Fast Direct Super-Resolution by Simple Functions

Chih-Yuan Yang and Ming-Hsuan Yang
Electrical Engineering and Computer Science, University of California at Merced
{cyang35,mhyang}@ucmerced.edu

Abstract

The goal of single-image super-resolution is to generate a high-quality high-resolution image based on a given low-resolution input. It is an ill-posed problem which requires exemplars or priors to better reconstruct the missing high-resolution image details. In this paper, we propose to split the feature space into numerous subspaces and collect exemplars to learn priors for each subspace, thereby creating effective mapping functions. The use of split input space facilitates both feasibility of using simple functions for super-resolution, and efficiency of generating high-resolution results. High-quality high-resolution images are reconstructed based on the effective learned priors. Experimental results demonstrate that the proposed algorithm performs efficiently and effectively over state-of-the-art methods.

1. Introduction

Single-image super-resolution (SISR) aims to generate a visually pleasing high-resolution (HR) image from a given low-resolution (LR) input. It is a challenging and ill-posed problem because numerous pixel intensities need to be predicted from limited input data. To alleviate this ill-posed problem, it is imperative for most SISR algorithms to exploit additional information such as exemplar images or statistical priors. Exemplar images contain abundant visual information which can be exploited to enrich the super-resolution (SR) image details [4, 1, 5, 19, 6, 17, 3, 18]. However, numerous challenging factors make it difficult to generate SR images efficiently and robustly. First, there exist fundamental ambiguities between the LR and HR data as significantly different HR image patches may generate very similar LR patches as a result of downsampling process. That is, the mapping between HR and LR data is many to one and the reverse process from one single LR image patch alone is inherently ambiguous. Second, the success of this approach hinges on the assumption that a high-fidelity HR patch can be found from the LR one (aside from ambiguity which can be alleviated with statistical priors), thereby requiring a large and adequate dataset at our disposal. Third, the ensuing problem with a large dataset is how to determine similar patches efficiently.

In contrast, statistical SISR approaches [2, 16, 15, 9, 24, 20] have the marked advantage of performance stability and low computational cost. Since the priors are learned from numerous examples, they are statistically effective to represent the majority of the training data. The computational load of these algorithms is relatively low, as it is not necessary to search exemplars. Although the process of learning statistical priors is time consuming, it can be computed offline and only once for SR applications. However, statistical SISR algorithms are limited by specific image structures modeled by their priors (e.g., edges) and ineffective to reconstruct other details (e.g., textures). In addition, it is not clear what statistical models or features best suit this learning task from a large number of training examples.

In this paper, we propose a divide-and-conquer approach [25, 23] to learn statistical priors directly from exemplar patches using a large number of simple functions. We show that while sufficient amount of data is collected, the ambiguity problem of the source HR patches is alleviated. While LR feature space is properly divided, simple linear functions are sufficient to map LR patches to HR effectively. The use of simple functions also facilitates the process to generate high-quality HR images efficiently.

The contributions of this work are summarized as follows. First, we demonstrate a direct single-image super-resolution algorithm can be simple and fast when effective exemplars are available in the training phase. Second, we effectively split the input domain of low-resolution patches based on exemplar images, thereby facilitating learning simple functions for effective mapping. Third, the proposed algorithm generates favorable results with low computational load against existing methods. We demonstrate the merits of the proposed algorithm in terms of image quality and computational load by numerous qualitative and quantitative comparisons with the state-of-the-art methods.

2. Related Work and Problem Context

The SISR problem has been intensively studied in computer vision, image processing, and computer graphics. Classic methods render HR images from LR ones through certain mathematical formulations [13, 11] such as bicubic interpolation and back-projection [8]. While these algorithms can be executed efficiently, they are less effective to reconstruct high-frequency details, which are not modeled in the mathematical formulations.

Recent methods exploit rich visual information contained in a set of exemplar images. However, there are many challenges to exploit exemplar images properly, and many methods have been proposed to address them. To reduce the ambiguity between LR and HR patches, spacial correlation is exploited to minimize the difference of overlapping HR patches [4, 1, 19]. For improving the effectiveness of exemplar images, user guidance is required to prepare precise ones [19, 6]. In order to increase the efficiency of reconstructed HR edges, small scaling factors and a compact exemplar patch set are proposed by generating from the input frame [5, 3]. For increasing the chance to retrieve effective patches, segments are introduced for multiple-level patch searching [17, 6].

Statistical SISR algorithms learn priors from numerous feature vectors to generate a function mapping features from LR to HR. A significant advantage of this approach is the low computational complexity as the load of searching exemplars is alleviated. Global distributions of gradients are used [15] to regularize a deconvolution process for generating HR images. Edge-specific priors focus on reconstructing sharp edges because they are important visual cues for image quality [2, 16]. In addition, priors of patch mapping from LR to HR are developed based on dictionaries via sparse representation [24, 21], support vector regression [12], or kernel ridge regression [9].

Notwithstanding much demonstrated success of the algorithms in the literature, existing methods require computationally expensive processes in either searching exemplars [4, 1, 5] or extracting complex features [12, 16, 17, 6]. In contrast, we present a fast algorithm based on simple features. Instead of using one or a few mapping functions, we learn a large number of them. We show this divide-and-conquer algorithm is effective and efficient for SISR when the right components are properly integrated.

3. Proposed Algorithm

One motivation of this work is to generate SR images efficiently but also to handle the ambiguity problem. To achieve efficiency, we adopt the approach of statistical SISR methods, e.g., we do not search for a large exemplar set at the test phase. Compared with existing statistical methods using complicated features [16, 24, 21], we propose simple features to reduce computational load. To handle the ambiguity problem using simple features, we spend intensive computational load during the training phase. We collect a large set of LR patches and their corresponding HR source patches. We divide the input space into a large set of subspaces from which simple functions are capable to map LR features to HR effectively. Although the proposed algorithm entails processing a large set of training images, it is only carried out offline in batch mode.

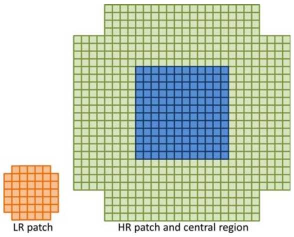


Figure 1. Training LR and HR pairs (four corner pixels are discarded). A set of functions is learned to map a LR patch to a set of pixels at the central (shaded) region of the corresponding HR patch (instead of the entire HR patch).

We generate a LR image I_l from a HR one I_h by

$$I_l = (I_h \otimes G) \downarrow_s, \tag{1}$$

where $\otimes$ is a convolution operator, G is a Gaussian kernel, $\downarrow$ is a downsampling operator and s is the scaling factor. From each I_h and the corresponding I_l image, a large set of corresponding HR and LR patch pairs can be cropped. Let P_h and P_l be two paired patches. We compute the patch mean of P_l as μ, and extract the features of P_h and P_l as the intensities minus μ to present the high-frequency signals. For HR patch P_h, we only extract features for pixels at the central region (e.g., the shaded region in Figure 1) and discard boundary pixels. We do not learn mapping functions to predict the HR boundary pixels as the LR patch P_l does not carry sufficient information to predict those pixels.

We collect a large set of LR patches from natural images to learn K cluster centers of their extracted features. Figure 2 shows 4096 cluster centers learned from 2.2 million natural patches. Similar to the heavy-tailed gradient distribution in natural images [7], more populous cluster centers correspond to smoother patches as shown in Figure 3. These K cluster centers can be viewed as anchor points to represent the feature space of natural image patches.

For some regions in the feature space where natural patches appear fairly rarely, it is unnecessary to learn mapping functions to predict patches of HR from LR. Since each cluster represents a subspace, we collect a certain number of exemplar patches in the segmented space to training a mapping function. Since natural images are abundant and easily acquired, we can assume that there are sufficient exemplar patches available for each cluster center.

Suppose there are l LR exemplar patches belonging to the same cluster. Let $\mathbf{v}_i$ and $\mathbf{w}_i$ ($i = 1, \ldots, l$) be vectorized features of the LR and HR patches respectively, in dimensions m and n. We propose to learn a set of n linear regression functions to individually predict the n feature values in HR. Let $\mathbf{V} \in \mathbb{R}^{m \times l}$ and $\mathbf{W} \in \mathbb{R}^{n \times l}$ be the matrices of $\mathbf{v}_i$ and $\mathbf{w}_i$. We compute the regression coefficients

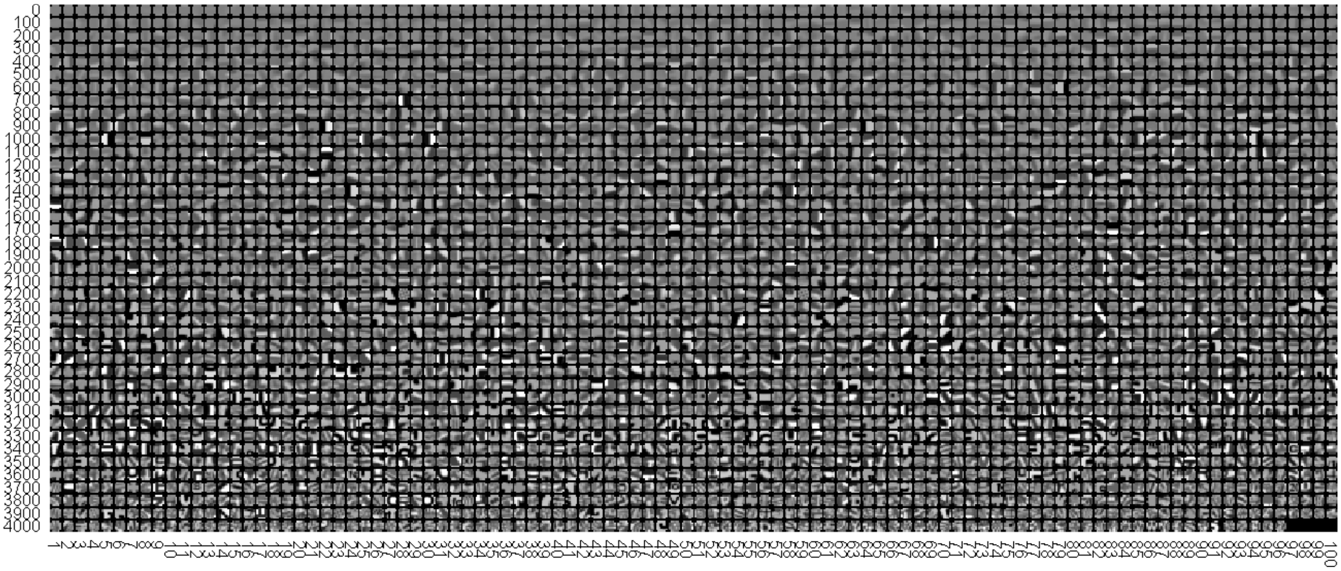


Figure 2. A set of 4096 cluster centers learned from 2.2 million natural patches. As the features for clustering are the intensities subtracting patch means, we show the intensities by adding their mean values for visualization purpose. The order of cluster centers is sorted by the amounts of clustered patches, as shown in Figure 3. Patches with more high-frequency details appear less frequently in natural images.

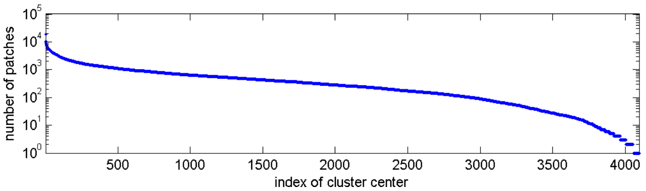


Figure 3. Histogram of clustered patches from a set of 2.2 million natural patches with cluster centers shown in Figure 2. While the most populous cluster consists of 18489 patches, the 40 least populous clusters only have one patch. A cluster has 537 patches on average.

$\mathbf{C}^* \in \mathbf{R}^{n\times(m+1)}$ by

$$\mathbf{C}^* = \underset{\mathbf{C}}{\operatorname{argmin}} \left\| \ \mathbf{W} - \mathbf{C} \begin{pmatrix} \mathbf{V} \\ \mathbf{1} \end{pmatrix} \ \right\|^2, \tag{2}$$

where $\mathbf{1}$ is a $1 \times l$ vector with all values as 1. This linear least-squares problem is easily solved.

Given a LR test image, we crop each LR patch to compute the LR features and search for the closest cluster center. According to the cluster center, we apply the learned coefficients to compute the HR features by

$$\mathbf{w} = \mathbf{C}^* \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}. \tag{3}$$

The predicted HR patch intensity is then reconstructed by adding the LR patch mean to the HR features.

The proposed method generates effective HR patches because each test LR patch and its exemplar LR patches are highly similar as they belong to the same compact feature subspace. The computational load for generating a HR image is low as each HR patch can be generated by a LR patch through a few additions and multiplications. The algorithm can easily be executed in parallel because all LR patches are upsampled individually. In addition, the proposed method is suitable for hardware implementations as only few lines of code are required.

4. Experimental Results

Implementation: For color images, we apply the proposed algorithm on brightness channel (Y) and upsample color channels (UV) by bicubic interpolation as human vision is more sensitive to brightness change. For a scaling factor 4, we set the Gaussian kernel width in Eq. 1 to 1.6 as commonly used in the literature [16]. The LR patch size is set as 7×7 pixels, and the LR feature dimension is 45 since four corner pixels are discarded. The central region of a HR patch is set as 12×12 pixels (as illustrated in Figure 1). Since the central region in LR is 3×3 pixels, a pixel in HR is covered by 9 LR patches and the output intensity is generated by averaging 9 predicted values, as commonly used in the literature [5, 24, 3, 21]. We prepare a training set containing 6152 HR natural images collected from the Berkeley segmentation and LabelMe datasets [10, 14] to generate a LR training image set containing 679 million patches.

Number of clusters: Due to the memory limitation on a machine (24 GB), we randomly select 2.2 million patches to learn a set of 4096 cluster centers, and use the learned cluster centers to label all LR patches in training image set. As the proposed function regresses features from 45 dimensions to one dimension only (each row of $\mathbf{C}^*$ in Eq. 2 is assumed to be independent) and most training features are highly similar, a huge set of training instances is unnecessary. We empirically choose a large value, e.g., 1000, as the size of training instances for each cluster center and col-

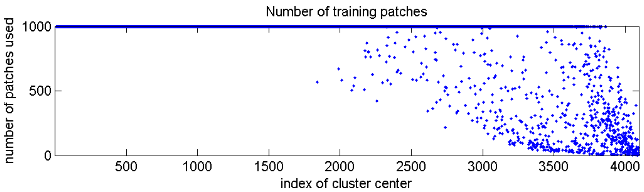
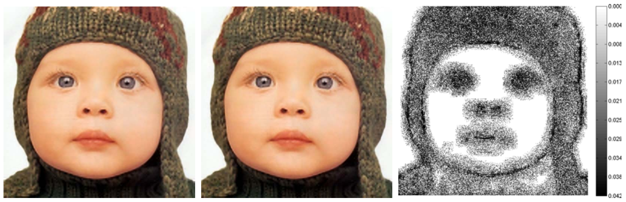


Figure 4. Numbers of patches used to train regression coefficients in our experiments. Since some patches are rarely observed in natural images, there are fewer than 1000 patches in some clusters.



(a) 512 clusters (b) 4096 clusters (c) Difference map

Figure 5. Super resolution results using different cluster numbers. Images best viewed on a high-resolution display where each image is shown with at least 512 × 512 pixels (full resolution).

lect training instances randomly from the labeled patches. Figure 4 shows the actual numbers of training patches. Since some patches are rarely observed in natural images, there are fewer than 1000 patches in a few clusters. For such cases we still compute the regression coefficients if there is no rank deficiency in Eq. 2, i.e., at least 46 linear independent training vectors are available. Otherwise, we use bilinear interpolation to map LR patches for such clusters.

The number of clusters is a trade-off between image quality and computational load. Figure 5 shows the results generated by 512 and 4096 clusters with all other same setup. While the low-frequency regions are almost the same, the high-frequency regions of the image generated by more clusters are better in terms of less jaggy artifacts along the face contours. With more clusters, the input feature space can be divided into more compact subspaces from which the linear mapping functions can be learned more effectively.

In addition to linear regressors, we also evaluate image quality generated by support vector regressor (SVR) with a Radial Basis Function kernel or a linear kernel. With the same setup, the images generated by SVRs and linear regressors are similar visually (See the supplementary material for examples). However, the computational load of SVRs is much higher due to the cost of computing the similarity between each support vector and the test vector. While linear regressors take 14 seconds to generate an image, SVRs take 1.5 hours.

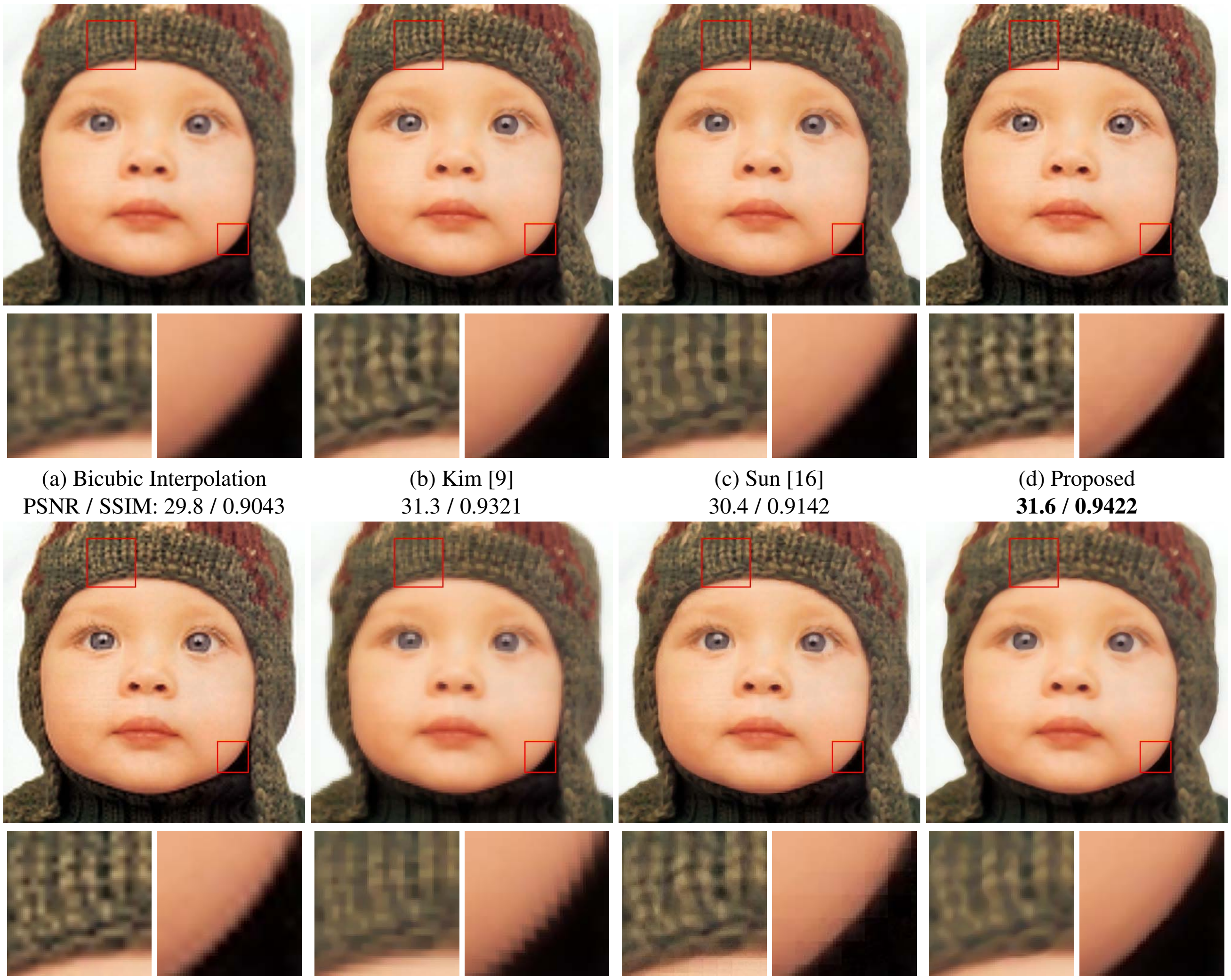
Evaluation and analysis: We implement the proposed algorithm in MATLAB, which takes 14 seconds to upsample an image of 128 × 128 pixels with a scaling factor 4 on a 2.7 GHz Quad Core machine. The execution time can be further reduced by other implementations and GPU. We use the released code from the authors [15, 24, 21] to generate HR images, and implement other state-of-the-art algorithms [8, 16, 5, 3] as the source code is not available. Our code and dataset are available at the project web page https://eng.ucmerced.edu/people/cyang35.

Table 1. Average evaluated values of 200 images from the Berkeley segmentation dataset [10]. While the generated SR images by the proposed method are comparable to those by the self-exemplar SR algorithm [5], the required computational load is much lower (14 seconds vs. 10 minutes).

Algorithm	PSNR	SSIM [22]
Bicubic Interpolation	24.27	0.6555
Back Projection [8]	25.01	0.7036
Sun [16]	24.54	0.6695
Shan [15]	23.47	0.6367
Yang [24]	24.31	0.6205
Kim [9]	25.12	0.6970
Wang [21]	24.32	0.6505
Freedman [3]	22.22	0.6173
Glasner [5]	**25.20**	0.7064
Proposed	25.18	**0.7081**

Figure 6-11 show SR results of the proposed algorithm and the state-of-the-art methods. More results are available in the supplementary material. We evaluate the method numerically in terms of PSNR and SSIM index [22] when the ground truth images are available. Table 1 shows averaged results for a set of 200 natural images. The evaluations are presented from the four perspectives with comparisons to SR methods using statistical priors [9, 16], fast SR algorithms [8, 15], self-exemplar SR algorithms [5, 3], and SR approaches with dictionary learning [24, 21].

SR methods based on statistical priors: As shown in Figure 6(b)(c), Figure 8(a), Figure 10(c), and Figure 11(b)(c), the proposed algorithm generates textures with better contrast than existing methods using statistical priors [9, 16]. While a kernel ridge regression function is learned in [9] and a gradient profile prior is trained in [16] to restore the edge sharpness based on an intermediate bicubic interpolated image, the high-frequency texture details are not generated due to the use of the bicubic interpolated intermediate image. Furthermore, a post-processing filter is used in [9] to suppress median gradients in order to reduce noise generated by the regression function along edges. However, mid-frequency details at textures may be wrongly reduced and the filtered textures appear unrealistic. There are several differences between the proposed method and the existing methods based on statistical priors. First, the proposed method upsamples the LR patches directly rather than using an intermediate image generated by bicubic interpolation, and thus there is no loss of texture details. Second, the proposed regressed features can be applied to any type of patches, while existing methods focus only on edges. Third, no post-processing filter is required in the proposed method



(a) Bicubic Interpolation
PSNR / SSIM: 29.8 / 0.9043

(b) Kim [9]
31.3 / 0.9321

(c) Sun [16]
30.4 / 0.9142

(d) Proposed
31.6 / 0.9422

(e) Back Projection [8]
PSNR / SSIM: 31.1 / 0.9391

(f) Shan [15]
27.8 / 0.8554

(g) Yang [24]
30.1 / 0.9152

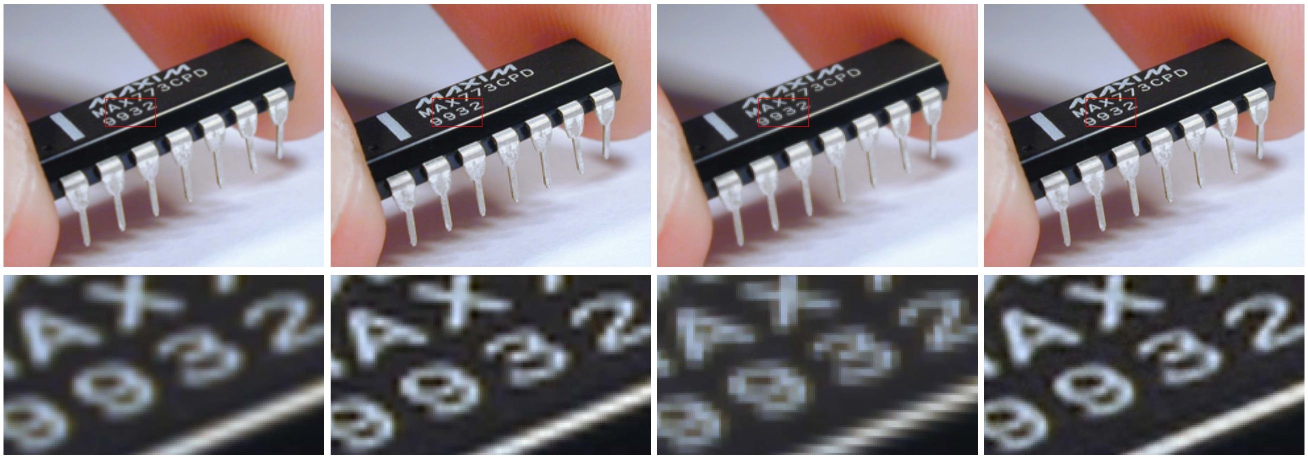
(h) Wang [3]
29.5 / 0.8859

Figure 6. Child. Results best viewed on a high-resolution display with adequate zoom level where each image is shown with at least 512 × 512 pixels (full resolution).

to refine the generated HR images. Fourth, existing methods learn a single regressor for the whole feature space, but the proposed method learns numerous regressors (one for each subspace), thereby making the prediction more effective.

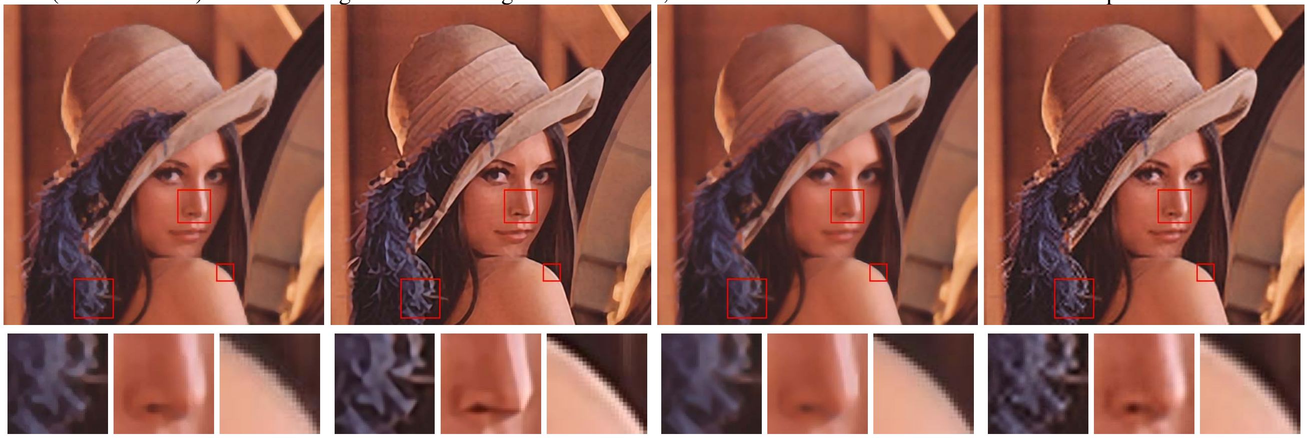
Fast SR methods: Compared with existing fast SR methods [8, 15] and bicubic interpolation, Figure 6(a)(e)(f), Figure 7, and Figure 11(a)(b) show that the proposed method generates better edges and textures. Although bicubic interpolation is the fastest method, the generated edges and textures are always over-smoothed. While back-projection [8] boosts contrast in SR images, displeasing jaggy artifacts are also generated. Those problems are caused by the fixed back-projection kernel, which is assumed isotropic. However, the image structures along sharp edges are highly anisotropic, and thus an isotropic kernel wrongly compensates the intensities. A global gradient distribution is exploited as constraints in [15] to achieve fast SR. However, although the global gradient distribution is reconstructed by [15] in Figure 6(f) and Figure 7(c), the local gradients are not constrained. Thus, over-smoothed textures and jaggy edges are generated by this method. The proposed method generates better edges and textures as each LR patch is upsampled by a specific prior learned from a compact subspace of similar patches. Thus, the contrast and local structures are better preserved with less artifacts.

SR methods based on self exemplars: Figure 8(b)(d), Figure 9(a)(d), Figure 10(b)(d), and Figure 11(c)(d) show the results generated by self-exemplar SR methods and the proposed algorithm. Self-exemplar SR algorithms [5, 3] iteratively upsample images with a small scaling factor (e.g., 1.25). Such an approach has an advantage of generating sharp and clear edges because it is easy to find similar edge



(a) Bicubic Interpolation (b) Back Projection [8] (c) Shan [5] (d) Proposed

Figure 7. IC. Results best viewed on a high-resolution display with adequate zoom level where each image is shown with at least 974 × 800 pixels (full resolution). Because the ground truth image does not exist, the PSNR and SSIM indexes can not be computed.

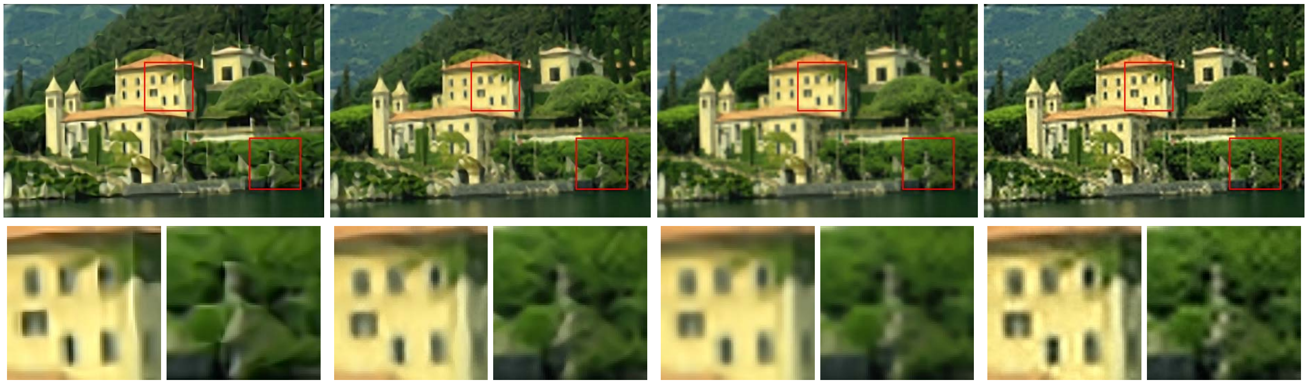


(a) Sun [16] PSNR / SSIM: 28.7 / 0.8827 (b) Glasner [5] 29.1 / 0.9002 (c) Wang [21] 28.5 / 0.8642 (d) Proposed **29.6 / 0.9071**

Figure 8. Lena. Results best viewed on a high-resolution display with adequate zoom level where each image is shown with at least 512 × 512 pixels (full resolution).

patches in the slightly downsampled input images. However, the exemplar patches generated by the input image are few. To reduce the errors caused by using a small exemplar patch set, the back-projection technique is facilitated as a post-processing in [5] to refine the generated image in each upsampling iteration. However, the post-processing may over-compensate SR images and generate artifacts, as shown in Figure 8(b) and Figure 11(c), the edges and textures are over-sharpened and unnatural. In addition, since all exemplar patches are generated from the input image, it entails a computationally expensive on-line process to find similar patches and makes the method less suitable for real-time applications. An simplified algorithm [3] reduces the computational load by searching local patches only, but the restriction also reduces the image quality. As shown in Figure 9(a), the structure of windows and the texture of bushes are distorted. The details of nose structure are almost lost in Figure 10(b) with unrealistic stripes near the hand and rock region. In contrast, the proposed method overcomes the difficulty of finding rare patches by using a huge exemplar set, which improves the probability to find similar edge patches. The proposed method exploits the well labeled edge patches in training phase to generated effective SR edges in test phase. As shown in Figure 6(d), Figure 7(d), Figure 8(d), Figure 10(d), and Figure 11(d), the proposed method generates SR images with sharp edges effectively.

SR methods based on dictionary learning: Figure 6(g)(h), Figure 8(c), and Figure 10(a) show images generated by SR algorithms based on sparse dictionary learning [24, 21]. The proposed algorithm is different from these algorithms in several aspects. The proposed algorithm splits the feature space and learns numerous mapping functions individually, but the existing algorithms [24, 21] learn one mapping function (through the paired dictionaries) for all patches. Therefore, the learned dictionaries may not capture



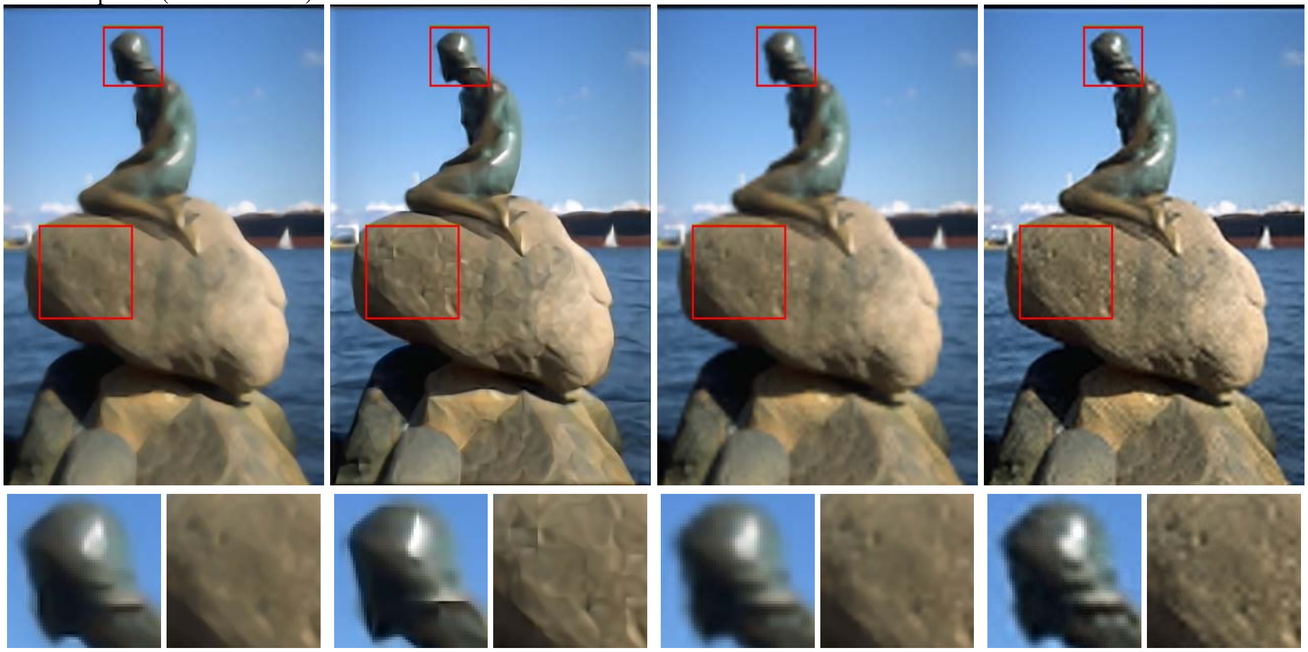
(a) Freedman [3]
PSNR / SSIM: 18.7 / 0.3876

(b) Kim [9]
20.4 / 0.4699

(c) Sun [16]
20.1 / 0.4395

(d) Proposed
20.5 / 0.4921

Figure 9. Mansion. Results best viewed on a high-resolution display with adequate zoom level where each image is shown with at least 480 × 320 pixels (full resolution).



(a) Wang [21]
PSNR / SSIM: 25.9 / 0.6746

(b) Freedman [3]
23.2 / 0.6374

(c) Sun [16]
25.8 / 0.6606

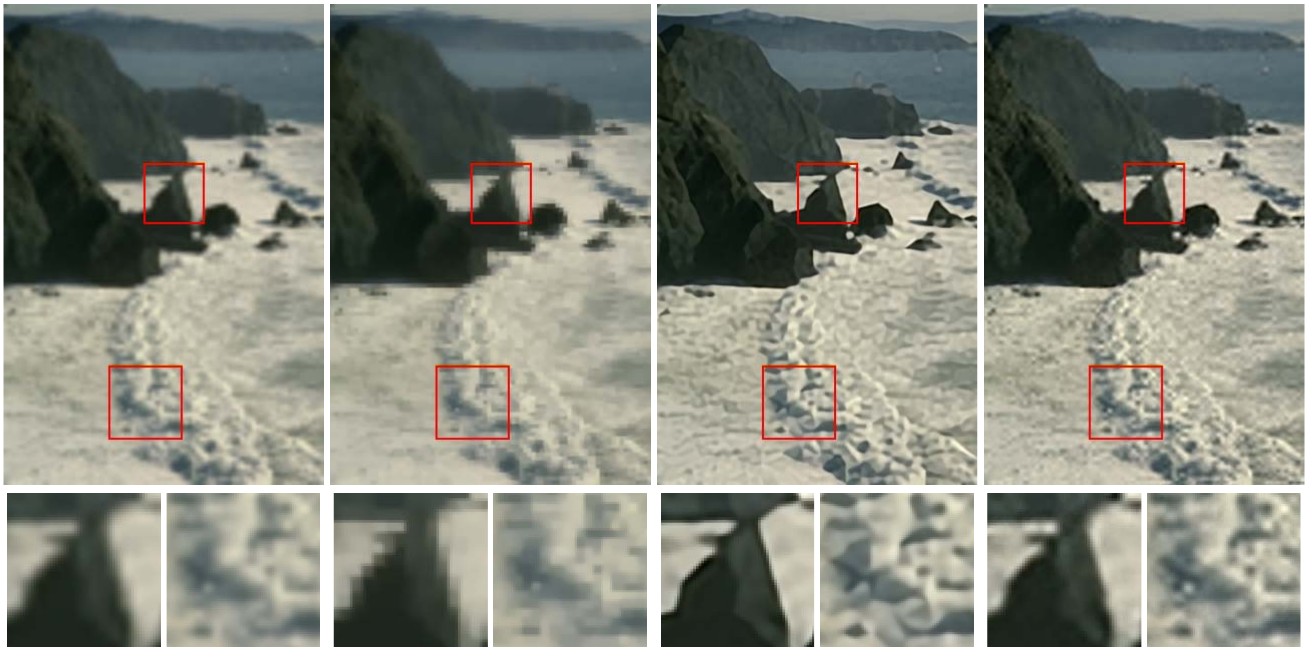
(d) Proposed
26.5 / 0.7133

Figure 10. Mermaid. Results best viewed on a high-resolution display with adequate zoom level where each image is shown with at least 320 × 480 pixels (full resolution).

the details from some infrequent patches. Since patches of sharp edges are less frequent than smooth patches in natural images, blocky edges can be observed in Figure 6(g). To improve the accuracy of patch mapping through a pair of dictionaries, an additional transform matrix is proposed in [21] to map LR sparse coefficients to HR ones. As shown in Figure 6(h), Figure 8(c) and Figure 10(a), the edges are sharp without blocky artifacts. However, the additional transform matrix blurs textures because the mapping of sparse coefficients becomes many-to-many rather than one-to-one, which results in effects of averaging. In contrast, the proposed method exploits the advantage of the divide-and-conquer approach to ensure each linear function effectively works in a compact feature subspace. Using simple feature and linear functions, the proposed method generates sharper edges than [24] and richer textures than [21], as shown in Figure 6(d)(g)(h), Figure 8(c)(d), and Figure 10(a)(d).

5. Conclusions

In this paper, we propose a fast algorithm which learns mapping functions to generate SR images. By splitting the feature space into numerous subspaces and collecting sufficient training exemplars to learn simple regression functions, the proposed method generates high-quality SR images with sharp edges and rich textures. Numerous experi-



(a) Bicubic Interpolation
PSNR / SSIM: 23.1 / 0.5241

(b) Shan [15]
22.4 / 4908

(c) Glasner [5]
23.7 / **0.5839**

(d) Proposed
23.8 / 0.5835

Figure 11. Shore. Results best viewed on a high-resolution display with adequate zoom level where each image is shown with at least 320 × 480 pixels (full resolution).

ments with qualitative and quantitative comparisons against several state-of-the-art SISR methods demonstrate the effectiveness and stability of the proposed algorithm.

6. Acknowledge

This work is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004.

[2] R. Fattal. Image upsampling via imposed edge statistics. In *SIGGRAPH*, 2007.

[3] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *TOG*, 30(2):1–11, 2011.

[4] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, pages 56–65, 2002.

[5] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009.

[6] Y. HaCohen, R. Fattal, and D. Lischinski. Image upsampling via texture hallucination. In *ICCP*, 2010.

[7] J. Huang and D. Mumford. Statistics of natural images and models. In *CVPR*, 1999.

[8] M. Irani and S. Peleg. Improving resolution by image registration. *CGVIP*, 53(3):231–239, 1991.

[9] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *PAMI*, 32(6):1127 –1133, 2010.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, 2001.

[11] P. Milanfar, editor. *Super-Resolution Imaging*. CRC Press, 2010.

[12] K. Ni and T. Nguyen. Image superresolution using support vector regression. *IEEE TIP*, 16(6):1596–1610, 2007.

[13] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, pages 21–36, 2003.

[14] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157 –173, 2008.

[15] Q. Shan, Z. Li, J. Jia, and C.-K. Tang. Fast image/video upsampling. In *SIGGRAPH Asia*, 2008.

[16] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *CVPR*, 2008.

[17] J. Sun, J. Zhu, and M. F. Tappen. Context-constrained hallucination for image super-resolution. In *CVPR*, 2010.

[18] L. Sun and J. Hays. Super-resolution from internet-scale scene matching. In *ICCP*, 2012.

[19] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin. Super resolution using edge prior and single image detail synthesis. In *CVPR*, 2010.

[20] J. Wang, S. Zhu, and Y. Gong. Resolution enhancement based on learning the sparse association of image patches. *Pattern Recognition Letters*, 31(1):1–10, 2010.

[21] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *CVPR*, 2012.

[22] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600 –612, 2004.

[23] H. Wei, X. Liang, W. Zhihui, F. Xuan, and W. Kai. Single image super-resolution by clustered sparse representation and adaptive patch aggregation. *China Communications*, 10(5):50–61, 2013.

[24] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE TIP*, 2010.

[25] S. Yang, M. Wang, Y. Chen, and Y. Sun. Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding. *IEEE TIP*, 21(9):4016–4028, 2012.