to learn the PCA transform matrix $\mathbf{P}^{(r)}$, LR dictionary atoms $\{\boldsymbol{d}_k^{(r)}\}_{k=1}^K$, and projection matrices $\{\boldsymbol{F}_k^{(r)}\}_{k=1}^K$ using the similar training pipeline as our previous work [30]. Different from A+ [20] using the Euclidean distance between atoms and the training samples for neighborhoods grouping, we use mutual coherence $\boldsymbol{c}_{k,i}$ between $\boldsymbol{d}_k^{(r)}$ and $\boldsymbol{y}_L^i$ via

$$c_{k,i} = \left|\left[d_k^{(r)}\right]^T y_L^i\right| \tag{6}$$

where $\boldsymbol{d}_k^{(r)}$ denotes kth atom in the learned $\boldsymbol{D}^{(r)} = \{\boldsymbol{d}_k^{(r)}\}_{k=1}^K$ and $\boldsymbol{y}_L^i$ is an LR feature. With the usage of mutual coherence $\boldsymbol{c}_{k,i}$, we comply with distance metrics between LR features and dictionary atoms in the reconstruction phase. We treat each iteration as a layer, where we learn the mapping model $\boldsymbol{M}^{(r)}$.

Before we train $\boldsymbol{M}^{(r+1)}$, we use the learned model $\boldsymbol{M}^{(r)}$ to upscale the original LR training images based on our previous method [30], and the outputs $\{\boldsymbol{I}_H^{i(r)}\}$ are then viewed as LR training images. We use $\{\boldsymbol{I}_H^{i(r)}\}$ and the original HR training images $\{\boldsymbol{I}_H^i\}$ to learn the mapping model $\boldsymbol{M}^{(r+1)}$ for the next iteration. After conducting R iterations, we will obtain $R+1$ mapping models. In a specific case, the CRC model can be viewed as our previous method [30], when we set R as 0 (we name this case as ICR). The multilayer mapping model learning phase of CRC is summarized in Algorithm 1.

C. Reconstruction Phase in CRC

After R iterations in the training stage, we obtain $R+1$ relationship projection models $\{\boldsymbol{M}^{(r)}\}_{r=0}^R$. Then image SR problem

Algorithm 1 Multilayer Mapping Models Learning

```
Input: Training images $\{I_L^i, I_H^i\}_{i=1}^{N_I}$, scaling factor $s$, number of iterations $R$, and dictionary size $K$.
Learn mapping model $M^{(0)} = \{\mathbf{P}^{(0)}, D^{(0)}, F^{(0)}\}$ from $\{I_L^i, I_H^i\}_{i=1}^{N_I}$ via our previous method [30].
Upscale $\{I_L^i\}_{i=1}^{N_I}$ to $\{I_H^{i(0)}\}_{i=1}^{N_I}$ with scaling factor $s$ by using the mapping model $M^{(0)}$.
for $r = 1$; $r \leq R$; $r = r + 1$; do
    Extract LR/HR features from $\{I_H^{i(r-1)}, I_H^i\}_{i=1}^{N_I}$.
    Compute the PCA transform matrix $\mathbf{P}^{(r)}$ from the LR feature space.
    Learn LR dictionary $D^{(r)} = \{d_i^{(r)}\}_{i=1}^{K}$ with mutual incoherence via (1).
    for $k = 1$; $k \leq K$; $k = k + 1$; do
        Sample the LR/HR neighborhoods $N_{L,k}^{(r)}$ and $N_{H,k}^{(r)}$ from the LR/HR feature space;
        Compute the projection matrix $F^{(r)} = \{F_k^{(r)}\}_{k=1}^{K}$ by (4).
    end for
Upscale $\{I_L^{i(r-1)}\}_{i=1}^{N_I}$ to $\{I_H^{i(r)}\}_{i=1}^{N_I}$ with scaling factor $s$ by mapping model $M^{(r)} = \{\mathbf{P}^{(r)}, D^{(r)}, F^{(r)}\}$.
end for
Output: $R$+1 mapping models $\{M^{(r)}\}_{r=0}^{R}$.
```

can be solved progressively layer by layer (see image SR stage in Fig. 4). In our CRC model, we use the output of our previous method [30] as the initial input $I_H^{(0)}$.

In the rth cascade layer of our CRC model, we extract LR feature and patch sets $\{y_L^{i(r-1)}, p_L^{i(r-1)}\}$ from the intermediate recovered results $I_H^{(r-1)}$. For each LR feature $y_L^{i(r-1)}$, we search its most related atom $d_k^{(r)}$ with the highest coherence $c_{k,i}$ from $D^{(r)}$, followed by reconstructing its HR counterpart $y_H^{i(r)}$ via (5). By using (7), the HR patch $p_H^{i(r)}$ is then obtained by adding interpolated LR patch $p_L^{i(r-1)}$ to HR feature $y_H^{i(r)}$. Then we obtain the final HR output $I_H^{(r)}$ by combining all the recovered HR patches and averaging intensity values in the overlapping areas.

After R iterations, the Rth SR result $I_H^{(R)}$ is recovered and used as final HR image I_H. Actually, the SR results in the intermediate cascade layer of CRC can also be viewed as the desired HR output, if more iterations cannot promote the SR results significantly. More details about how cascade layer numbers influence the performance would be presented in Section IV-B. We summarize the reconstruction phase of CRC in Algorithm 2.

IV. Experimental Results

A. Experimental Settings

1) Training Data: We use the 91 training images [35] as the training data, which are also used in our compared methods [12], [14]–[16], [18]–[21], [25], [27], [28]. As indicated in [42], data augmentation (e.g., rotation and flipping) would further improve the performance of image SR methods. For a fair comparison and better evaluate the effectiveness of CRC, we do not use data augmentation.

Algorithm 2 Image SR via CRC

```
Input: Scaling factor $s$, LR input $I_L$, mapping models $\{M^{(r)}\}_{r=0}^{R}$.
Upscale $I_L$ to $I_H^{(0)}$ with $s$ via our previous method [30];
for $r = 1$; $r \leq R$; $r = r + 1$; do
    Initial the intermediate HR result $I_H^{(r)} = \mathbf{0}$.
    Extract LR patches $\{p_L^{i(r-1)}\}_{i=1}^{N}$ and the corresponding dimension-reduced LR features $\{y_L^{i(r-1)}\}_{i=1}^{N}$ from $I_H^{(r-1)}$.
    for $i = 1$; $i \leq N$; $i = i + 1$; do
        Search index of the nearest LR dictionary atom of $y_L^{i(r-1)}$ by:
            $k = arg \max_{j=1,..,K} |d_j^{(r)^{\mathrm{T}}} y_L^{i(r-1)}|;$
        Obtain the corresponding HR feature $y_H^{i(r)}$ by using (5);
        Recover the corresponding HR patch $p_H^{i(r)}$ by using
            $p_H^{i(r)} = y_H^{i(r)} + p_L^{i(r-1)};$   (7)
        Add $p_H^{i(r)}$ to the corresponding pixels in $I_H^{(r)}$.
    end for
    Average $I_H^{(r)}$ in the overlapping region.
end for
Output: The final SR result $I_H = I_H^{(R)}$.
```

Fig. 5. Ten test images in Set10 from left to right: *Bike*, *Butterfly*, *Cameraman*, *Fence*, *Hat*, *House*, *Leaves*, *Parrots*, *Plants*, and *Starfish*.

2) Testing Data: In Fig. 5, we use Set10 chosen from [45]–[47], Set5 [16], Set14 [15], and SetSRF [28] to evaluate the performance of each SR method with upscaling factors being 2, 3, and 4. We also demonstrate the efficiency of our method via represented real-world LR images shown in Figs. 13(a) and 14(a).

3) Compared Methods: We compare CRC model with the recent published image SR methods. NE+LLE: neighbor embedding + locally linear embedding algorithm [12]. ScSR: SR method based on l_1-norm regularized sparse coding in [14]. K-SVD: sparse coding-based SR method using l_0-norm regularized optimization in [15]. NE+NNLS: neighbor embedding + nonnegative least squares method [16]. ANR: anchored neighborhood regression method [18]. BPJDL: a beta process prior is applied to learn the over-complete dictionaries in a Bayesian SR method [19]. A+: adjusted anchored neighborhood regression method [20]. SPM: a statistical prediction model based on sparse representation for SR [21]. SRCNN: SR using convolutional neural network [25]. DNC: a deep network cascade model to gradually upscale LR images layer by layer [26]. JOR: jointly optimized regressors for image SR [27]. ICR: a unified optimized SR framework using the same form of mutual coherence in dictionary learning, anchored neighborhoods sampling, and reconstruction phases [30]. ARFL: an improved variant of RFL (a standard model using SR forests [28]) using a modified training scheme for random forests presented in [51]. All implementations are from the publicly available codes provided by the authors except for DNC, whose results are provided by its authors. NE+LLE and NE+NNLS are improved versions as implemented in [18].

4) Evaluation Metrics: Apart from the widely used peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [52] indices, we also use another four image quality evaluation metrics, namely information fidelity criterion (IFC) [53], multiscale structure similarity index (MS-SSIM) [54], visual information fidelity (VIF) [55], and feature similarity (FSIM) [56]. The quantitative results are evaluated on the luminance channel between the recovered and original HR images.

5) Other Illustrations: We utilize bicubic interpolation to down-sample the real HR images, from which LR images are mimicked. Because human visual system is more sensitive to the luminance information than its chrominance counterparts [57], [58], we convert RGB images to YCbCr color space. We only considering performing image SR on the luminance channel (i.e., Y). We up-scale chromatic channels (i.e., Cb and Cr) with bicubic interpolation. Finally, we combine the SR results on three channels and convert them back to the RGB color space. Similar as A+ [20] and our previous work [30], we set $L = 3$, $K = 1024$, max $= 2048$, $\lambda_1 = 0.09$, and $\lambda_2 = 0.1$ throughout our experiments.

TABLE II
AVERAGE PERFORMANCE OF CRC ON SET10, SET5, SET14, AND SETSRF WITH DIFFERENT NUMBER OF CASCADE LAYERS R

Set10	Scale	Number of cascade layers					
		0	1	2	3	4	5
PSNR	×2	32.307	32.606	**32.675**	32.654	32.630	32.583
	×3	28.247	28.402	**28.433**	28.398	28.363	28.316
	×4	26.037	**26.151**	26.127	26.023	25.911	25.878
SSIM	×2	0.9311	0.9337	0.9343	0.9344	**0.9345**	0.9343
	×3	0.8556	0.8595	0.8608	0.8610	**0.8611**	0.8607
	×4	0.7826	0.7869	**0.7876**	0.7868	0.7864	0.7855
TIME	×2	0.44	0.90	1.37	1.83	2.28	2.74
	×3	0.27	0.56	0.85	1.11	1.41	1.68
	×4	0.18	0.47	0.76	1.06	1.35	1.66

Set5	Scale	Number of cascade layers					
		0	1	2	3	4	5
PSNR	×2	36.531	36.726	**36.753**	36.715	36.652	36.604
	×3	32.604	**32.751**	32.744	32.673	32.616	32.576
	×4	30.315	**30.400**	30.339	30.197	30.041	29.967
SSIM	×2	0.9542	0.9556	0.9559	0.9560	**0.9560**	0.9560
	×3	0.9091	0.9117	0.9124	0.9125	**0.9125**	0.9123
	×4	0.8615	0.8649	**0.8651**	0.8641	0.8632	0.8622
TIME	×2	0.80	1.64	2.48	3.35	4.18	5.04
	×3	0.45	0.96	1.46	1.98	2.49	3.03
	×4	0.33	0.87	1.41	1.96	2.64	3.13

Set14	Scale	Number of cascade layers					
		0	1	2	3	4	5
PSNR	×2	32.341	32.486	**32.501**	32.472	32.455	32.423
	×3	29.187	**29.280**	29.269	29.226	29.199	29.173
	×4	27.379	**27.443**	27.405	27.330	27.254	27.178
SSIM	×2	0.9049	0.9073	0.9079	0.9079	**0.9080**	0.9079
	×3	0.8198	0.8224	**0.8230**	0.8230	0.8229	0.8226
	×4	0.7512	**0.7538**	0.7535	0.7527	0.7521	0.7512
TIME	×2	1.61	3.30	4.99	6.72	8.42	10.14
	×3	0.92	1.96	3.00	4.07	5.12	6.16
	×4	0.67	1.73	2.81	3.86	4.96	6.05

SetSRF	Scale	Number of cascade layers					
		0	1	2	3	4	5
PSNR	×2	35.224	35.458	35.541	35.551	**35.551**	35.530
	×3	32.073	32.215	**32.224**	32.161	32.166	32.150
	×4	30.086	**30.155**	30.107	30.019	29.923	29.931
SSIM	×2	0.9443	0.9462	0.9468	0.9470	0.9472	**0.9472**
	×3	0.9011	0.9032	0.9038	0.9038	**0.9041**	0.9039
	×4	0.8634	**0.8650**	0.8646	0.8640	0.8639	0.8634
TIME	×2	2.36	4.85	7.34	9.85	12.35	14.89
	×3	1.34	2.87	4.39	5.93	7.46	8.99
	×4	1.00	2.57	4.18	5.81	7.44	9.08

B. Number of Cascade Layers in CRC

We upscale the LR input in an iterative way, and the more number of cascade layers we use, the more running time and memory utilization our CRC will consume. Therefore, in our experiments, we first investigate the number of cascade layers in CRC on Set10, Set5 [16], Set14 [15], and SetSRF [28] to pursue an optimal cascade layer number making a good tradeoff between the running time and reconstruction quality.

Table II displays the average performance and running time of CRC with different number of cascade layers R. First, it should be noted that we use our previous method [30] as CRC-0, which has achieved excellent performance. It is hard to obtain more gains from CRC-0. But our CRC-1 or CRC-2 would obtain notable gains from CRC-0 in most cases. As we can see from Table II, when scaling factor $s = 2$, in most testsets, quantitative results (e.g., SSIM) would continue to increase. When $R = 1$, the quality of SR results are enhanced largely. This tendency is coincident with the observation in Figs. 1 and 2. This is mainly because that the scaling factor s is not large, less artifacts can be produced in the reconstruction of intermediate HR images. However, when s is becoming larger (e.g., 3 and 4), more cascade layers would not enhance the reconstruction quality or even make the results worse qualitatively. Take results ($s = 4$) in Set14 as an example, only when $R = 1$ would the quantitative gains be positive. As R becomes larger, both PSNR and SSIM values would decrease. The main reason lies in that the initial interpreted HR images we use suffer from more blurring artifacts resulting the loss of textural details. This type of artifacts would weaken the reconstruction ability of our CRC model, which depends heavily on the LR features as shown in Fig. 3. In order to demonstrate our explanation visually, we extract some textural parts from the intermediate HR images as follows.

In Fig. 6, we use the representative image *Lena* full of textural information from Set14 to conduct the SR experiments with scaling factor $s = 4$. As we can see from Fig. 6(b), most textural details are missed in the SR process because of blurring effect of the interpolation-based SR method (Here we use bicubic interpolation). As a result, mapping models in CRC with larger depth would be harder to reconstruct more textural details, most of which in fact have been lost in the earlier iteration stages. As a result, the texture parts in Fig. 6(e)–(h) have been over-smoothed and fail to be recovered. The quantitative results (PNSR and SSIM) also begin to decrease as $R \geq 2$. So a larger R is not always good or suitable.

It should also be noticed that as R becomes larger, the memory and running time consumption in both training and reconstruction phases would also increase. A smaller value of R is preferred, especially when the scaling factor s is larger or there is plenty of textural information in the LR images.

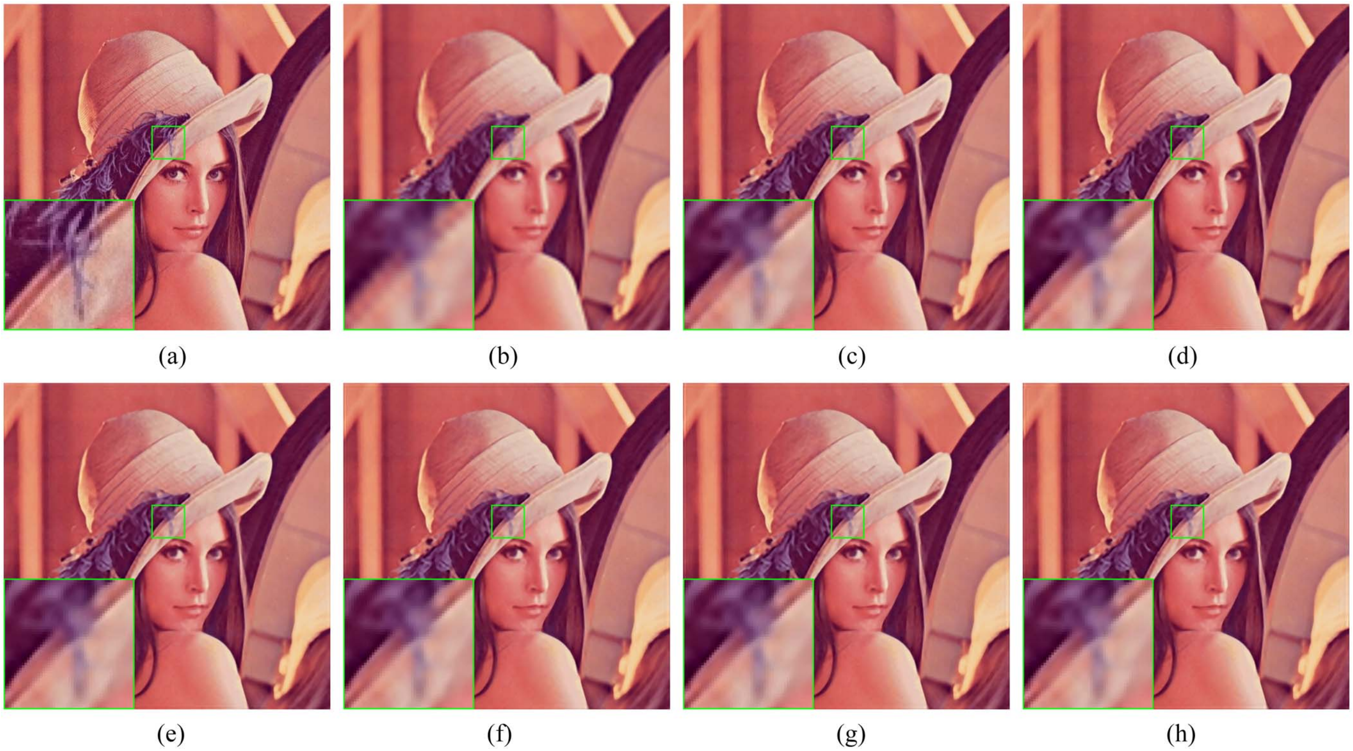


Fig. 6. Visual quality and quantitative results (PSNR/SSIM) of our CRC for the *Lena* image using different number of cascade layers ($s = 4$). (a) Original/PSNR/SSIM. (b) Bicubic/29.843/0.8140. (c) CRC-0/31.421/0.8457. (d) CRC-1/31.485/0.8475. (e) CRC-2/31.447/0.8473. (f) CRC-3/31.341/0.8465. (g) CRC-4/31.235/0.8459. (h) CRC-5/31.199/0.8452.

TABLE III
AVERAGE QUANTITATIVE RESULTS OF PSNR (dB), SSIM, IFC, MSSIM, VIF, FSIM, AND RUNNING TIME(S) OF DIFFERENT METHODS ON SET10

Set10	Scale	NE+LLE [12]	ScSR [14]	K-SVD [15]	NE+NNLS [16]	ANR [18]	BPJDL [19]	A+ [20]	SPM [21]	SRCNN [25]	DNC [26]	JOR [27]	ICR [30]	ARFL [28]	CRC-1 (R=1)	CRC-2 (R=2)
PSNR	×2	31.375	-	31.379	31.148	31.412	31.827	32.257	31.730	32.234	-	-	32.307	32.476	32.606	**32.675**
	×3	27.387	27.322	27.491	27.291	27.449	27.807	28.207	27.927	28.111	28.142	28.163	28.247	28.185	28.402	**28.433**
	×4	25.366	-	25.468	25.322	25.440	-	26.026	-	25.889	-	25.902	26.037	25.948	**26.151**	26.127
SSIM	×2	0.9218	-	0.9216	0.9193	0.9225	0.9257	0.9302	0.9231	0.9282	-	-	0.9311	0.9316	0.9337	**0.9343**
	×3	0.8326	0.8262	0.8351	0.8298	0.8341	0.8432	0.8543	0.8463	0.8466	0.8507	0.8531	0.8556	0.8516	0.8595	**0.8608**
	×4	0.7527	-	0.7576	0.7502	0.7553	-	0.7809	-	0.7712	-	0.7771	0.7826	0.7752	0.7869	**0.7876**
IFC	×2	7.4701	-	7.4392	7.1928	7.5970	7.2235	8.0644	7.0320	7.2390	-	-	8.0627	8.1460	8.2987	**8.3039**
	×3	4.1041	3.0761	4.1252	3.9761	4.1834	4.1333	4.5856	4.1287	4.0464	4.2224	4.5587	4.5894	4.5181	4.6976	**4.7357**
	×4	2.5812	-	2.5972	2.5047	2.6292	-	2.9175	-	2.5660	-	2.8509	2.9256	2.8370	3.0020	**3.0125**
MSSIM	×2	0.9903	-	0.9901	0.9899	0.9904	0.9908	0.9914	0.9904	0.9911	-	-	0.9915	0.9917	0.9919	**0.9920**
	×3	0.9661	0.9632	0.9661	0.9653	0.9666	0.9684	0.9707	0.9691	0.9689	0.9705	0.9706	0.9711	0.9703	0.9720	**0.9724**
	×4	0.9319	-	0.9327	0.9310	0.9331	-	0.9405	-	0.9364	-	0.9394	0.9411	0.9397	0.9428	**0.9434**
VIF	×2	0.7047	-	0.7028	0.6934	0.7076	0.7236	0.7350	0.7066	0.7219	-	-	0.7382	0.7439	0.7509	**0.7539**
	×3	0.4597	0.4020	0.4621	0.4504	0.4642	0.4830	0.5034	0.4796	0.4764	0.4939	0.5015	0.5065	0.5005	0.5165	**0.5202**
	×4	0.3175	-	0.3202	0.3116	0.3214	-	0.3544	-	0.3258	-	0.3479	0.3570	0.3490	0.3654	**0.3674**
FSIM	×2	0.9354	-	0.9362	0.9336	0.9354	0.9414	0.9475	0.9371	0.9463	-	-	0.9482	0.9510	0.9511	**0.9515**
	×3	0.8558	0.8542	0.8599	0.8537	0.8568	0.8700	0.8803	0.8713	0.8771	0.8792	0.8795	0.8815	0.8820	0.8849	**0.8854**
	×4	0.7967	-	0.8017	0.7946	0.7992	-	0.8227	-	0.8231	-	0.8178	0.8232	0.8236	**0.8267**	0.8259
TIME	×2	2.78	-	1.46	14.61	0.48	190.12	0.53	4.15	1.37	-	-	0.46	0.57	0.94	1.45
	×3	1.26	19.03	0.67	6.20	0.27	183.06	0.28	8.36	1.38	-	2.21	0.26	0.45	0.56	0.86
	×4	0.99	-	0.44	3.75	0.19	-	0.21	-	1.33	-	1.45	0.20	0.35	0.50	0.82

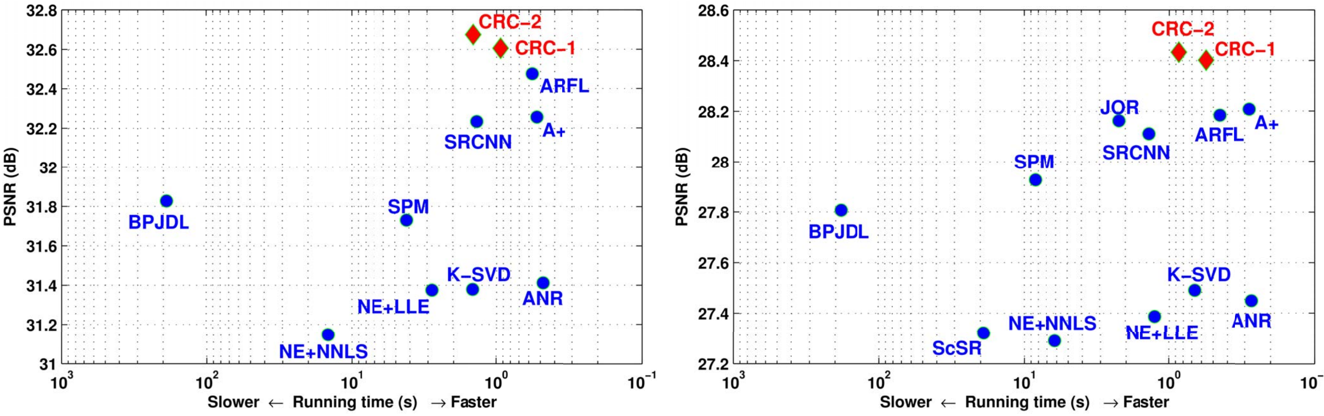
As a consequence, we would only consider the cases (i.e., $0 \leq R \leq 2$) in the following experiments.

C. Performance

As presented in Table III, our CRC achieves the best average performance in all the scaling factors in terms of all evaluation metrics on Set10. We also presented the quantitative comparisons with other methods on other three public evaluation sets Set5, Set14, and SetSRF in Table IV in terms of PSNR and SSIM. As Table IV shows, our CRC still achieves the best average performance. For scaling factor $s = 2$, the average PSNR gains obtained by CRC-2 are 0.20, 0.10, 0.12, and 0.19 dB, higher than that of the second best method ARFL [28] on the four datasets. For scaling factor $s = 3$, the average PSNR gains obtained by CRC-2 are 0.19, 0.08, and 0.15 dB, higher than that of the second best method ICR [30], on Set10, Set14, and SetSRF. The average PSNR gain obtained by CRC-2 is 0.12 dB higher than that of the second best method A+ [20] on SetSRF. For scaling factor $s = 4$, the

TABLE IV
AVERAGE QUANTITATIVE RESULTS OF PSNR (dB) AND SSIM OF DIFFERENT METHODS ON SET5, SET14, AND SETSRF

Sets	Eva. Mat.	Scale	NE+LLE [12]	ScSR [14]	K-SVD [15]	NE+NNLS [16]	ANR [18]	BPJDL [19]	A+ [20]	SPM [21]	SRCNN [25]	DNC [26]	JOR [27]	ICR [30]	ARFL [28]	CRC-1 (R=1)	CRC-2 (R=2)
Set5	PSNR	×2	35.776	-	35.768	35.412	35.835	36.102	36.576	35.964	36.354	-	-	36.531	36.651	36.726	**36.753**
		×3	31.847	31.469	31.922	31.647	31.903	32.206	32.621	32.284	32.409	32.509	32.590	32.604	32.545	**32.751**	32.744
		×4	29.629	-	29.707	29.487	29.707	-	30.306	-	30.108	-	30.229	30.315	30.203	**30.400**	30.339
	SSIM	×2	0.9490	-	0.9492	0.9470	0.9499	0.9506	0.9541	0.9484	0.9522	-	-	0.9542	0.9545	0.9556	**0.9559**
		×3	0.8954	0.8827	0.8970	0.8924	0.8963	0.9015	0.9091	0.9029	0.9035	0.9071	0.9079	0.9091	0.9069	0.9117	**0.9124**
		×4	0.8403	-	0.8432	0.8368	0.8419	-	0.8606	-	0.8533	-	0.8584	0.8615	0.8555	0.8649	**0.8651**
Set14	PSNR	×2	31.807	-	31.859	31.598	31.848	32.020	32.308	31.967	32.221	-	-	32.341	32.384	32.486	**32.501**
		×3	28.635	28.395	28.715	28.525	28.681	28.871	29.174	28.964	29.043	29.123	29.144	29.187	29.145	**29.280**	29.269
		×4	26.846	-	26.929	26.799	26.905	-	27.374	-	27.239	-	27.315	27.379	27.329	**27.443**	27.405
	SSIM	×2	0.8999	-	0.8990	0.8960	0.9009	0.9032	0.9044	0.8995	0.9043	-	-	0.9049	0.9058	0.9073	**0.9079**
		×3	0.8091	0.7974	0.8090	0.8045	0.8105	0.8154	0.8196	0.8140	0.8156	0.8182	0.8192	0.8198	0.8183	0.8224	**0.8230**
		×4	0.7356	-	0.7364	0.7321	0.7376	-	0.7506	-	0.7425	-	0.7488	0.7512	0.7470	**0.7538**	0.7535
SetSRF	PSNR	×2	34.357	-	34.526	34.246	34.394	34.622	35.185	34.624	35.092	-	-	35.224	35.347	35.458	**35.541**
		×3	31.229	30.919	31.420	31.200	31.266	31.560	32.054	31.790	31.927	31.826	32.018	32.073	32.040	32.215	**32.224**
		×4	29.379	-	29.548	29.342	29.436	-	30.079	-	29.851	-	30.041	30.086	30.031	**30.155**	30.107
	SSIM	×2	0.9380	-	0.9391	0.9372	0.9392	0.9398	0.9439	0.9381	0.9429	-	-	0.9443	0.9447	0.9462	**0.9468**
		×3	0.8903	0.8808	0.8926	0.8892	0.8916	0.8942	0.9007	0.8955	0.8979	0.8977	0.9006	0.9011	0.8994	0.9032	**0.9038**
		×4	0.8509	-	0.8535	0.8495	0.8524	-	0.8629	-	0.8580	-	0.8623	0.8634	0.8596	**0.8650**	0.8646



(a) (b)

Fig. 7. Proposed CRC method achieves the best super-resolution quality, while maintaining competitive speed when compared with other leading methods. These two charts are based on the results on Set10 summarized in Table III. Scaling factor (a) $s = 2$ and (b) $s = 3$.

average PSNR gains obtained by CRC-2 are 0.09, 0.02, 0.03, and 0.02 dB, higher than that of the second best approach ICR [30], on the four datasets.

When the scaling factor changes, the second best method may also change. However, our CRC will always achieve the best performance among all methods and scaling factors, which helps to illustrate the robustness of CRC. While, the average performance gains of CRC-2 over CRC-1 is not very significant. Consequently, we choose $R = 1$ or $R = 2$ as an optimal tradeoff balancing the computation complexity and quality of SR results, which is coincident with the discussion in Section IV-B.

Furthermore, we also show the running time comparisons of several leading SR algorithms, as well as their reconstruction performance on Set10 in Fig. 7. We record the running time of each methods in a MATLAB 2012a environment using the same machine (3.20 GHz Core i5-3470 with 16-GB RAM). The results of all compared methods are achieved by using the corresponding authors MATLAB + MEX implementation. Because all of the original HR testing images in Set10 keep the same size: 256 × 256, Fig. 7 shows the relationship between performance and speed more clearly, regardless of the influence of image resolution. When $s = 2$ [see Fig. 7(a)], ARFL [28] outperforms A+ [20], and when $s = 3$ [see Fig. 7(b)], the performance of ARFL [28] would be slightly worse than that of A+ [20]. Similar situation can also be observed between BPJDL [19] and SPM [21]. However, our CRC would outperform most of other competing methods, regardless of the scaling factor. It is clear that our CRC would always be the best method with competitive running time, achieving a good tradeoff between performance and speed.

D. Visual Quality

To visually illustrate the effectiveness of our CRC, in Figs. 9–12, we compare our image SR results with these of ANR [18], BPJDL [19], SPM [21], SRCNN [25], A+ [20], and ARFL [28], as they are recently published leading algorithms and have very excellent performance.

As shown by these visual results, ANR [18] would generate some blurred edges in Fig. 10(b) and unpleasing artifacts in Fig. 9(b). This is mainly because ANR uses only a small number of dictionary atoms to compute the regressors for each anchored atom. BPJDL [19] suppresses the blurring

TABLE V
AVERAGE PSNR AND SSIM VALUES OF OUR CRC ON SET10 USING FOUR DIFFERENT INITIALIZATIONS (SCALING FACTOR $s = 3$)

Methods	*bike*	*butterfly*	*cameraman*	*fence*	*hat*	*house*	*leaves*	*parrots*	*plants*	*starfish*	**Average**
Bicubic	22.826 0.7040	24.040 0.8217	23.770 0.7782	22.077 0.5905	29.329 0.8353	29.806 0.8357	23.498 0.8046	27.937 0.8799	31.299 0.8720	27.013 0.8140	26.159 0.7936
Bicubic+	**24.759** **0.7997**	**27.372** **0.9114**	**25.133** **0.8279**	**22.730** **0.6577**	**31.375** **0.8796**	**32.121** **0.8696**	**26.511** **0.9083**	**29.989** **0.9151**	**33.825** **0.9222**	**28.652** **0.8646**	**28.247** **0.8556**
A+ [20]	24.685 0.7986	27.354 0.9100	25.092 0.8271	22.659 0.6545	31.331 0.8785	32.036 0.8688	26.455 0.9054	30.044 0.9150	33.760 0.9208	28.652 0.8647	28.207 0.8543
A++	**24.877** **0.8067**	**27.625** **0.9162**	**25.226** **0.8310**	**22.679** **0.6586**	**31.437** **0.8814**	**32.128** **0.8709**	**26.692** **0.9117**	**30.203** **0.9174**	**34.029** **0.9251**	**28.851** **0.8699**	**28.375** **0.8589**
JOR [27]	24.617 0.7956	27.077 0.9038	25.034 0.8257	22.708 0.6560	31.226 0.8768	31.916 0.8678	26.536 0.9055	30.032 0.9143	33.768 0.9202	28.713 0.8657	28.163 0.8531
JOR+	**24.837** **0.8045**	**27.479** **0.9140**	**25.142** **0.8296**	**22.761** **0.6623**	**31.415** **0.8807**	**31.983** **0.8697**	**26.685** **0.9126**	**30.233** **0.9167**	**33.986** **0.9243**	**28.849** **0.8701**	**28.337** **0.8584**
ICR [30]	24.759 0.7997	27.372 0.9114	25.133 0.8279	22.730 0.6577	31.375 0.8796	32.121 0.8696	26.511 0.9083	29.989 0.9151	33.825 0.9222	28.652 0.8646	28.247 0.8556
ICR+	**24.908** **0.8071**	**27.604** **0.9161**	**25.242** **0.8311**	**22.744** **0.6610**	**31.516** **0.8826**	**32.196** **0.8709**	**26.735** **0.9129**	**30.146** **0.9173**	**34.093** **0.9262**	**28.838** **0.8697**	**28.402** **0.8595**

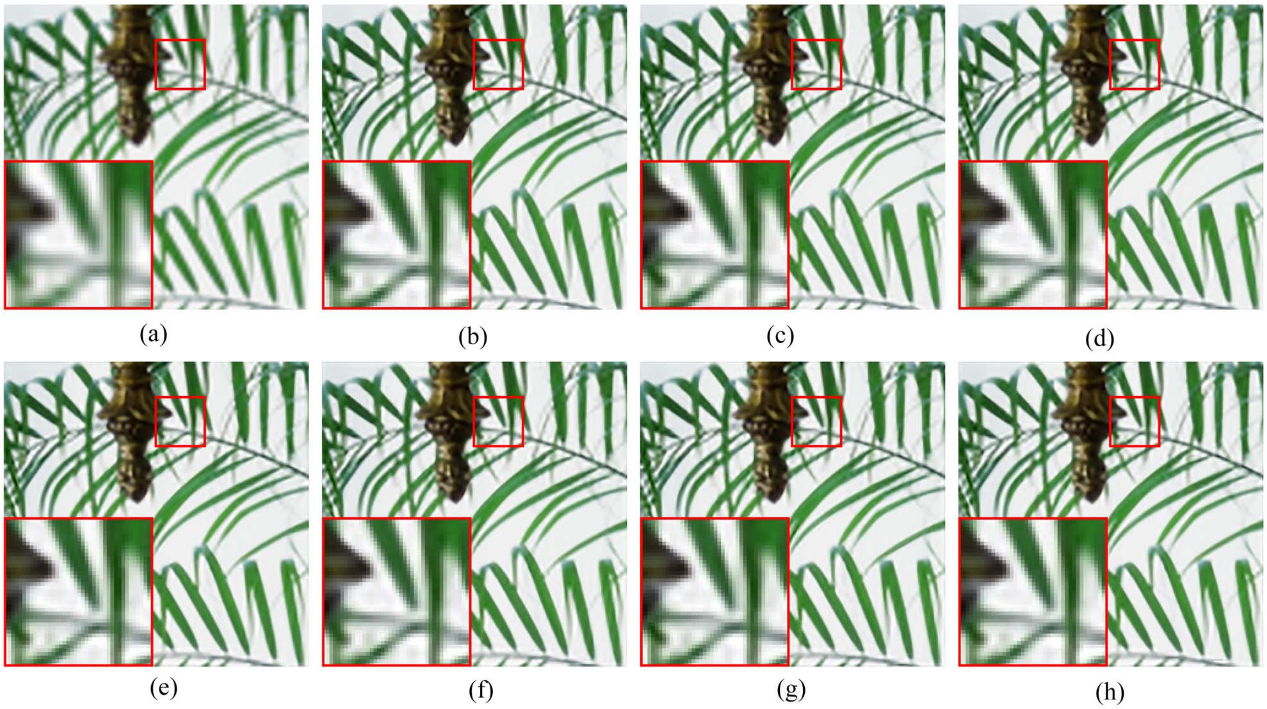


Fig. 8. Visual quality and quantitative results (PSNR/SSIM) of our CRC method for the *Leaves* image by using different initial methods ($s = 3$). Images in the first row are produced by the representative baselines and recent leading SR methods. The images in the second row are the results by incorporating our CRC into the corresponding methods. (a) Bicubic/23.498/0.8046. (b) A+ [20]/26.455/0.9054. (c) JOR [27]/26.536/0.9055. (d) ICR [30]/26.511/0.9083. (e) Bicubic+/26.511/0.9083. (f) A++/26.692/0.9117. (g) JOR+/26.685/0.9126. (h) ICR+/26.735/0.9129.

effects to a certain degree, however, it also produces blurred edges in Fig. 10(c) and jaggy artifacts in Fig. 12(c). As shown in Figs. 10(d) and 11(d), SPM [21] presents its strong ability on deblurring along the main edges. While it generates over-sharped edges in Fig. 10(d) and also unable to recover the detailed textures in Fig. 11(d). These results indicate that dictionary learning-based method would often introduce obvious artifacts along the main edges and the dictionary atoms lack abundant presentation ability. As shown in Figs. 10(e) and 12(e), SRCNN [25] obtains relatively high PSNR values, however, it generates results suffering from serious jaggy artifacts along the dominant edges. This helps to explain why SRCNN [25] obtains relatively high PSNR values while failing to show similar advantages in terms of other evaluation metrics (e.g., SSIM, IFC, MSSIM, VIF, and FSIM) as presented in Tables III and IV. On the other hand, A+ [20] and ARFL [28], as regression-based methods, can alleviate the blurring effects and ringing artifacts along the edges significantly compared to former discussed methods. However, they can also generate jaggy artifacts along the edges [see Figs. 10(f) and (g) and 11(f) and (g)] and fail to recover more detailed textures [see Fig. 9(f) and (g)]. We can observe that CRC achieves sharper edges without obvious unpleasing artifacts [see Figs. 10(h) and 12(h)] and finer textures with more details [see Fig. 9(h)], thus being more faithful to the original HR counterparts.

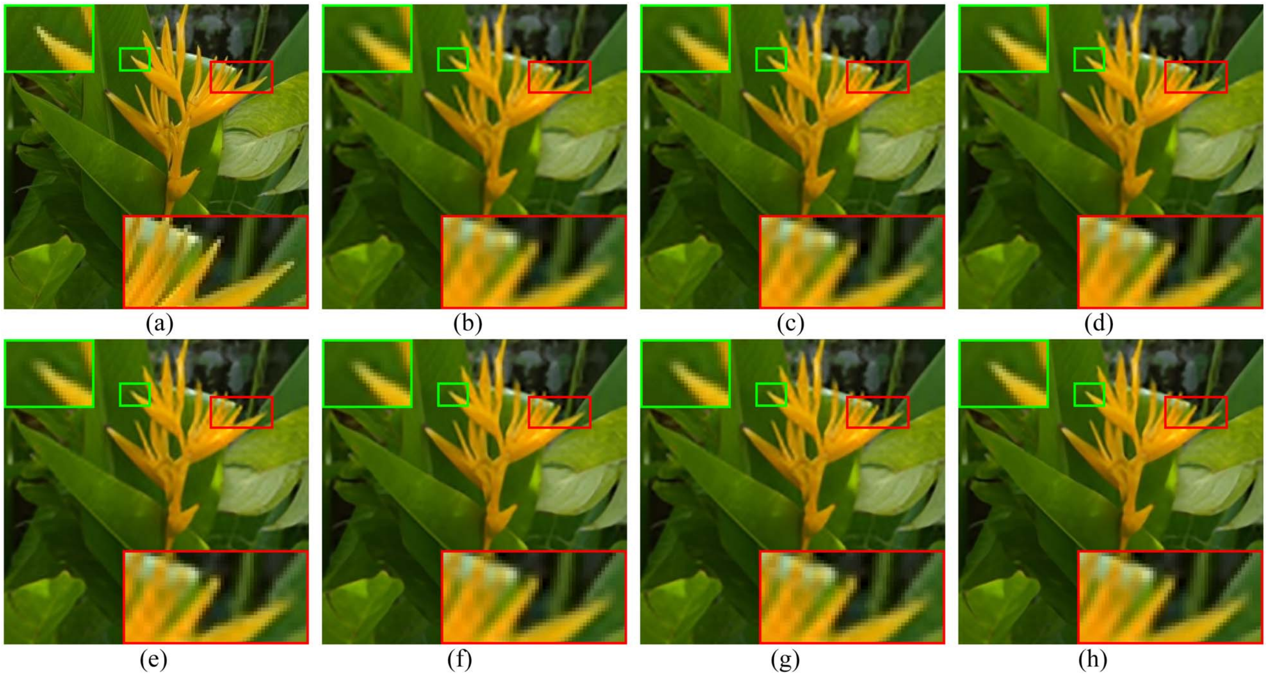


Fig. 9. Visual comparisons on "*Plants*" from Set10 by using different SR methods ($s = 3$). (a) Original/PSNR/SSIM. (b) ANR [18]/32.796/0.9052. (c) BPJDL [19]/33.126/0.9099. (d) SPM [21]/33.139/0.9066. (e) SRCNN [25]/33.523/0.9111. (f) A+ [20]/33.760/0.9208. (g) ARFL [28]/33.634/0.9165. (h) CRC-2/34.169/0.9274.

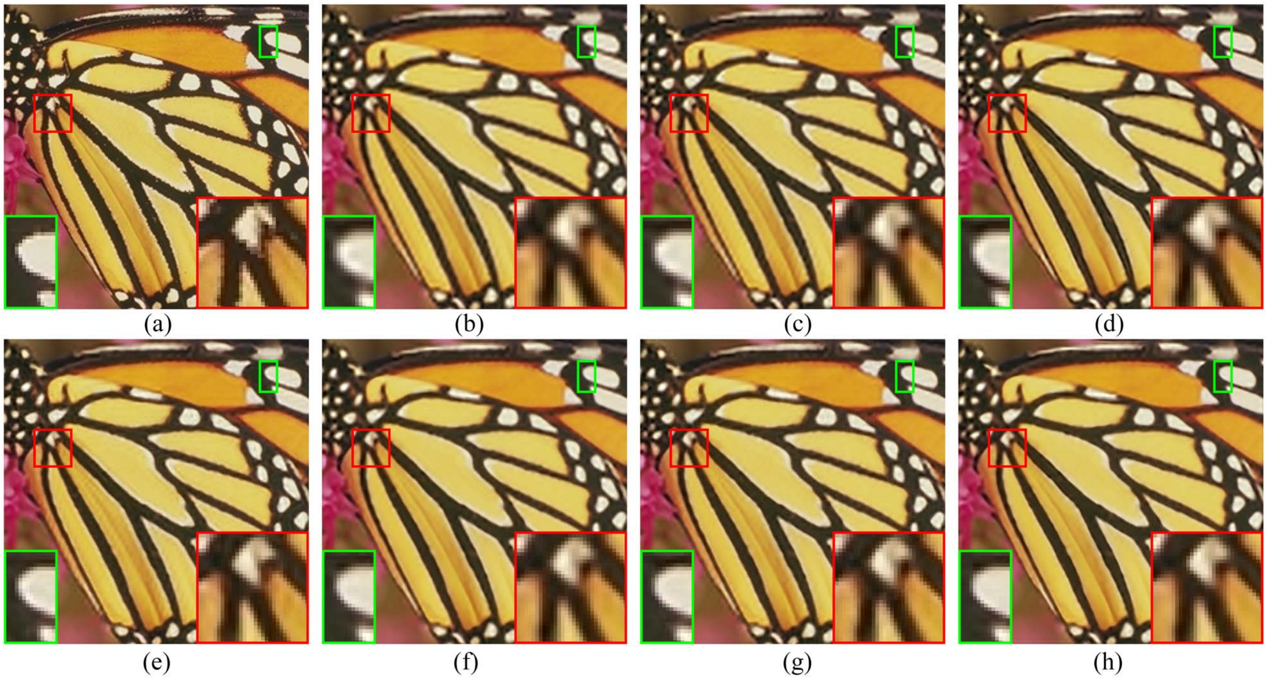
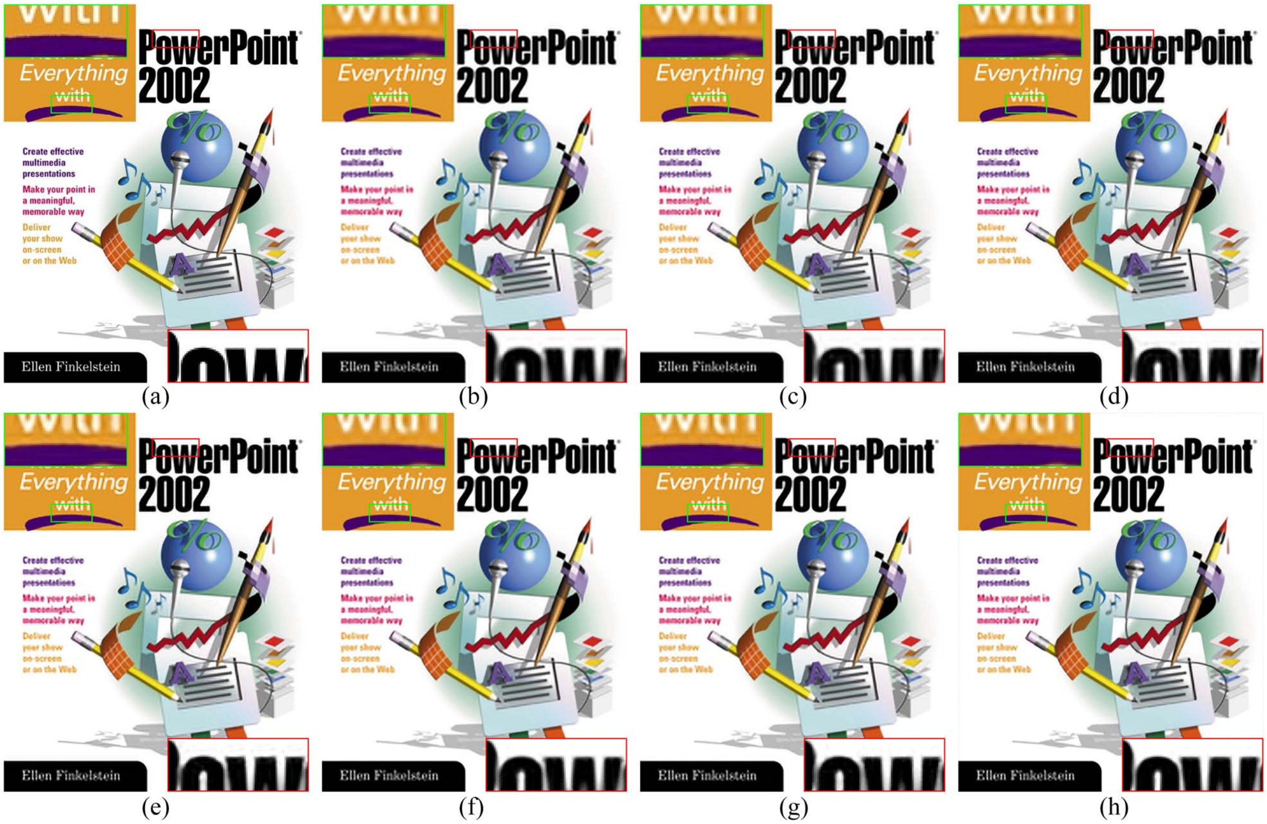


Fig. 10. Visual comparisons on "*Butterfly*" from Set5 by using different SR methods ($s = 3$). (a) Original/PSNR/SSIM. (b) ANR [18]/25.832/0.8690. (c) BPJDL [19]/26.516/0.8875. (d) SPM [21]/26.784/0.8986. (e) SRCNN [25]/27.589/0.9012. (f) A+ [20]/27.354/0.9100. (g) ARFL [28]/27.389/0.9063. (h) CRC-2/27.631/0.9177.

E. Different Initializations in CRC

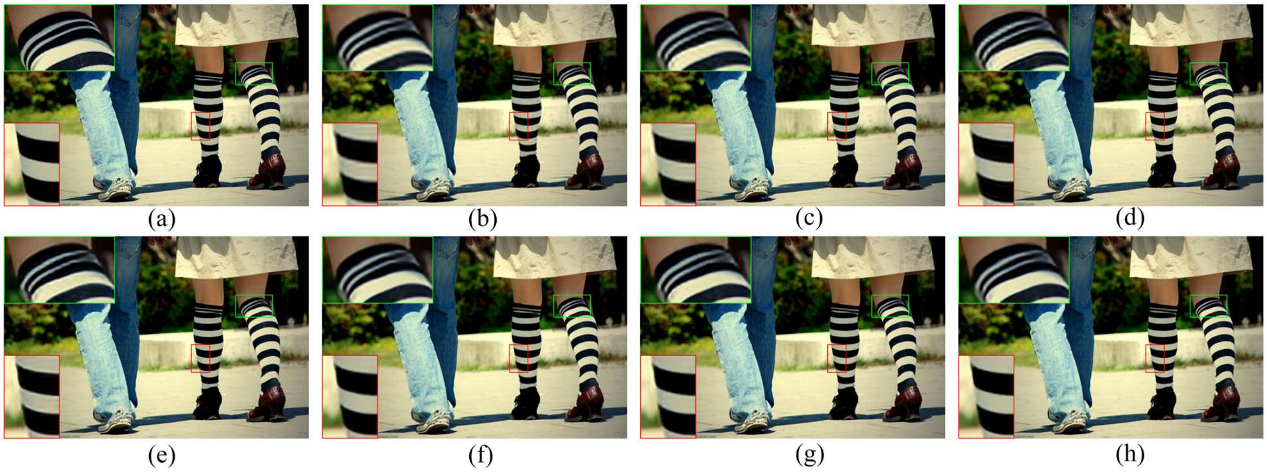
In this paper, we choose the result of our previous work [30] as the initialization in our CRC, different from our former work [41], which uses A+ [20] for the initialization. Here, we investigate how different initializations would affect the performance of CRC. The bicubic interpolation is extended as Bicubic+ by using only one iteration, namely one cascade layer. We also extend several recent published image SR algorithms A+ [20], JOR [27], and ICR [30] as A++, JOR+, and ICR+, respectively, by using one cascade layer.

In Table V, we evaluate the performance (e.g., PSNR and SSIM) of our CRC on Set10 with four SR methods, bicubic,


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. Visual comparisons on "*ppt3*" from Set14 by using different SR methods ($s = 3$). (a) Original/PSNR/SSIM. (b) ANR [18]/25.276/0.9130. (c) BPJDL [19]/25.548/0.9206. (d) SPM [21]/25.867/0.9257. (e) SRCNN [25]/26.318/0.9290. (f) A+ [20]/26.348/0.9397. (g) ARFL [28]/26.201/0.9350. (h) CRC-2/26.714/0.9459.


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 12. Visual comparisons on "*Legs*" from SetSRF by using different SR methods ($s = 3$). (a) Original/PSNR/SSIM. (b) ANR [18]/30.155/0.8855. (c) BPJDL [19]/30.664/0.8910. (d) SPM [21]/31.172/0.8928. (e) SRCNN [25]/31.407/0.8954. (f) A+ [20]/31.628/0.9017. (g) ARFL [28]/31.630/0.8999. (h) CRC-2/32.066/0.9054.

A+ [20], JOR [27], and ICR [30] as the initial methods, respectively. When $R = 0$, we do not use CRC to enhance SR results further. Then we conduct CRC only once (i.e., $R = 1$) based on the initial results. As we can see from Table V, with the usage of CRC, all the values of PSNR and SSIM for each baseline methods are higher than those of their initial results. When comparing Table III with Table V under scaling factor $s = 3$, we find that most of extended methods (e.g., A++, JOR+, and ICR+) outperform recently published state-of-the-art methods (e.g., A+ [20] and ARFL [28]). This demonstrates the effectiveness of our proposed CRC framework.

In order to visually demonstrate the influence of CRC, we give four pairs of experiments on *Leaves* image in Fig. 8. We can see that the SR results by bicubic interpolation would suffer from seriously blurring effects and lose many details

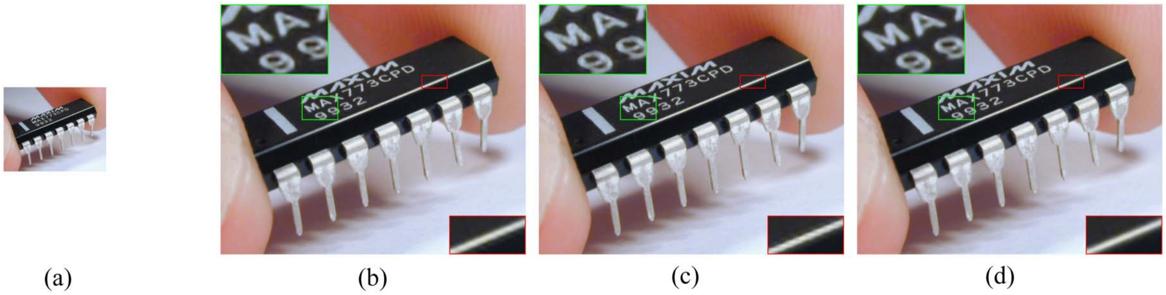


Fig. 13. Visual image and running time(s) comparisons with different SR methods on the "*Chip*" image (scaling factor $s = 3$). (a) Original LR image. (b) BPJDL [19]. (c) SRCNN [25]. (d) CRC-2.

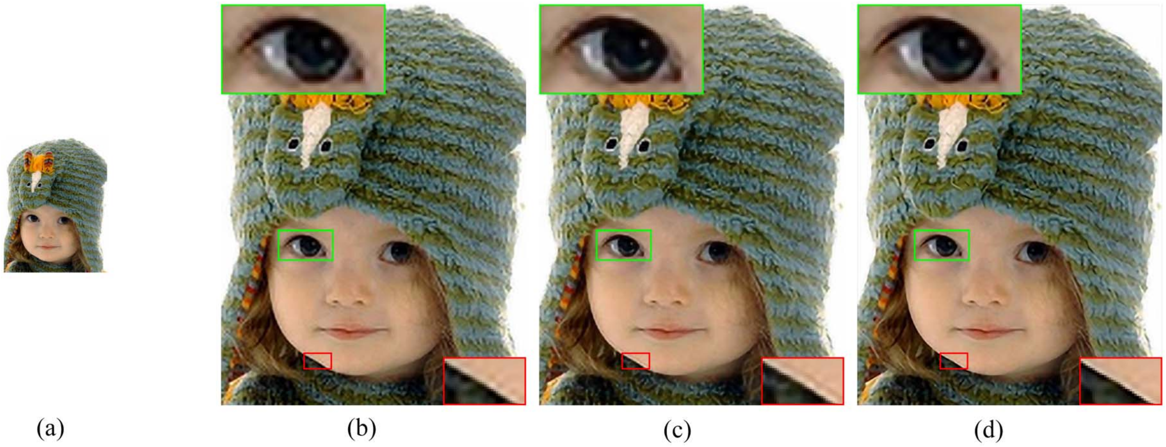


Fig. 14. Visual image and running time(s) comparisons with different SR methods on the "*Hatc*" image (scaling factor $s = 3$). (a) Original LR image. (b) BPJDL [19]. (c) SRCNN [25]. (d) CRC-2.

[see Fig. 8(a)], which can be alleviated by using CRC only once [see Fig. 8(e)]. Even though A+ [20] obtains state-of-the-art results, its results can also be further enhanced by CRC [see Fig. 8(b) and (f)]. Similar conclusions can be obtained from the results by JOR+ and ICR+. To sum up, no matter we use simple interpolation-based method (e.g., bicubic interpolation) or one of state-of-the-art methods (e.g., A+ [20]) as the initialization, our CRC would improve the quality of most SR results in an iterative way.

F. Comparison on Real-World LR Images

In the above experiments, we conduct image SR using simulated LR images, which is downsampled by bicubic interpolation. That is not the same case as LR images in our real world. To better demonstrate the effectiveness of CRC model, similar as [38], we choose two typical real-world images as LR inputs, namely the *Clip* [shown in Fig. 13(a) with 244 × 200 pixels] and *Hatc* [shown in Fig. 14(a) with 133 × 174 pixels]. We choose the representative dictionary learning-based method BPJDL [19] and deep learning-based method SRCNN [25] for comparison. As the original HR images are not available for those real-world LR images, we mainly compare the visual results ($s = 3$) and report the running time used in the reconstruction phase by each method to compare the efficiency of different image SR methods.

As we can see from Fig. 13, a comparison of the magnified letters and edge shows that both BPJDL [19] [see Fig. 13(b)] and SRCNN [25] [see Fig. 13(c)] produce results with obvious ringing artifacts. While CRC-2 would produce HR images with less artifacts and sharper edges [see Fig. 13(d)]. Similar observations can also be found in Fig. 14, where displeasing artifacts are produced around the eye [see Fig. 14(b) and (c)] and along the face [see Figs. 13(b) and 14(c)]. Nevertheless, our CRC-2 shows promise to reduce those artifacts [see Fig. 14(d)]. This comparison shows the superior reconstruction ability of our proposed method, because additional cascade layers help to reconstruct more details using the intermediate HR images.

At the same time, the running time used by CRC-2 also ranks the least compared to the other two recent published SR methods. Specifically, CRC-2 is 197–300 times faster than that of BPJDL [19] and about 3 times faster than that of SRCNN [25]. It is mainly because we use collaborative representation for fast image SR. This running time comparison along with the visual comparison above can further demonstrate the efficiency of our CRC model, which can produce high-quality HR images with less obvious artifacts while using less running time. This advantage would be very useful for the practical cases.

V. Conclusion

In this paper, we have proposed a simple yet effective image SR method called CRC. We observe that the intermediate recovered results contain more helpful information than that of interpolated ones. The intermediate SR results allows us to extract LR features from them and enhance SR results further. In CRC model, multilayer mapping models are learned from the each cascade layer, whose intermediate outputs are mapped to their corresponding HR images. By using these multilayer mapping models, one SR method is incorporated to upscale the LR input as an initialization. LR features extracted from these intermediate results can be used to recover HR images for the next cascade layer. When compared with recent leading SR algorithms, CRC presents compelling SR results quantitatively and visually. We conjecture that additional performance gains can be achieved by utilizing more effective feature extraction strategy. Then, on the basis of better features, more cascade layers would further improve SR performance. In the future works, we firmly believe our CRC model could be applied to other image restoration tasks such as compression artifact removal, inpainting, deblocking, and denoising.

References

[1] N. A. Dodgson, "Quadratic interpolation for image resampling," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1322–1326, Sep. 1997.

[2] J. Suo, L. Lin, S. Shan, X. Chen, and W. Gao, "High-resolution face fusion for gender conversion," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 2, pp. 226–237, Mar. 2011.

[3] X. Lu, Y. Yuan, and P. Yan, "Image super-resolution via double sparsity regularized manifold learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 12, pp. 2022–2033, Dec. 2013.

[4] J. Shen, J. Shen, T. Mei, and X. Gao, "Landmark reranking for smart travel guide systems by combining and analyzing diverse media," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1492–1504, Nov. 2016.

[5] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoustics Speech Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[6] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[7] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.

[8] J. Sun, J. Zhu, and M. F. Tappen, "Context-constrained hallucination for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 231–238.

[9] H. Xu, G. Zhai, and X. Yang, "Single image super-resolution with detail enhancement based on local fractal analysis of gradient," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1740–1754, Oct. 2013.

[10] L. Wang, S. Xiang, G. Meng, H.-Y. Wu, and C. Pan, "Edge-directed single-image super-resolution via adaptive gradient magnitude self-interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1289–1299, Aug. 2013.

[11] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with non-local means and steering kernel regression," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4544–4556, Nov. 2012.

[12] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, Jun./Jul. 2004, pp. 1–6.

[13] W. Li, D. Zhang, Z. Liu, and X. Qiao, "Fast block-based image restoration employing the improved best neighborhood matching approach," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 4, pp. 546–555, Jul. 2005.

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

[15] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. 7th Int. Conf. Curves Surfaces*, Avignon, France, Jun. 2010, pp. 711–730.

[16] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2012, pp. 1–10.

[17] L. Guan, "An optimal neuron evolution algorithm for constrained quadratic programming in image restoration," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 26, no. 4, pp. 513–518, Jul. 1996.

[18] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput.Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1920–1927.

[19] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 345–352.

[20] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. IEEE Asian Conf. Comput. Vis.*, Nov. 2014, pp. 111–126.

[21] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2569–2582, Jun. 2014.

[22] H. Xiao, L. H. Lee, and C.-H. Chen, "Optimal budget allocation rule for simulation optimization using quadratic regression in partitioned domains," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 7, pp. 1047–1062, Jul. 2015.

[23] W. Yang, T. Yuan, W. Wang, F. Zhou, and Q. Liao, "Single-image super-resolution by subdictionary coding and kernel regression," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[24] X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li, "Geometry constrained sparse coding for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1648–1655.

[25] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 184–199.

[26] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 49–64.

[27] D. Dai, R. Timofte, and L. Van Gool, "Jointly optimized regressors for image super-resolution," *Comput. Graph. Forum (Eurographics)*, vol. 34, no. 2, pp. 95–104, 2015.

[28] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3791–3799.

[29] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5197–5206.

[30] Y. Zhang, K. Gu, Y. Zhang, J. Zhang, and Q. Dai, "Image super-resolution based on dictionary learning and anchored neighborhood regression with mutual incoherence," in *Proc. Int. Conf. Image Process.*, Québec City, QC, Canada, Sep. 2015, pp. 591–595.

[31] Q. Wang and Y. Yuan, "High quality image resizing," *Neurocomputing*, vol. 131, pp. 348–356, May 2014.

[32] Y. Tang, H. Chen, Z. Liu, B. Song, and Q. Wang, "Example-based super-resolution via social images," *Neurocomputing*, vol. 172, pp. 38–47, Jan. 2016.

[33] J. Kim, J. K. Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1646–1654.

[34] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[35] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[36] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[37] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 561–568.

[38] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong, "Learning multiple linear mappings for efficient single image super-resolution," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 846–861, Mar. 2015.

[39] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[40] Y. Zhang, Y. Zhang, J. Zhang, and Q. Dai, "CCR: Clustering and collaborative representation for fast single image super-resolution," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 405–417, Mar. 2016.

[41] Y. Zhang, Y. Zhang, J. Zhang, H. Wang, and Q. Dai, "Single image super-resolution via iterative collaborative representation," in *Proc. Pac.-Rim Conf. Multimedia*, Sep. 2015, pp. 63–73.

[42] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1865–1873.

[43] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 349–356.

[44] J. Kim, J. K. Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1637–1645.

[45] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1838–1857, Jul. 2011.

[46] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1382–1394, Apr. 2013.

[47] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[48] Z. Cao *et al.*, "A fast orientation estimation approach of natural images," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1589–1597, Nov. 2016.

[49] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.

[50] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 660–667.

[51] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Alternating regression forests for object detection and pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 417–424.

[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[53] H. R. Sheikh, A. C. Bovik, and G. De Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2117–2128, Dec. 2005.

[54] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. Conf. Rec. 37th Asilomar Conf. Signals Syst. Comput.*, vol. 2. Pacific Grove, CA, USA, 2004, pp. 1398–1402.

[55] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[56] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

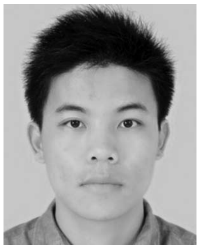
[57] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, Oct. 2000.

[58] Y. Tang, P. Yan, Y. Yuan, and X. Li, "Single-image super-resolution via local learning," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 1, pp. 15–23, Mar. 2011.

Yongbing Zhang received the B.A. degree in English and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively.

He joined the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China, in 2010, where he is currently an Associate Research Fellow. His current research interests include video processing, image and video coding, video streaming, and transmission.

Dr. Zhang was a recipient of the Best Student Paper Award of the IEEE International Conference on Visual Communication and Image Processing in 2015.



Yulun Zhang (S'15) received the B.E. degree from the School of Electronic Engineering, Xidian University, Xi'an, China, in 2013. He is currently pursuing the M.E. degree with the Department of Automation, Tsinghua University, Beijing, China.

His current research interests include generative adversarial networks, sparse representation, super-resolution, and transfer learning.

Mr. Zhang was a recipient of the Best Student Paper Award of the IEEE International Conference on Visual Communication and Image Processing in 2015.

Jian Zhang (M'14) received the B.Sc. degree in mathematics from the Harbin Institute of Technology (HIT), Harbin, China, in 2007, and the M.Eng. and Ph.D. degrees from the School of Computer Science and Technology, HIT, in 2009 and 2014, respectively.

From 2014 to 2016, he was a Post-Doctoral Fellow with the Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing. He is currently a Post-Doctoral Fellow with the Image and Video Understanding Laboratory, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. His current research interests include image/video restoration and compression, compressive sensing, and deep learning.

Dr. Zhang was a recipient of the Best Paper Award and the Best Student Paper Award of the IEEE International Conference on Visual Communication and Image Processing in 2011 and 2015.

Dong Xu (M'07–SM'13) received the B.Eng. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2005, respectively, where he is currently pursuing the Ph.D. degree.

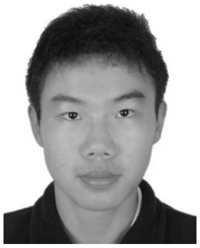
He was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Hong Kong, for more than two years. He is the Chair of Computer Engineering with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW, Australia. He was also a Postdoctoral Research Scientist with Columbia University, New York, NY, USA, from 2006 to 2007, and a Faculty Member with Nanyang Technological University, Singapore, from 2007 to 2015. He has published over 100 papers in the IEEE TRANSACTIONS and top tier conferences, including the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision, European Conference on Computer Vision, International Conference on Machine Learning, ACM Multimedia, and The Medical Image Computing and Computer Assisted Intervention Society.

Dr. Xu was a recipient of the Best Student Paper Award of the CVPR in 2010, the IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award in 2014, and the IEEE Computational Intelligence Society Outstanding Early Career Award for 2017. He is a Fellow of The International Association for Pattern Recognition.

Yun Fu (S'07–M'08–SM'11) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, Xi'an, China, respectively, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA.

He has been an Interdisciplinary Faculty Member with the College of Engineering and the College of Computer and Information Science, Northeastern University, Boston, MA, USA, since 2012. He has extensive publications in leading journals, books/book chapters, and international conferences/workshops. His current research interests include machine learning, computational intelligence, big data mining, computer vision, pattern recognition, and cyber-physical systems.

Dr. Fu was a recipient of seven Prestigious Young Investigator Awards from NAE, ONR, ARO, the IEEE, the International Neural Network Society (INNS), UIUC, and Grainger Foundation, seven Best Paper Awards from the IEEE, IAPR, SPIE, and SIAM, and three major Industrial Research Awards from Google, Samsung, and Adobe. He serves as an Associate Editor, the Chair, a Program Committee Member, and a Reviewer of several top journals and international conferences/workshops. He is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEANING SYSTEMS. He is a Fellow of IAPR, a Lifetime Senior Member of ACM and SPIE, a Lifetime Member of AAAI, The Optical Society, and the Institute of Mathematical Statistics, a member of Global Young Academy and INNS, and a Beckman Graduate Fellow from 2007 to 2008.



Yisen Wang received the B.S. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

His current research interests include machine learning, deep learning, and related applications.

Xiangyang Ji (M'10) received the B.S. degree in materials science and the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

He joined Tsinghua University, Beijing, in 2008, where he is currently a Professor with the Department of Automation, School of Information Science and Technology. He has authored over 100 referred conference and journal papers. His current research interests include signal processing, image/video compression and communication, and intelligent imaging.

Qionghai Dai received the M.S. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively.

He is currently a Professor with the Department of Automation and the Director of the Broadband Networks and Digital Media Laboratory, Tsinghua University, Beijing, China. He has authored or co-authored over 200 conference and journal papers and two books. His current research interests include computational photography and microscopy, computer vision and graphics, and intelligent signal processing.

Dr. Dai is an Associate Editor of the *Journal of Visual Communication and Image Representation*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON IMAGE PROCESSING.