Deep Network Cascade for Image Super-resolution

Zhen Cui[1,2], Hong Chang[1], Shiguang Shan[1], Bineng Zhong[2], and Xilin Chen[1]

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China
[2] School of Computer Science and Technology, Huaqiao University, Xiamen, China
{zhen.cui,hong.chang}@vipl.ict.ac.cn, {sgshan,xlchen}@ict.ac.cn, bnzhong@hqu.edu.cn

Abstract. In this paper, we propose a new model called deep network cascade (DNC) to gradually upscale low-resolution images layer by layer, each layer with a small scale factor. DNC is a cascade of multiple stacked collaborative local auto-encoders. In each layer of the cascade, non-local self-similarity search is first performed to enhance high-frequency texture details of the partitioned patches in the input image. The enhanced image patches are then input into a collaborative local auto-encoder (CLA) to suppress the noises as well as collaborate the compatibility of the overlapping patches. By closing the loop on non-local self-similarity search and CLA in a cascade layer, we can refine the super-resolution result, which is further fed into next layer until the required image scale. Experiments on image super-resolution demonstrate that the proposed DNC can gradually upscale a low-resolution image with the increase of network layers and achieve more promising results in visual quality as well as quantitative performance.

Keywords: Super-resolution, Auto-encoder, Deep learning.

1 Introduction

In visual information processing, high-resolution (HR) images are still desired for more useful information [26]. However, due to the limitation of physical devices, we can only obtain low-resolution (LR) images of the specific object in some scenes such as a long-distance shooting. To handle this problem, the super-resolution (SR) technique is usually employed to recover the lost information in the source image. With increasing applications in video surveillance, medical, remote sensing images, etc., the SR technique has been attracting more and more attention in the computer vision community over the past decades [7,9,33,10,8].

The conventional super-resolution methods attempt to recover the source image by solving the ill-posed inverse problem, $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$, where $\mathbf{x}$ is the unknown HR image to be estimated, $\mathbf{y}$ is the observed LR image, $\mathbf{H}$ is the degradation matrix, and $\mathbf{v}$ is the additional noise vector. Under the scarcity of observed LR images, the inverse process is a underdetermined problem, thus

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 49–64, 2014.

the solution is not unique. To find a reasonable solution, some sophisticated statistical priors of natural images are usually incorporated into the reconstruction process [7]. However, these reconstruction-based methods have a limit of magnification factor [1,21].

To address this problem, recently, example-based SR methods [33,10,5,8,32] have been proposed in succession. They use machine learning techniques to predict the missing frequency band of upsampled image from an external dataset [33] or the testing image itself [10,5,32]. Typically, in view of human visual mechanism, sparse representation (or sparse coding) is employed to super-resolution. They either emphasize on the construction of more representative dictionaries, such as learning coupled dictionaries of LR and HR counterpart patches [33] or multi-scale dictionaries [34], or focus on the robustness of sparse coefficients by integrating some priors, such as using centralized sparse constraints [5] and manifold structure constraints [22]. Generally, a huge training set is required to capture the rich characteristics of natural images for image super-resolution.

In contrary, some recent studies [3,10,32] indicate that local image structures tend to redundantly recur many times within and across different image scales, and super-resolution may be conducted on those self-similarity examples from the testing image itself. Specifically, Glasner *et al.* [10] utilized recurrence of patches to generate virtual LR image patches, which are fed into the classical reconstruction-based SR scheme. Yang *et al.* [32] further refined the self-similarity by in-place self-similarity. The self-similarity technique is empirically found to work well especially for a small upscaling factor. To robustly recover a LR image with a properly large scale, these few studies have also begun to more or less use an iterative strategy to upscale the LR image. Even so, more efficient gradual SR models still remain to be developed for image SR. Moreover, some crucial problems need to be studied in the gradual unscaling models, such as the propagation of estimated error, the collaboration of overlapping patches, etc.

In this paper, we propose a deep learning scheme called deep network cascade (DNC) to gradually upscale low-resolution images layer by layer, which is the first time to our knowledge. On one hand, to reasonably enhance high-frequency texture details, we employ non-local self-similarity (NLSS) search on the input image in multi-scale, which can bypass the assumption on the image degradation process. On the other hand, by taking the NLSS results as the input, the collaborative local auto-encoder (CLA) is proposed to suppress the noises and meanwhile collaborate the overlapping reconstructed patches. In CLA, to reduce the learnable parameters and make auto-encoder easily controllable, we adopt weight-tying on all patches and L_1 sparse constraint on hidden neurons. Closing the loop on the two steps forms a cascade layer, named stacked collaborative local auto-encoder (stacked CLA or SCLA), which refines the super-resolution image well.

Multiple SCLA models can be successively concatenated into the deep network cascade, where the higher layer takes the output SR image of the lower layer as input. With the increase of network layers, the magnification factor of the learned SR image can be enlarged gradually. Inevitably, in the deep network,

the synthetic "error" textures (e.g., noises, artifacts, etc.) will propagate and even spread in next network layers, which leads to a large deviation from the source HR image. To reduce the effects, we use the back-projected technique [14] to contrain the super-resolved image in each layer. To train the DNC model, we adopt the greedy layer-wise optimization strategy. Extensive experiments on single image super-resolution demonstrate the effectiveness of the proposed method on visual quality as well as objective evaluation.

2 Related Work

Generally, existing super-resolution methods fall into three categories: interpolation-based [35], reconstruction-based [7,27] and example-based [9,11,4,33,10,5,8,34,25,32] methods. Interpolation-based methods are simple and effective for SR, such as bilinear, bicubic or other sampling methods [35]. However, with the increasing magnification factor, they are prone to generate overly smooth edges. Reconstruction-based methods usually borrow a certain prior to predict the SR image, but they are still limited to small magnification factors [21] or a scarcity of observed LR images.

Example-based methods break the limitation. They attempt to learn the high-frequency details from an external training dataset or the testing image itself. Freeman *et al.* [9] used the nearest neighbor (NN) to estimate the high-frequency information and a Markov network to handle the compatibility between patches. Later on, by assuming the similarity of manifold structures between LR and HR counterparts, Chang *et al.* [4] used locally linear embedding (LLE) to predict the HR patches. More recently, the sparse coding based methods [6,33] were proposed for image restoration. Typically, the coupled filters (or dictionaries) [33] are learnt to share sparse structures on HR and LR counterparts, but it requires a large amount of training pairs. Aimed at this problem, the non-local prior may be employed to enrich the textural information [10,5,22]. They used the non-local prior with a designed degradation process [5] or a shallow model [22]. Specifically, the self-similarity prior of the testing image itself is used to generate virtual observed LR examples [10,32]. Empirically, the self-similarity works better on small upscaling factors [36,32]. To address this problem, a few studies [10,32] start to gradually upscale the LR image, but they lack a more explicit layer-wise model. Different from previous works, here we develop a new layer-wise model, referred to deep network cascade, to upscale the input LR image layer by layer, each layer with a refined SR result.

Deep learning attempts to learn layered, hierarchical representations of high-dimensional data [12,13], and has been successfully applied in many computer vision problems. Classical unit learning models in deep architecture include sparse coding [17], restricted Boltzmann machine (RBM) [12], anto-encoder [13,2,29], etc. Specifically, the (stacked) denoising auto-encoder (DA) [28,31] has shown effectiveness in image inverse problems such as denoising and inpainting. Our method differs from DA in two ways: first, DA requires clean data (ground truth) in the training process, assuming that the degradation function is implicitly known, while the CLA relaxes the condition by feeding the model with NLSS

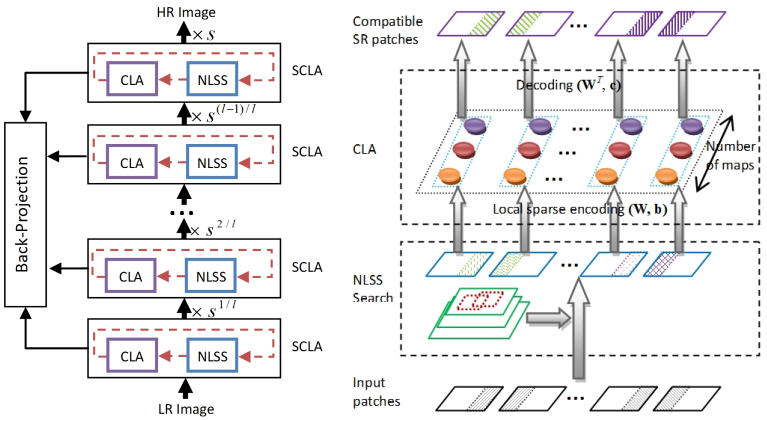
search results; second, our method is imposed on local patches with tied weight, sparse and compatibility constraints, which can greatly reduce the number of the learnable parameters when suppressing the noises.

3 The Proposed Network Cascade

Below we denote a vector/matrix by a lowercase/uppercase letter in bold. The transposition of a vector or matrix is denoted by the superscript $^\intercal$. We denote the input LR image by $\mathbf{x} \in \mathbb{R}^N$, and denote the extracted i-th ($i = 1, 2, \cdots, n$) patch from $\mathbf{x}$ by $\mathbf{x}_i = \mathcal{F}_i\mathbf{x} \in \mathbb{R}^d$, where $\mathcal{F}_i$ is the extracting patch operation (*i.e.*, a matrix in math). Given the LR image $\mathbf{x}$, the aim is to recover its HR image with a magnification factor s.

As shown in Fig.1(a), we upscale the LR image with the deep network of l cascade layers, each with a small scaling constant $s^{1/l}$. With the super-resolved image of the former layer as the input, we can successively stack l cascade layers to upscale the image. By doing this, we can obtain a stable SR solution for large scaling factors, while the reconstruction-based methods have a limit of a upscaling factor [21], and even the example-based methods usually work well on small upsampling factors [20,36,32]. However, as the mutuality only exists in two adjacent layers, minor distortions and estimation errors might propagate and accumulate from layer to layer, which easily leads to a large deviation from the source HR image for the final SR result. To reduce this effect, a global "back-projection" [14] constraint is used to make super-resolved images evolve along a proper direction. To do super-resolution more credibly in each network unit (or layer), we encapsulate two blocks: NLSS search and CLA. By iteratively stacking them as a cascade layer of DNC, referred to SCLA, the super-resolved image of each layer can be gradually refined. Below we further illustrate SCLA.

As discussed above, the image degradation process from the source image to the observed LR image often accompanies with complicated variations (*e.g.*, blur, downscaling, noise, etc.), and is always unknown in real-world problems. Therefore, it is intractable and impractical to learn the transformation format between HR and LR images, as many related methods do. To this end, we employ the NLSS prior on the input image itself to enhance textural high-frequency information. Since natural image patches recur many times within an image and even across different scales [10], we can always find some similar patches for a given patch. Concretely, in a network unit, we denote the input image by $\mathbf{x}$, which comes from the SR image of the former layer or the source LR image in the first layer. Before super-resolution, the bicubic interpolation is imposed on the input image to generate the initialized SR image (marked as $\mathbf{x}$ again for simplification). For a patch $\mathbf{x}_i$ extracted from $\mathbf{x}$, we perform the non-local self-similarity search in multi-scale images, which may come from blur and successive downscaling (*e.g.*, $s^{1/l}$ scaling factor) versions of $\mathbf{x}$. Given $\mathbf{x}_i$, suppose the



(a) Deep network cascade (b) Collaborative local auto-encoder

Fig. 1. Illustration of the proposed deep network cascade for image SR. Note that we don't plot the back-projection step in the right figure for simplification.

top K nearest neighbors, $\mathbf{x}_i^1, \cdots, \mathbf{x}_i^K$, are chosen from these multi-scale images, we can roughly estimate a new enhanced patch $\widehat{\mathbf{x}}_i$ as

$$\widehat{\mathbf{x}}_i = \sum_{j=1}^{K} \varpi_i^j \mathbf{x}_i^j, \tag{1}$$

where the weight ϖ_i^j may be set to Gaussian kernel with normalization. The estimated patch $\widehat{\mathbf{x}}_i$ usually contains more abundant texture information than the input patch $\mathbf{x}_i$.

With the high-frequency details generated from the NLSS search, the structure distortions or estimation errors also often accompany in the enhanced patches, and might be further propagated and even magnified in next layers of the deep network cascade. To relieve this phenomenon, an extension of auto-encoder, called CLA, is proposed to suppress the noises as well as collaborate the overlapping patches.

The CLA is adopted on the patches $\widehat{\mathbf{x}}_i (i = 1, \cdots, n)$ with constraints coming from two sides. First, a weight-tying scheme like convolutional network [19] is used to reduce the parameter space as well as preserve a certain flexibility to other variances. Moreover, a few of hidden neurons should be activated for a given stimulus (*i.e.*, a patch here) in view of human visual mechanism, which refers to the sparsity of codes. Second, the compatibility constraint on overlapping patches is added into auto-encoder to induce more smooth and natural textures for the integrated SR image. Actually, the output patch $\mathbf{z}_i$ of CLA can be combined

into a SR image $\widetilde{\mathbf{x}}$ by averaging the overlapping part among patches, which is formally computed from the following equation,

$$\widetilde{\mathbf{x}} = (\sum_{i=1}^{n} \mathcal{F}_i^{\mathsf{T}} \mathcal{F}_i)^{-1} \sum_{i=1}^{n} (\mathcal{F}_i^{\mathsf{T}} \mathbf{z}_i). \tag{2}$$

Due to the requirement of compatibility among patches, $\mathbf{z}_i$ should be ideally equal to $\mathcal{F}_i\widetilde{\mathbf{x}}$, which is regarded as the compatibility constraint presented in Section 4.

In addition, a pre-learned (denoising) auto-encoder trained with a large amount of patches from other external images or only the testing image may be used to initialize CLA. Given the enhanced image patches as the output of NLSS search, the revised auto-encoder network can be adapted to suppress the noises or artifacts. Furthermore, to avoid a large deviation and accelerate the optimization, we implicitly regularize the network parameters by feeding the learnt parameters of the former layer into the next layer. As a whole, the collaborative local auto-encoder plays an important role in accomplishing a more natural SR image with milder texture structures.

4 Collaborative Local Auto-encoder

In this section, we will first give the formulation of collaborative local auto-encoder, then provide a gradient-based optimization algorithm for CLA.

4.1 Formulation

Given the patches $\mathbf{x}_i (i = 1, \cdots, n)$ sampled from the LR image $\mathbf{x}$, we compute $\widehat{\mathbf{x}}_i \in \mathbb{R}^d$ through the NLSS search (Eqn. (1)), which are then input to CLA. As discussed in Section 3, CLA contains two constraints: the sparse constraint and the compatibility constraint. Concretely, CLA can be formulated into the following optimization problem,

$$\min_{\mathbf{W},\mathbf{b},\mathbf{c}} \quad l(\mathbf{x}, \mathbf{W}, \mathbf{b}, \mathbf{c}) + \gamma g(\mathbf{U}) + \eta h(\mathbf{Z}), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{m \times d}$ is the tied weights of network (m is the number of maps/filters, $m \gg d$), $\mathbf{b}, \mathbf{c}$ are respectively the bias of the encoder and decoder, $\mathbf{U}, \mathbf{Z}$ are the hidden neurons on the encoder and decoder level (or $\mathbf{Z}$ can be considered as the output of auto-encoder), γ and η are the balance parameters. Below we discuss the three terms in detail.

The first term represents the reconstruct error of auto-encoder on local patches. Like convolutional networks [19], each filter is tied on all patches within the image to form one map during encoding, which sharply reduces the number of learnable parameters while maintaining a certain flexibility by the over-complete filter bank.

Formally, the loss function can be written as,

$$l(\mathbf{x}, \mathbf{W}, \mathbf{b}, \mathbf{c}) = \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{z}_i - \widehat{\mathbf{x}}_i\|^2, \tag{4}$$

$$\mathbf{y}_i = \sigma(\mathbf{W}\widehat{\mathbf{x}}_i + \mathbf{b}), \tag{5}$$

$$\mathbf{u}_i = \frac{\mathbf{y}_i}{\sqrt{\mathbf{y}_i^\intercal \mathbf{y}_i}}, \tag{6}$$

$$\mathbf{z}_i = \mathbf{W}^\intercal \mathbf{u}_i + \mathbf{c}. \tag{7}$$

Eqn.(5) represents a nonlinear encoder with a point-wise hyperbolic tangent function $\sigma(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}$ (the gain a), which is easy to implement sparsity because its values range from -1 to 1. Eqn.(7) represents a decoder with the bias $\mathbf{c}$. In order to reduce the effect of filter scale, the L_2-normalization is performed on all hidden nodes of the encoder level as expressed in Eqn.(6).

The second term is the sparse constraint on all hidden neurons of the encoder level. Previous sparse deep learning methods [23,18,28,31] usually employ the deviation of the expected activation (*i.e.*, K-L divergence) to regularize the sparsity. However, we have to painstakingly tune the activation rate and balance parameter to reach a certain sparsity. Differed from those works, we directly use the L_1 norm, *i.e.*,

$$g(\mathbf{U}) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{u}_i\|_1, \tag{8}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n]$. The L_1 norm of the hyperbolic tangent operation can easily produce zero-value neurons and avoid tuning the extensive hyperparameters, which has been used in sparse filters with better effectiveness [24]. In addition, the L_1 norm on L_2 normalized features can be implemented through a few lines of MATLAB code.

The third term denotes the compatibility constraint on the reconstructed patches from the decoder. The compatibility on overlapping parts of the reconstructed patches is necessary for suppressing the artifacts. Ideally, the reconstructed patches and the corresponding patches extracted from the estimated image $\widetilde{\mathbf{x}}$ in Eqn. (2) should be as similar as possible. Formally, we can incorporate a regularization term into our model, *i.e.*,

$$h(\mathbf{Z}) = \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{z}_i - \mathcal{F}_i \widetilde{\mathbf{x}}\|^2, \tag{9}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n]$.

4.2 Optimization

To optimize the objective function in Eqn.(3), we employ the limited-memory BFGS (L-BFGS) method [16], which is often used to solve nonlinear optimization problems without any constraints. L-BFGS is particularly suitable for the problems with a large amount of variables under the moderate memory requirement. To utilize L-BFGS, the gradients of the object function need to be derived.

Algorithm 1. Image Super-resolution with DNC

Input: LR image $\mathbf{x}$, patch size p, upscale factor s, stacked layer number l, neighbor number K, balance parameters γ, η.

Output: SR image $\widetilde{\mathbf{x}}, \mathbf{W}, \mathbf{b}, \mathbf{c}$.

1. Set $t = 1$. Initialize $\mathbf{W}^t, \mathbf{b}^t, \mathbf{c}^t$ with (denoising) auto-encoder;
2. **repeat**
3. Initialize $\mathbf{x}^{(t)}$ by interpolating $\mathbf{x}^{(t-1)}$ ($\mathbf{x}^{(0)} = \mathbf{x}$) with a scale factor $s^{1/l}$.
4. **repeat**
5. Sample overlapping patches $\mathbf{X}^{(t)} = [\mathcal{F}_1\mathbf{x}^{(t)}, \cdots, \mathcal{F}_n\mathbf{x}^{(t)}]$ from $\mathbf{x}^{(t)}$.
6. Compute $\widehat{\mathbf{X}}^{(t)}$ in Eqn.(1) by using the NLSS search.
7. L-BFGS optimization for new $\mathbf{W}^{(t)}, \mathbf{b}^{(t)}, \mathbf{c}^{(t)}$ by using Eqn.(12).
8. Predict new SR image $\widetilde{\mathbf{x}}^{(t)}$ by using Eqn.(5), (6), (7) and (2).
9. Perform the back-projection operation for $\widetilde{\mathbf{x}}^{(t)}$.
10. $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)}; \mathbf{b}^{(t+1)} = \mathbf{b}^{(t)}; \mathbf{c}^{(t+1)} = \mathbf{c}^{(t)}; \mathbf{x}^{(t+1)} = \widetilde{\mathbf{x}}^{(t)}$.
11. $t = t + 1$.
12. **until** reach a satisfied solution.
13. **until** reach l layers.
14. **return** SR image $\mathbf{x}^{(t)}$.

However, the L_1 norm in Eqn. (8) is not first-order differentiable at zero value. For this, we use the soft-absolute function $\|x\|_1 = \sqrt{x^2 + \epsilon}$, where ϵ is a small constant (*e.g.*, $1.0E-9$). So Eqn. (8) can be rewritten as,

$$g(\mathbf{U}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_m^\intercal \sqrt{\mathbf{u}_i \otimes \mathbf{u}_i + \epsilon \mathbf{1}_m}, \tag{10}$$

where $\otimes$ is an element-wise multiplication operation and $\mathbf{1}_m$ is a column vector with m ones. Next we define some matrices to facilitate the derivation as listed in the following,

- $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2, \cdots, \widehat{\mathbf{x}}_n]$, a matrix of n patches obtained from NLSS search.
- $\widetilde{\mathbf{X}} = [\mathcal{F}_1\widetilde{\mathbf{x}}, \mathcal{F}_2\widetilde{\mathbf{x}}, \cdots, \mathcal{F}_n\widetilde{\mathbf{x}}]$, a matrix of n patches extracted from the reconstructed SR image in Eqn. (2).
- $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$, a matrix of the encoding neurons before L_2 normalization.
- $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n]$, a matrix of the hidden neurons after L_2 normalization.
- $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n]$, a matrix of the output units after decoding.

Thus the object function in Eqn.(3) can be reformulated as,

$$f(\mathbf{x}, \mathbf{W}, \mathbf{b}, \mathbf{c}) = \frac{1}{2n}\|\mathbf{Z} - \widehat{\mathbf{X}}\|_F^2 + \frac{\eta}{2n}\|\mathbf{Z} - \widetilde{\mathbf{X}}\|_F^2 + \frac{\gamma}{n}\mathbf{1}_m^\intercal(\sqrt{\mathbf{U} \otimes \mathbf{U} + \epsilon \mathbf{1}_{m\times n}})\mathbf{1}_n, \tag{11}$$

After a series of derivations, we can obtain the gradients of the function f with respect to the variables $\mathbf{W}, \mathbf{b}, \mathbf{c}$ as follows:

$$\frac{\partial f}{\partial \mathbf{W}} = \frac{1}{n}\mathbf{U}\mathbf{A}^\intercal + \frac{1}{n}\mathbf{C}\widehat{\mathbf{X}}^\intercal, \quad \frac{\partial f}{\partial \mathbf{b}} = \frac{1}{n}\mathbf{C}\mathbf{1}_n, \quad \frac{\partial f}{\partial \mathbf{c}} = \frac{1}{n}\mathbf{A}\mathbf{1}_n, \tag{12}$$

where

$$\mathbf{A} = (\mathbf{Z} - \widehat{\mathbf{X}}) + \eta(\mathbf{Z} - \widetilde{\mathbf{X}}), \quad \mathbf{B} = \mathbf{W}\mathbf{A} + \gamma\mathbf{U} \oslash \sqrt{\mathbf{U} \otimes \mathbf{U} + \epsilon}, \tag{13}$$

$$\mathbf{R} = \mathbf{1}_{m\times n} \oslash \left(\mathbf{1}_m \sqrt{\mathbf{1}_m^\intercal (\mathbf{Y} \otimes \mathbf{Y} + \epsilon)}\right), \quad \mathbf{Q} = \mathbf{1}_{m\times n} - \mathbf{Y} \otimes \mathbf{Y}, \tag{14}$$

$$\mathbf{C} = (\mathbf{B} - (\mathbf{1}_m(\mathbf{1}_m^\intercal(\mathbf{U} \otimes \mathbf{B}))) \otimes \mathbf{Y} \otimes \mathbf{R}) \otimes \mathbf{R} \otimes \mathbf{Q}. \tag{15}$$

In the above equations, $\otimes$ and $\oslash$ are respectively the element-wise multiplication and division operation.

4.3 Stacked CLA

The collaborative local auto-encoder is then stacked to get a deep architecture, where the input of each layer is obtained from the output SR results of the former layer. The whole SR process with SCLA is concluded in Algorithm 1.

5 Experiments

In this section, we evaluate the performance of the proposed method on the examples frequently-used in those SR literatures. Here we also consider two classic evaluation criterions: human visual quality, objective performance on PSNR and SSIM [30]. The magnification factor is set to 3 or 4 used in most SR literatures. Empirically, we also find the SR images with a larger magnification factor (*e.g.*, 6∼8) are satisfactory on visual performance. With an overlarge factor, the SR image will deviate from the ground-truth, which naturally leads to a worse quantitative performance. Due to space limitation, more SR results may be downloaded from the website: http://vipl.ict.ac.cn/paperpage/DNC/.

Experimental Configuration. We use patch size with 7×7 pixels, *i.e.*, the window size of the filter in CLA. The sampling step of patches is set to 2 pixels. The layer number of the network cascade is set to $l = 5$ as default. In the generation process of multi-scale images used for the NLSS search, we employ a low-pass Gaussian filter with a standard deviation of 0.55 as used in [32]. In the NLSS search, we choose the first nearest neighbor to predict the new patch. In CLA, the parameter a in the hyperbolic tangent function is set to 1, the number of filters (or maps) is set to 200, the sparse parameter γ and the collaborative parameter η are respectively set to 0.01 and 0.1. To suppress the noises in SR, we pre-train a denoising auto-encoder with Gaussian noises $\sigma = 1, 5, 10$ as the initial network of CLA, where the training data only contains the input image and its downscaled images. In addition, due to the sensibility to the luminance component, we transform RGB images into YCbCr images and then conduct super-resolution on the luminance channel of YCbCr, as most SR methods do.

5.1 Visual Performance

In the first experiment, we show the whole super-resolution process in visual performance with increasing layers. An example is shown in Fig.2. The input low-resolution image is very blur especially for those characters in the bottom lines.

Table 1. PSNR (dB) and SSIM results of the reconstructed high-resolution images on luminance components (3×). For each method, two columns are PSNR and SSIM respectively. The bold (*resp.* underlined) values denote the best (*resp.* second best.)

Images	Yang *et al.* [33]	Kim *et al.* [15]	Lu *et al.* [22]	DNC	
				NSLL	NSLL+CLA
Bike	24.01 / 0.773	24.43 / 0.784	23.78 / 0.767	24.29 / 0.782	**24.56** / **0.796**
Butterfly	26.16 / 0.877	27.09 / 0.894	25.48 / 0.857	26.64 / 0.894	**27.83** / **0.914**
Flower	28.73 / 0.837	28.91 / 0.832	28.30 / 0.829	28.89 / 0.840	**29.15** / **0.849**
Girl	33.38 / 0.823	33.00 / 0.807	33.13 / 0.819	33.43 / 0.823	**33.48** / **0.826**
Hat	30.45 / 0.858	30.67 / 0.847	30.29 / 0.854	30.83 / 0.863	**31.02** / **0.868**
Parrots	29.60 / 0.906	29.76 / 0.893	29.20 / 0.900	29.69 / 0.906	**30.18** / **0.913**
Parthenon	27.07 / 0.795	27.14 / 0.790	26.44 / 0.729	27.40 / 0.802	**27.49** / **0.811**
Plants	32.72 / 0.903	32.90 / 0.891	32.33 / 0.899	33.05 / 0.907	**33.13** / **0.913**
Raccoon	29.12 / **0.772**	29.03 / 0.760	28.81 / 0.758	29.05 / 0.767	**29.15** / **0.772**

We gradually upscale the LR image into a 3 factor HR image with five cascade layers, each with a equal scale factor[1] of $3^{1/5} \approx 1.25$. With the increase of layers, the characters become more and more clear. Compared with the state-of-the-art methods [10,15][2], DNC achieves comparable, even better visual performance for most SR characters, which have less noticeable artifacts. Note these result are best viewed in zoomed PDF.

In the next experiment, we compare our method with the recent state-of-the-art methods [33,10,15,8] in terms of visual quality. They either learn a transformation from LR patches to HR patches by external examples [33,15], or utilize the self-similarity of the input image itself to perform single image super-resolution [10,8]. We adopt those common testing examples in their literatures. Most results are quoted from the related literatures except [15] for which we use their released codes. Fig.3 shows the SR results of different methods on "child" by 4× and "cameraman" by 3×. The results of [10] appear to be overly sharp with some artifacts, *e.g.*, some ghost artifacts along the face contour in "child", and some jags in camera area of "cameraman". The SR images of [15] and [33] take on some smoothness on edges (*e.g.*, corner of mouth) and accompany with small artifacts along salient edges.

In addition, we also test our method on the images of natural landscapes, as shown in Fig.4. Those type of images usually contain diverse textures and rich fine structures. As shown in this figure, the textures super-resolved by [8] are a little blurry, as [32] does. In comparison, for our method, the restored edges are much sharper and clearer, and more textural structures are also recovered. As a whole, the SR images of our method look more natural.

[1] We also tried the setting where each layer magnifies the same number of pixels, which achieves a similar performance with the equal scaling setting.

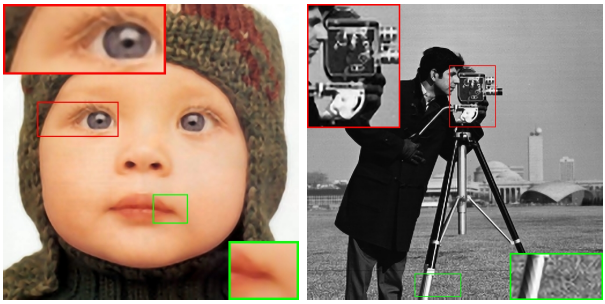
[2] Since Kim's method [15] uses additional samples to pre-learn a model, here we use the general model released from their website for fair comparison.

(a) Input (b) DNC-1 (c) DNC-2 (d) DNC-3

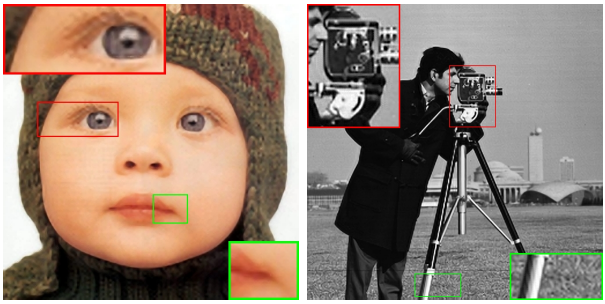
(e) DNC-4 (f) DNC-5

(g) Kim *et al.*[15] (h) Glasner *et al.*[10]

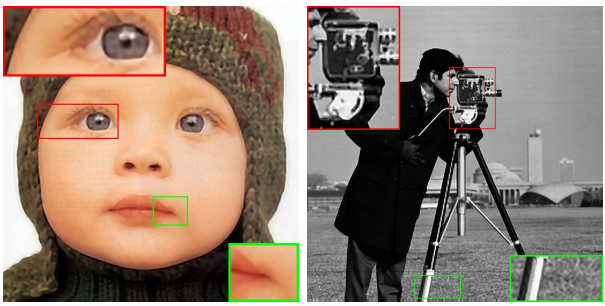
Fig. 2. An example of gradual SR by 3×. The proposed network cascade magnifies the input LR image layer by layer in (b)-(f). Please zoom PDF with 324 × 405 pixels.



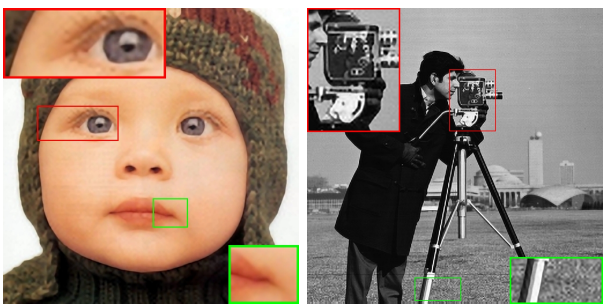
(a) Kim *et al.*[15]



(b) Yang *et al.*[33]

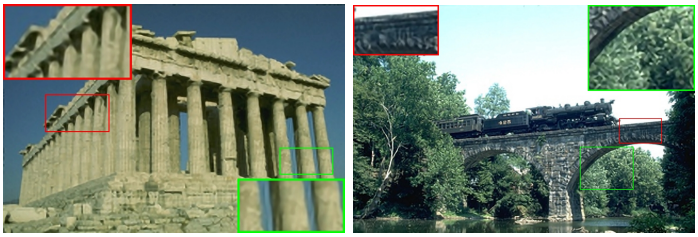


(c) Glasner *et al.*[10]

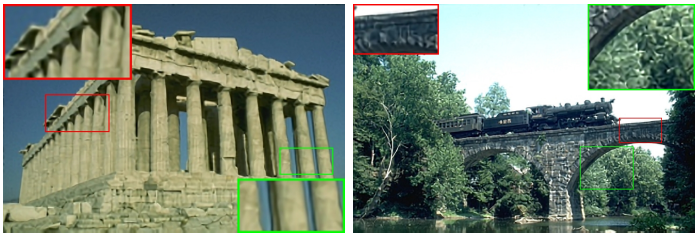


(d) DNC

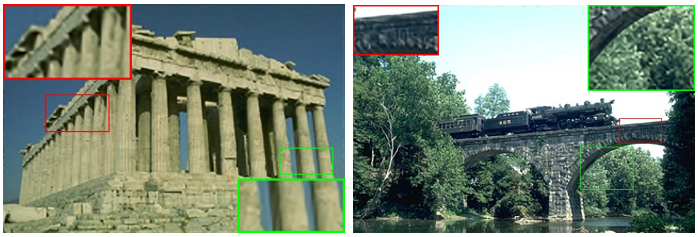
Fig. 3. Super-resolution results on "child" (4×) and "cameraman" (3×). Please zoom PDF with 512×512 and 768×768 pixels respectively for "child" and "cameraman".



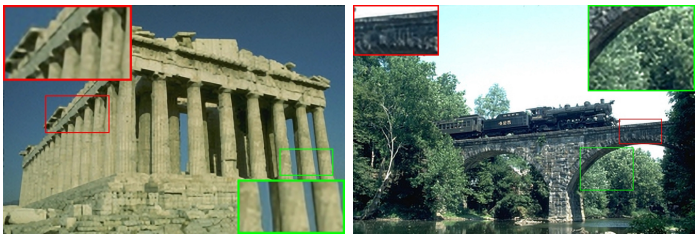
(a) Kim *et al.*[15]



(b) Freeman & Fattal[8]



(c) Yang *et al.*[32]



(d) DNC

Fig. 4. Super resolution results (3×). Results best viewed in zoomed PDF, at least 642×429 pixels for the former and 963×642 pixels for the latter.

5.2 Objective Evaluation

To evaluate the SR methods objectively, PSNR and SSIM [30] are used to measure their performance. Since the SR process is only imposed on the luminance channel of the color image, we only consider the quantitative difference on the luminance channel between the SR image and the original image. According to the protocol in [5], we conduct super-resolution on 9 images by a scale factor of 3. As shown in Tab.1, we also compare the other state-of-the art methods [33,15,22]. For our method, NLSS actually works in a deep model mode, *i.e.*, it stacks several NLSS even without auto-encoder, which we think accounts for its competitive results over these related methods with single-layer model. In addition, CLA also plays an important role in removing noises and artifacts especially for the deep network, though it doesn't bring a great improvement on PSNR and SSIM in most cases [3]. That is, CLA may efficiently suppress those noises and artifacts while NLSS produces rich textural details. As a whole, our method is more efficient to perform super-resolution due to its elaborate design in the production of texture structures and the suppression of noises.

Computational Efficiency. The computational cost of the proposed method mainly spends on the NLSS search and the computation of CLA. In practice, each cascade layer only needs to stack three CLAs (*i.e.*, loops), which can reach a satisfying solution. Currently, for each image in our experiments, it takes about several minutes without any optimized Matlab code on a general PC. However, the algorithm can easily be parallelized with GPU for fast processing, as those common deep learning algorithms do. In addition, we may speed up the NLSS search process in a local neighbor area like in-place self-similarity in [32].

6 Conclusion

In this paper, we propose a deep network cascade to conduct the image super-resolution problem layer by layer. In each layer, we elaborately integrate the non-local self-similarity search and collaborative local auto-encoder. The CLA can efficiently suppress artifacts and collaborate the compatibility among overlapping patches while the NLSS search enriches the textural detail of patches. By iteratively stacking the NLSS search and the CLA with the back-projection constraint, the super-resolution results can be refined, which also makes the SR information properly propagate in the network cascade. After the concatenation of multiple SCLA models, each with a small scale factor, the input LR image can be gradually upscaled into a more natural-looking HR image. Extensive experiments demonstrate the proposed method is more effective and more promising in the task of image super-resolution. In future work, we will consider how to accelerate the algorithm.

[3] The quantitative score as well as visual performance is actually very subtle for the competitive SR methods.

Acknowledgements. This work is partially supported by Natural Science Foundation of China under contracts Nos. 61025010, 61222211, 61202297, and 61272319.

References

1. Baker, S., Kanade, T.: Limits on super-resolution and how to break them. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(9), 1167–1183 (2002)
2. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems (NIPS), vol. 19, p. 153 (2007)
3. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2005)
4. Chang, H., Yeung, D.-Y., Xiong, Y.: Super-resolution through neighbor embedding. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, p. I–275 (2004)
5. Dong, W., Zhang, L., Shi, G.: Centralized sparse representation for image restoration. In: IEEE International Conference on Computer Vision (ICCV), pp. 1259–1266 (2011)
6. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing 15(12), 3736–3745 (2006)
7. Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P.: Fast and robust multiframe super resolution. IEEE Transactions on Image Processing 13(10), 1327–1344 (2004)
8. Freedman, G., Fattal, R.: Image and video upscaling from local self-examples. ACM Transactions on Graphics 30(2), 12 (2011)
9. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications 22(2), 56–65 (2002)
10. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: IEEE International Conference on Computer Vision (ICCV), pp. 349–356 (2009)
11. Gunturk, B.K., Batur, A.U., Altunbasak, Y., Hayes III, M.H., Mersereau, R.M.: Eigenface-domain super-resolution for face recognition. IEEE Transactions on Image Processing 12(5), 597–606 (2003)
12. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)
13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
14. Irani, M., Peleg, S.: Improving resolution by image registration. Graphical Models and Image Processing 53(3), 231–239 (1991)
15. Kim, K.I., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1127–1133 (2010)
16. Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Ng, A.Y.: On optimization methods for deep learning. In: International Conference on Machine Learning, ICML (2011)
17. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Advances in neural information processing systems (NIPS), vol. 19, p. 801 (2007)
18. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area v2. In: Advances in neural information processing systems (NIPS), vol. 20, pp. 873–880 (2008)

19. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: International Conference on Machine learning (ICML), pp. 609–616 (2009)
20. Lin, Z., He, J., Tang, X., Tang, C.-K.: Limits of learning-based superresolution algorithms. International Journal of Computer Vision 80(3), 406–420 (2008)
21. Lin, Z., Shum, H.-Y.: Fundamental limits of reconstruction-based superresolution algorithms under local translation. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(1), 83–97 (2004)
22. Lu, X., Yuan, H., Yan, P., Yuan, Y., Li, X.: Geometry constrained sparse coding for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1648–1655 (2012)
23. Ranzato, M., Boureau, Y.L., LeCun, Y.: Sparse feature learning for deep belief networks. In: Advances in neural information processing systems (NIPS), vol. 20, pp. 1185–1192 (2007)
24. Ngiam, J., Koh, P.W., Chen, Z., Bhaskar, S., Ng, A.Y.: Sparse filtering. In: Advances in Neural Information Processing Systems (NIPS), vol. 24, pp. 1125–1133 (2011)
25. Nguyen, K., Sridharan, S., Denman, S., Fookes, C.: Feature-domain super-resolution framework for gabor-based face and iris recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2642–2649 (2012)
26. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. Signal Processing Magazine 20(3), 21–36 (2003)
27. Tipping, M.E., Bishop, C.M.: Bayesian image super-resolution. In: Advances in Neural Information Processing Systems (NIPS), vol. 15, pp. 1279–1286 (2002)
28. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: International Conference on Machine learning (ICML), pp. 1096–1103. ACM (2008)
29. Wang, W., Cui, Z., Chang, H., Shan, S., Chen, X.: Deeply coupled auto-encoder networks for cross-view classification. arXiv preprint arXiv:1402.2031 (2014)
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error measurement to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
31. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 350–358 (2012)
32. Yang, J., Lin, Z., Cohen, S.: Fast image super-resolution based on in-place example regression. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2013)
33. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
34. Zhang, K., Gao, X., Tao, D., Li, X.: Multi-scale dictionary for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1114–1121 (2012)
35. Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Transactions on Image Processing 15(8), 2226–2238 (2006)
36. Zontak, M., Irani, M.: Internal statistics of a single natural image. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2011)