This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Convolutional Sparse Coding for Image Super-resolution

Shuhang Gu[1], Wangmeng Zuo[2], Qi Xie[3], Deyu Meng[3], Xiangchu Feng[4], Lei Zhang[1,*]
[1]Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[3]School of Mathematics and Statistics, Xi′an Jiaotong University, Xi′an, China
[4]Dept. of Applied Mathematics, Xidian University, Xi·an, China
{cssgu, cslzhang}@comp.polyu.edu.hk, cswmzuo@gmail.com,
qixie.liwu@stu.xjtu.edu.cn, dymeng@mail.xjtu.edu.cn, xcfeng@mail.xidian.edu.cn

Abstract

Most of the previous sparse coding (SC) based super resolution (SR) methods partition the image into overlapped patches, and process each patch separately. These methods, however, ignore the consistency of pixels in overlapped patches, which is a strong constraint for image reconstruction. In this paper, we propose a convolutional sparse coding (CSC) based SR (CSC-SR) method to address the consistency issue. Our CSC-SR involves three groups of parameters to be learned: (i) a set of filters to decompose the low resolution (LR) image into LR sparse feature maps; (ii) a mapping function to predict the high resolution (HR) feature maps from the LR ones; and (iii) a set of filters to reconstruct the HR images from the predicted HR feature maps via simple convolution operations. By working directly on the whole image, the proposed CSC-SR algorithm does not need to divide the image into overlapped patches, and can exploit the image global correlation to produce more robust reconstruction of image local structures. Experimental results clearly validate the advantages of CSC over patch based SC in SR application. Compared with state-of-the-art SR methods, the proposed CSC-SR method achieves highly competitive PSNR results, while demonstrating better edge and texture preservation performance.

1. Introduction

The purpose of super-resolution (SR) is to reconstruct a high resolution (HR) image from a single low resolution (LR) image or a sequence of LR images. SR provides a way to overcome the inherent resolution limitations of low-cost imaging sensors, and it also offers a solution to enhance the existing images which were generated by old type imaging equipment. Compared with SR from a sequence of images, single image SR (SISR) is more ill-posed because less information is provided. A key issue of single image SR is to build the relationship between the LR image and the HR image. Since information was lost in the down-sampling procedure, prior knowledge is needed to provide extra information for estimating the HR image. In the early years of studies, some simple smooth assumptions were utilized to estimate the missing pixels of the HR image, and different analytical interpolation methods have been proposed to zoom up LR images. However, such kind of simple smooth assumptions are far from enough for reconstructing complex structures in natural images.

The pioneer work in [8] proposed to use an external dataset and Markov random field (MRF) to model the image priors. Inspired by [8], many methods have been developed to model prior knowledge on local structures or patches using natural images [4, 7, 19, 29]. Methods in [7, 18, 19] learn the gradient distribution from high quality natural images to guide the HR estimation in the testing phase. Considering that natural images are complex and locally structured, instead of modeling the prior on the entire image, most SISR methods utilize the prior knowledge on image patches, which can be further grouped into three categories: example-based, mapping-based, and sparse coding-based methods. For example-based methods, both the external [3, 8, 27] dataset and internal cross-scale relationship [9] can be employed to provide examples of the LR and HR patch pairs. For mapping-based methods, mapping function between the LR and HR images is directly learned using the LR/HR patch pairs to implicitly incorporate prior knowledge [5, 6, 26, 11]. For sparse coding-based methods, motivated by the progress of sparse coding and dictionary learning, a couple of dictionaries are trained for LR and HR image patches, and several approaches have been suggested to model the relationship between the LR and HR patches in the coding vector domain [12, 22, 29].

Although patch based methods can greatly reduce the

*Corresponding author.

problem size and obtain state-of-the-art performance in SISR [25], previous studies usually process the overlapped patches independently, and the final results are achieved by averaging the overlapped pixels between each patch. It is commonly accepted that more overlapped pixels between neighboring patches will deliver better reconstruction results since each pixel in the output image will be estimated for more times. However, such an "overlap-averaging" mechanism ignores an important constraint in solving the patch estimation problem, i.e., pixels in the overlapped area of adjacent patches should be exactly the same (i.e., consistent). The consistency constraint provides prior information in dealing with each single estimation problem. Actually, in the seminal work of [8] the consistency prior is modeled by an MRF to select HR patches in the external database. Recently, researchers [13, 36] have proposed several elegant aggregation methods to alleviate the inconsistency of overlapped patches, and achieved significant performance improvement in image denoising. However, for SISR, more evidences and better approaches are still required to justify the importance of consistency constraint.

In this paper, we present a convolutional sparse coding (CSC) based SR method to demonstrate the effectiveness of consistency constraint and the advantage of global image based CSC over conventional patch based sparse coding. CSC was first proposed by Zeiler et al. [31]. Instead of sparsely representing a vector by the linear combination of dictionary atoms, CSC decomposes the input image into N sparse feature maps by N filters. The convolutional decomposition avoids dividing the whole image into overlapped patches and can naturally utilize the consistency prior in the decomposition procedure. CSC has been utilized in several works to extract features from images for object recognition [32]. However, compared with the great success of conventional patch based sparse coding, no work has been reported that CSC can achieve state-of-the-art performance in image reconstruction. In [2], extending the original patch based SC method to CSC for SR has been proposed as a potential application of CSC, but the authors only sketched out the idea and did not provide implementation details and experimental results. In [16], the author proposed to use a convolutional neural network (CNN) to approximate the CSC model for SR, the model is actually a CNN based SR model and the authors also did not compare the proposed method with state-of-the-art SR algorithms.

Previous joint dictionary learning methods encode decompose the interpolated LR image and use the corresponding HR dictionary to reconstruct the HR estimation. The LR and HR dictionaries have the same number of atoms. This scheme interpolation operation before sparse coding greatly increases the computation burden because we need to encode the interpolated image which has the same size of HR image. Furthermore, using the same number of atoms in the LR and HR dictionaries may limit the representation capacity of HR dictionary since HR images are much more complex than LR images. To address these problems, we use LR and HR filter groups which have different filter numbers and sizes to decompose and reconstruct the LR and HR images. A transformation mapping function is introduced to build the relationship between the LR and HR feature maps which have different sizes both in the spatial and coefficient domain. Such a mechanism not only reduces the computation burden of convolutional sparse coding in the LR image decomposition step, but also improves the representation capacity of HR filters to ensure the performance of our algorithm.

The contribution of this paper is three-fold. First, we show that compared with conventional sparse coding methods which process each overlapped patch independently, the global decomposition strategy in CSC is more suitable for image reconstruction. This may trigger new discussions on the commonly used overlapped patch dividing mechanism. Second, to take full advantage of the feature maps generated by the convolutional coding, we utilize the feature space information to train a sparse mapping function. Such a mechanism reduces the number of filters used to decompose the LR input image, and avoids performing sparse decomposition on the large interpolated image, greatly reducing the computation burden of convolutional sparse decomposition in the testing phase. Third, our experiments on commonly used test images show that the proposed method achieves very competitive SR results with the state-of-the-art methods not only in PSNR index, but also in visual quality.

2. Convolutional Sparse Coding

2.1. Sparse Coding for Super Resolution

Sparse representation encodes a signal vector $\boldsymbol{x}$ as the linear combination of a few atoms in a dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{x} \approx \boldsymbol{D}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the sparse coding vector. By far, sparse representation has achieved state-of-the-art results in various computer vision tasks [15, 24, 30]. As for single image super-resolution (SISR), Yang et al. first proposed a sparse coding super resolution (ScSR) method in [29]. In the training phase, given a group of low resolution (LR) and high resolution (HR) training patch pairs, ScSR aims to jointly learn an HR dictionary $\boldsymbol{D}^h$ and an LR dictionary $\boldsymbol{D}^l$ to reconstruct the HR and LR patches by assuming that each LR/HR patch pair shares the same sparse coding vector. In the testing phase, the input LR image is divided into overlapped patches, and each patch is encoded by the LR dictionary $\boldsymbol{D}^l$ with the sparse coefficient $\boldsymbol{\alpha}$. The corresponding HR patch is reconstructed by $\boldsymbol{D}^h$ and $\boldsymbol{\alpha}$ with $\boldsymbol{D}^h\boldsymbol{\alpha}$. Finally, the HR image can be obtained by aggregating all the estimated HR patches into a whole image.

Inspired by ScSR [29], many sparse coding and dictio-

nary learning based methods have been proposed for SISR. By relaxing the constraint that the LR/HR patch pair has the same coding vector, Wang et al. [22] introduced a transform matrix to allow more complex relationship between the HR and LR coding vectors, and proposed a semi-coupled dictionary learning (SCDL) method for SISR. Subsequently, more complex models have been proposed for better modeling the relationship between the LR and HR spaces with coupled dictionaries. He et al. [12] utilized a non-parametric Bayesian approach to learn dictionaries to build relationship between the LR and HR spaces. Peleg and Elad [17] proposed a statistical model which uses restricted Boltzmann machine (RBM) to model the relationship between the LR and HR coding vectors. Zhu et al. [35] suggested to enhance the flexibility of the HR dictionary by permitting certain deformation in each HR patch.

2.2. Convolutional Sparse Coding (CSC)

Despite its wide applications, sparse coding on an image patch has some drawbacks. First, the scalability of the ℓ_0 or ℓ_1 optimization is poor, which limits the application of sparse coding in large scale problems. Second, most of the previous sparse coding based methods partition the whole image into overlapped patches to reduce the burden of modeling and computation. However, the consistency between overlapped patches is ignored and the existing aggregation and averaging strategies can only alleviate this problem.

To take consistency into account, Zeiler et al. [31] proposed a convolutional implementation of sparse coding to sparsely encode the whole image. Instead of decomposing a signal vector as the multiplication of dictionary matrix and coding vector, the so-called convolutional sparse coding (CSC) model represents an image as the summation of convolutions of the feature maps and the corresponding filters:

$$\min_{\boldsymbol{Z}} \| \boldsymbol{X} - \sum_{i=1}^{N} \boldsymbol{f}_i \otimes \boldsymbol{Z}_i \|_F^2 + \lambda \sum_{i=1}^{N} \| \boldsymbol{Z}_i \|_1, \quad (1)$$

where $\boldsymbol{X}$ is an $m \times n$ input image, $\{\boldsymbol{f}_i\}_{i=1,2,\ldots,N}$ is a group of $s \times s$ filters, and $\boldsymbol{Z}_i$ is the feature map corresponding to $\boldsymbol{f}_i$ with size $(m+s-1) \times (n+s-1)$. In the CSC model, each feature map $\boldsymbol{Z}_i$ has nearly the same size as $\boldsymbol{X}$. The reconstruction is obtained by a summation (instead of averaging in patch-based model) of the convolution outputs $\boldsymbol{f}_i \otimes \boldsymbol{Z}_i$. Thus, the inconsistency problem in patch based implementation is avoided.

On the other hand, the convolutional decomposition mechanism also brings some difficulties in optimization. Zeile et al. [31] adopted the continuation method to relax the equality constraints, and employed the conjugate gradient (CG) decent to solve the convolutional least square approximation problem. Bristow et al. [1] proposed a fast CSC algorithm by considering the property of block circulant with circulant block (BCCB) matrix in the *Fourier* domain. Recently, Wohlberg [23] further improved this algorithm and proposed an efficient alternating direction method of multipliers (ADMM) for CSC.

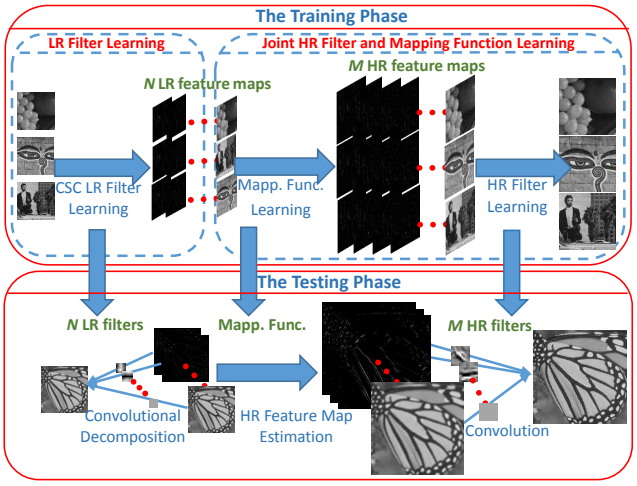


Figure 1. Flowchart of the proposed algorithm.

Despite the study of fast algorithms to solve the CSC problem, little attention was given on validating the advantages of CSC over conventional patch based sparse coding for image reconstruction. Can CSC really benefit image reconstruction? In this work, we attempt to answer this question and develop an effective CSC based SISR algorithm.

3. Convolutional Sparse Coding for Super Resolution

In this section, we present our convolutional sparse coding based super-resolution (CSC-SR) method. Like most existing SISR methods, the proposed CSC-SR method also involves a training phase and a testing phase. In the training phase, we learn three groups of parameters: (i) LR filters; (ii) the mapping function between LR and HR feature maps; and (iii) HR filters. In the testing phase, the input LR image is first decomposed into sparse LR feature maps by using the learned LR filters. Then, the mapping function is employed to estimate HR feature maps from LR feature maps, and the HR image is reconstructed by simple convolution operation. The flowchart of our algorithm in the training and testing phases is shown in Fig. 1.

3.1. The Training Phase

In dictionary learning based SISR, a couple of dictionaries together with certain mapping function are generally used to model the relationship between LR and HR images. On one hand, the LR and HR dictionary learning can be formulated into one objective function, and be jointly learned using the training LR/HR patch pairs. However, for the joint dictionary learning methods in [22, 29], because the test HR image is not available, the mechanism of generating coding vectors in training is different from that in

testing, leading to the inconsistency of the coding vectors in the training and testing phases. Several strict joint learning models [14, 28] have been developed to avoid coding inconsistency, but they need to solve a bi-level optimization problem. On the other hand, recent studies [33] also showed that encouraging SISR performance can be obtained via separate training of the LR and HR dictionaries. In [33], an LR dictionary is first learned using the LR image dataset, and then an HR dictionary is trained to reconstruct the HR patches based on the sparse coding vectors of the corresponding LR patches. In this work, we extend the method in [33] to CSC, and learn the LR and HR filters for SISR.

3.1.1 LR filter learning for CSC decomposition

Suppose that we are given a group of HR images $\{\mathbf{x}_1, \mathbf{x}_k, ..., \mathbf{x}_K\}$ together with the corresponding LR images $\{\mathbf{y}_1, \mathbf{y}_k, ..., \mathbf{y}_K\}$ for training. Because the index k does not affect the understanding of our algorithm, in the remainder of this paper, we omit it for the purpose of simplicity.

In order to obtain sparser feature maps, we decompose the LR image into one smooth component and one residual component before SR. The smooth component is simply enlarged by the bi-cubic interpolator, and the proposed CSC-SR model is performed on the residual component. Actually, similar strategy of pre-decomposition has been used in many previous SR works [8, 3, 29].

To extract the smooth component of the LR image $\mathbf{y}$, we first solve the following optimization problem:

$$\min_{\boldsymbol{Z}} \|\mathbf{y}-\boldsymbol{f}^s\otimes\boldsymbol{Z}_y^s\|_F^2+\gamma\|\boldsymbol{f}^{dh}\otimes\boldsymbol{Z}_y^s\|_F^2+\gamma\|\boldsymbol{f}^{dv}\otimes\boldsymbol{Z}_y^s\|_F^2, \quad (2)$$

where $\boldsymbol{Z}_y^s$ is the low frequency feature map of LR image $\mathbf{y}$, $\boldsymbol{f}^s$ is a 3×3 low pass filter with all coefficients being 1/9. $\boldsymbol{f}^{dh}$ and $\boldsymbol{f}^{dv}$ are the horizontal and vertical gradient operators $[1,-1]$ and $[1;-1]$. The closed form solution of (2) can be efficiently solved in the *Fourier* domain:

$$\boldsymbol{Z}_y^s=\mathbb{F}^{-1}\Big(\frac{\widehat{\mathcal{F}}^s\circ\mathbb{F}(\mathbf{y})}{\widehat{\mathcal{F}}^s\circ\mathcal{F}^s+\gamma\widehat{\mathcal{F}}^{dh}\circ\mathcal{F}^{dh}+\gamma\widehat{\mathcal{F}}^{dv}\circ\mathcal{F}^{dv}}\Big),$$

where $\mathbb{F}$ and $\mathbb{F}^{-1}$ are the FFT and inverse FFT operations, $\mathcal{F}^s$, $\mathcal{F}^{dh}$ and $\mathcal{F}^{dv}$ are the FFT transformations of $\boldsymbol{f}^s$, $\boldsymbol{f}^{dh}$ and $\boldsymbol{f}^{dv}$. Symbol "$\widehat{\ }$" means complex conjugate and "$\circ$" denotes component-wise multiplication. The division is also performed component-wisely. Having $\boldsymbol{Z}_y^s$, we can decompose the LR image as:

$$\mathbf{y}=\boldsymbol{f}^s\otimes\boldsymbol{Z}_y^s+\boldsymbol{Y},$$

where $\boldsymbol{f}^s\otimes\boldsymbol{Z}_y^s$ denotes the smooth component of the LR image, and $\boldsymbol{Y}$ denotes the residual component which represents the high frequency edge and texture structures in the LR image.

We then learn a group of LR filters to decompose the residual component $\boldsymbol{Y}$ into N feature maps:

$$\min_{\boldsymbol{Z},\boldsymbol{f}}\|\boldsymbol{Y}-\sum_{i=1}^N\boldsymbol{f}_i^l\otimes\boldsymbol{Z}_i^l\|_F^2+\lambda\sum_{i=1}^N\|\boldsymbol{Z}_i^l\|_1, \quad s.t.\ \|\boldsymbol{f}_i^l\|_F^2\le 1, \quad (3)$$

where $\{\boldsymbol{f}_i^l\}_{i=1\sim N}$ are N LR filters, and $\boldsymbol{Z}_i^l$ is the sparse feature map of the ith filter.

Similar to other dictionary learning methods, we alternatively optimize the $\boldsymbol{Z}$ and $\boldsymbol{f}$ subproblems. The $\boldsymbol{Z}$ subproblem is a standard CSC problem that can be solved using the algorithm proposed in [23]. For the $\boldsymbol{f}$ subproblem:

$$\boldsymbol{f}^l=\arg\min_{\boldsymbol{f}}\|\boldsymbol{Y}-\sum_{i=1}^N\boldsymbol{f}_i^l\otimes\boldsymbol{Z}_i^l\|_F^2,\ s.t.\ \|\boldsymbol{f}_i^l\|_F^2\le 1. \quad (4)$$

We can solve it by the ADMM algorithm in the *Fourier* domain [23].

However, when ADMM is employed to solve the (4), the feature maps of all the training images are required to be loaded in the memory. If the number of training images or the number of LR filters are large, the ADMM algorithm suffers from the problem of high memory demand for solving (4). Fortunately, the marriage of the recently developed stochastic average (SA) algorithms and ADMM, i.e., SA-ADMM[34], can be utilized to optimize (4). Different from standard ADMM, SA-ADMM adopts the linearization technique which can be deployed to avoid the computation of matrix inversion in our case, and utilizes the SA strategy to avoid the storage of feature maps of all the training images. More details of the optimization procedure can be found in the supplementary materials.

3.1.2 Joint HR filter and mapping function learning

After the LR filters learning, we further learn the mapping function and the HR filters based on the LR feature maps and the corresponding HR images. Like the LR images, each HR image is decomposed into one smooth component and one residual component. First, bi-cubic interpolation is adopted to enlarge $\boldsymbol{Z}_y^s$, obtaining the low frequency HR feature map $\boldsymbol{Z}_x^s$. Then, the original HR image can be decomposed as:

$$\mathbf{x}=\boldsymbol{f}^s\otimes\boldsymbol{Z}_x^s+\boldsymbol{X},$$

where $\boldsymbol{f}^s\otimes\boldsymbol{Z}_x^s$ denotes the smooth component, and $\boldsymbol{X}$ denotes the residual component which conveys the high frequency edge and texture structures of HR image $\mathbf{x}$. Given the training set of LR feature maps and HR images, we are able to learn the HR filters and the corresponding feature mapping function.

In most of the previous dictionary learning based SR methods, the LR image is first interpolated to the same size as the HR image, and the sizes of the HR and LR dictionaries are the same. In this work, we show that a small number of LR filters with small filter size can also achieve satisfactory SISR results while saving the decomposition time in both the training and the testing phases. Thus, we directly perform CSC on the small LR image that is much smaller than the HR image. Furthermore, since the HR image is much more complex than the LR image, we propose to decompose the LR image by a small number of LR filters

to reduce the computation burden, while reconstruct the HR image by a larger number of HR filters with more flexible representation capacity.

However, one challenge of the scheme above is that a mapping function needs to be trained to zoom the LR feature maps to a higher resolution in terms of both spatial size and feature map number. To this end, we propose to train a mapping function between the LR and HR feature maps:

$$\boldsymbol{Z}_j^h(kx, ky) = g(\boldsymbol{Z}_1^l(x,y), \boldsymbol{Z}_2^l(x,y), ..., \boldsymbol{Z}_N^l(x,y); \boldsymbol{W}), \quad (5)$$

where k is the zooming factor, $\boldsymbol{Z}_j^h(kx, ky)$ is the coefficient in position (kx, ky) of feature map $\boldsymbol{Z}_j^h$, $\boldsymbol{Z}_i^l(x,y)$ is the coefficient in the corresponding point (x,y) in feature map $\boldsymbol{Z}_i^l$, and $\boldsymbol{W}$ is the parameter of mapping function $g(\bullet)$. For $\boldsymbol{Z}_j^h(x', y')$ with $mod(x', k) \neq 0$ or $mod(y', k) \neq 0$, we simply set $\boldsymbol{Z}_j^h(x', y') = 0$.

The function $g(\bullet)$ should have the ability to generate sparse output from sparse input, and we use a sparse linear transformation matrix to estimate the HR coefficient:

$$\begin{aligned} \boldsymbol{Z}_j^h(kx, ky) = g(\boldsymbol{Z}_:^l(x,y); \boldsymbol{w}_j) = \boldsymbol{w}_j^T \boldsymbol{z}_:^l(x,y), \\ s.t. \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1, \end{aligned} \quad (6)$$

where $\boldsymbol{z}_:^l(x,y)$ is a vector containing all the coefficients in point (x,y) of the N LR feature maps, and $\boldsymbol{w}_j$ is the transformation vector for the HR feature map $\boldsymbol{Z}_j^h$. We let $\boldsymbol{w}_j \succeq 0$ and $|\boldsymbol{w}_j|_1 = 1$ to ensure the sparsity of $\boldsymbol{W}$. The non-negative simplex constraint used in (6) is stronger than some sparsity regularizer (e.g., ℓ_1 norm). Another thing needs to be noticed is that both the number and size of LR feature maps are enlarged by the mapping function. Compared with the coefficients in the LR feature map, each coefficient in the HR feature map includes the spatial information from a larger local area. The spatial size of HR filters should also be set larger to reconstruct the HR image.

After choosing the form of mapping function, our joint HR filter and mapping function learning model is formulated as:

$$\begin{aligned} \{\boldsymbol{f}^h, \boldsymbol{W}\} = \min_{\boldsymbol{f},\boldsymbol{W}} \| \boldsymbol{X} - \sum\nolimits_{j=1}^{M} \boldsymbol{f}_j^h \otimes g(\boldsymbol{Z}_:^l; \boldsymbol{w}_j) \|_F^2, \\ s.t. \ \| \boldsymbol{f}_j^h \|_F^2 \leq e; \quad \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1, \end{aligned} \quad (7)$$

where e is a scalar to constraint the energy of HR filters. Since the size of HR filter is different from the size of LR filter, the energy constraint should also be different. We optimize the objective function by alternatively updating the filter $\boldsymbol{f}^h$ and the mapping function parameter $\boldsymbol{W}$. For fixed $\boldsymbol{W}$, the filter updating subproblem defined in (4) can be solved by the SA-ADMM algorithm. For fixed $\boldsymbol{f}$, the subproblem on $\boldsymbol{W}$ is more complex, and we need to solve the following optimization problem:

$$\begin{aligned} \{\boldsymbol{W}\} = \arg\min_{\boldsymbol{W}} \| \boldsymbol{X} - \sum\nolimits_{j=1}^{M} \boldsymbol{f}_j^h \otimes g(\boldsymbol{Z}_:^l; \boldsymbol{w}_j) \|_F^2, \\ s.t. \ \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1. \end{aligned} \quad (8)$$

We also solve (8) by the SA-ADMM algorithm. Please refer to the supplementary file for details of the optimization.

Algorithm 1 The Training Algorithm for Convolutional Sparse Coding based Super Resolution (CSC-SR)

Input: Training image pairs $\{\mathbf{x}, \mathbf{y}\}$, LR&HR filter number N and M, LR&HR filter sizes s^l and s^h, regularization parameter γ and λ;

1: Solve (2) to decompose LR image, get high frequency component $\boldsymbol{Y}$ of LR image;

2: Solve the CSC filter learning problem on $\boldsymbol{Y}$, get $\boldsymbol{Z}^l$ and $\boldsymbol{f}^l$;

3: Extract low frequency component of the bi-cubic interpolated LR image, get the texture structure of HR image $\boldsymbol{X}$;

4: Learn the HR filters and the mapping function;

Output: LR filters $\boldsymbol{f}^l$, HR filters $\boldsymbol{f}^h$ and mapping function $\boldsymbol{W}$

With the optimization algorithms for solving the $\boldsymbol{f}$ and $\boldsymbol{W}$ subproblems, we summarize the training algorithm for our CSC-SR method in Algorithm 1.

3.2. The Testing Phase

After training, we have the LR filters $\{\boldsymbol{f}^l\}$, HR filters $\{\boldsymbol{f}^h\}$ and the mapping function $g(\bullet; \boldsymbol{W})$. Given a testing LR image $\mathbf{y}$, we extract its texture structure and decompose it by the LR filters to get LR sparse feature maps $\{\boldsymbol{Z}^l\}$. Then, the HR feature map can be estimated by $\{\boldsymbol{Z}^h\} = g(\boldsymbol{Z}^l; \boldsymbol{W})$. Finally, the high frequency texture structure in the HR output image is obtained by the summation of the convolutions of HR feature maps and the corresponding HR filters:

$$\widehat{\boldsymbol{X}} = \sum\nolimits_{j=1}^{M} \boldsymbol{f}_j^h \otimes \boldsymbol{Z}_j^h. \quad (9)$$

We can then combine $\widehat{\boldsymbol{X}}$ with the smooth component to generate the final HR estimation. To achieve better SR performance, the back projection operation which is widely used in other sparse coding based SR methods [12, 29, 35] can also be utilized to improve the final HR estimation.

4. Experimental Results

In this section, we first provide a brief convergence analysis of the proposed training algorithm. Then, we present an experiment to illustrate the advantages of the convolutional decomposition mechanism over the patch based method. After a discussion of parameter setting, we compare our algorithm with representative SR methods. The Matlab code of the proposed CSC algorithm can be downloaded at http://www4.comp.polyu.edu.hk/~cslzhang/code/CSC_SR.Zip.

The experimental setting in this paper is the same as [29]. LR training and testing images are generated by resizing the HR groundtruth image by bi-cubic interpolation. Using the same 91 training images provided by Yang et al. [29], we randomly crop 1,000 72×72 smaller images from these images to train our model. To avoid boundary effects of *Fourier* domain implementation, 8 pixels are padded on the image boundary.

4.1. Convergence Analysis

In our CSC-SR training algorithm, apart from training filters to decompose the LR images, model (7) is also pro-

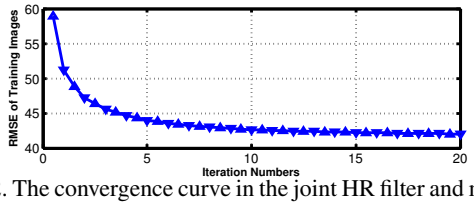


Figure 2. The convergence curve in the joint HR filter and mapping function training.

posed to jointly train the HR filters and mapping function. The objective function in (7) is a bi-convex optimization problem [10]. For fixed $\boldsymbol{W}$, the problem is convex to $\boldsymbol{f}$, and for fixed $\boldsymbol{f}$, the function is convex to $\boldsymbol{W}$. We alternatively optimize the $\boldsymbol{f}$ and the $\boldsymbol{W}$ sub-problems, which is actually an alternate convex search (ACS) algorithm [10]. Since our objective function has a general lower bound 0, if we can get the optimal solutions of updating $\boldsymbol{W}$ and $\boldsymbol{f}$, the joint HR filter and mapping function training is guaranteed to converge in terms of function energy.

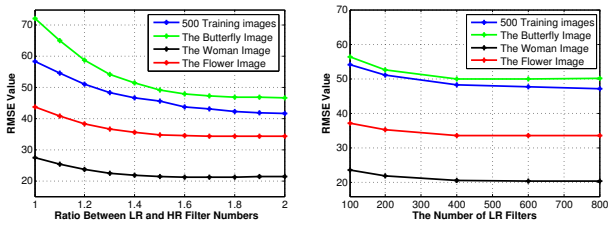
It is empirically found that the optimization of joint HR filter and mapping function training converges rapidly. Fig. 2 shows the convergence curve of our algorithm in an experiment with 200 training images. Because the energy of objective function is proportional to the number of pixels in training images, in Fig. 2 the energy of objective function is normalized by the pixel number of training images. The symbol "$\triangle$" represents the root mean square error (RMSE) between the training images and their HR estimates after updating filters $\boldsymbol{f}$ and the symbol "$\bigtriangledown$" shows the RMSE after updating the mapping function $g(\bullet; \boldsymbol{W})$. In most of our experiments, our algorithm will converge in 10 iterations.

4.2. CSC *vs.* Sparse Coding for SR

Table 1. SR results (PSNR, dB) by patch based sparse coding method ScSR [29] and the proposed convolutional based sparse coding method (without mapping function learning)

	Zooming Factor 2			
	ScSR_{256}	ScSR_{512}	CSC_{256}	CSC_{512}
Butterfly	30.43	31.10	30.97	31.56
Bird	40.02	40.44	40.20	40.51
Comic	27.75	27.98	27.90	28.10
Woman	34.48	34.89	34.62	34.99
Foreman	36.18	36.49	36.46	36.56

To validate our argument that global decomposition by convolution is more appropriate for SR, we compare the convolutional based CSC and a representative patch based sparse coding (SC) method. The ScSR [29] method is a classical patch based SC method for SR. It trains a pair of HR and LR dictionaries on the training set, and uses the sparse coding coefficients of the LR image to reconstruct the HR image by the HR dictionary. To have a fair comparison between CSC and SC methods, we omit the mapping function introduced in our method, and train a pair of LR filters and HR filters to reconstruct the LR and HR images with the same representation feature map. In the testing phase, we decompose the interpolated LR image and use exactly the same feature map to reconstruct the HR estimation. The SR



(a) (b)

Figure 3. (a) The RMSE values with different HR/LR filter number ratio on the training dataset and 3 testing images. (b) The RMSE values with different LR filter number on the training dataset and 3 testing images.

resluts (PSNR) by different methods (with dictionary size 256 and 512) on 5 images are shown in Table 1. The results on other images are similar. We see that CSC-SR is always better than ScSR with the same number of dictionary atoms.

4.3. Parameters Setting

A key parameter in all the dictionary based image reconstruction methods is the number of dictionary atoms. With a large dictionary atoms number, we are able to capture the image sparsity property better, but suffer from heavier space and time complexity. Here, we validate the effectiveness of using different filter numbers and choose an appropriate ratio between the LR filter number N and HR filter number M. We train different models on 500 images. The number of LR filters are fixed to 200 and the ratio between HR and LR filters is set from 1 to 2 with step length 0.1. The RMSE values on training images and 3 testing images are shown in Fig.3 (a). Compared with ratio 1, using more HR filters can provide better HR estimation. In all of our following experiments, we set the ratio between HR filter number and LR filter number as 3/2 to make a balance between SR performance and algorithm complexity.

Besides the ratio between LR and HR filter numbers, another important parameter in our algorithm is the number of LR filter number. We test a wide range of LR filter numbers with 500 training images, and the SR results with different LR filter numbers are shown in Fig.3 (b). Generally, the larger LR filter number leads to better SR results. To achieve the best performance, we train 800 LR filters in the following experiments.

Other parameters include the size of LR and HR filters, regularization parameters γ and λ, and the HR filter energy constraint parameter e. In all our following experiments, we set the size of LR filter as 5 and set the size of HR filter as $5\times$*Zooming Factor*. The regularization parameters γ and λ are set as 30 and 0.02, and the energy constraint parameter e is set as 4 and 9 for zooming factor 2 and 3, respectively.

4.4. Comparison with State-of-the-Arts

In this section, we compare the proposed CSC-SR methods with several state-of-the-art SR methods, including Sc-

Table 2. Super resolution results (PSNR, dB) by different methods.

	Zooming Factor=2									Zooming Factor=3								
	LLE	**ScSR**	**Zeyde**	**ANR**	**BPJDL**	**DPSR**	**CNN**	**A+**	**CSC**	**LLE**	**ScSR**	**Zeyde**	**ANR**	**BPJDL**	**DPSR**	**CNN**	**A+**	**CSC**
Butterfly	28.99	31.33	30.91	30.65	31.43	31.28	**32.20**	31.94	31.96	24.94	26.31	26.10	25.99	26.42	27.01	**27.58**	27.22	27.11
Face	35.57	35.69	35.69	35.70	**35.75**	35.64	35.60	35.72	35.71	33.40	33.52	33.62	33.65	33.45	33.61	33.57	33.74	**33.80**
Bird	38.93	40.53	40.25	40.23	40.98	39.77	40.63	40.98	**41.49**	33.84	34.42	34.75	34.70	34.53	34.82	34.92	34.48	**35.78**
Comic	27.34	28.02	27.89	27.92	28.24	27.98	28.28	28.29	**28.43**	23.79	24.09	24.10	24.12	24.13	24.24	24.40	24.39	**24.42**
Woman	33.71	34.95	34.74	34.70	35.23	34.64	34.93	35.27	**35.31**	29.63	30.38	30.53	30.42	30.50	30.75	30.92	31.19	**31.27**
Foreman	35.43	36.89	36.27	36.44	36.49	36.84	36.19	**36.91**	36.64	32.16	33.23	33.18	33.19	32.91	33.91	33.34	34.22	**34.24**
Coast.	30.30	30.60	30.61	30.56	30.63	30.55	30.49	30.60	**30.65**	27.03	27.01	27.22	27.13	27.07	27.20	27.19	**27.27**	**27.27**
Flowers	31.72	32.73	32.52	32.43	32.91	32.49	33.04	33.03	**33.15**	28.07	28.53	28.59	28.57	28.62	28.81	28.98	**29.05**	**29.05**
Zebra	32.54	33.46	33.58	33.36	33.62	33.22	33.30	33.67	**33.77**	28.00	28.60	28.80	28.65	28.73	28.97	28.90	29.06	**29.30**
Lena	35.95	36.46	36.39	36.42	36.58	36.27	36.48	36.57	**36.66**	32.60	33.04	33.15	33.15	33.13	33.25	33.39	33.50	**33.62**
Bridge	27.43	27.67	27.70	27.62	27.77	27.58	27.70	27.78	**27.84**	24.92	25.04	25.11	25.05	24.99	25.08	25.07	25.17	**25.20**
Baby	38.31	38.41	38.46	**38.55**	38.54	38.37	38.41	38.43	38.48	34.84	34.95	35.23	35.20	35.15	35.24	35.00	35.14	**35.28**
Peppers	35.82	36.72	36.60	36.38	36.71	36.55	36.75	**37.06**	36.90	33.15	33.88	34.13	33.88	34.02	34.29	34.35	34.71	**34.72**
Man	30.14	30.70	30.60	30.57	30.80	30.60	30.82	30.88	**30.97**	27.62	27.98	28.01	27.99	28.05	26.16	28.18	28.29	**28.34**
Barbara	28.59	28.70	28.75	28.62	28.68	28.60	28.59	28.70	**28.77**	26.77	26.71	**26.86**	26.74	26.82	26.82	26.65	26.47	26.67
AVE.	32.718	33.524	33.397	33.343	33.624	33.359	33.553	33.725	**33.782**	29.384	29.846	29.959	29.895	29.901	30.011	29.163	30.327	**30.405**

SR [29], LLE [3], the Zeyde′s method [33], anchored neighborhood regression method (ANR) [20], the Beta process joint dictionary learning method (BPJDL) [12], deformable patch super resolution method (DPSR) [35], adjusted ANR (A+) [21] and convolutional neural network based method CNN-SR [6]. All methods follow the experimental setting of [29], in which the LR images are resized from ground truth HR images by bi-cubic interpolation. We download the source codes from the author′s websites, and use the recommended parameters by the authors.

We perform the SR comparison on 15 widely used test images. The PSNR values by the competing methods are shown in Table 2. CSC-SR achieves better results than patch-based joint dictionary learning methods on most of testing images. Compared with the state-of-the-art CNN-SR and A+ method, the proposed CSC-SR methods also achieves higher PSNR index on more testing images. Overall, CSC-SR improves the average PSNR value of CNN-SR with more than 0.2 dB and improves the average PSNR value of A+ with about 0.1 dB.

Let′s then compare the visual quality of the SR results. In Figs. 4, 5 and 6, we show the SR results of images *Foreman*, *Barbara* and *Face*, by the competing algorithms. As highlighted in the small window, the SR results by competing algorithms either have clear ringing artifacts in strong edge area or over-smooth too much the edge, while the edges produced by the CSC-SR method are more natural. In summary, the results generated by the proposed CSC-SR method have more textures and less artifacts, producing visually more pleasant SR outputs. More examples of SR results can be found in the supplementary file.

4.5. Time Complexity

Table 3. Running time of different SR algorithms

Methods	**ScSR**	**Zeyde**	**ANR**	**BPJDL**	**DPSR**
Time(s)	259.14	6.17	1.95	528.62	52429.88
Methods	**CNN**	**A+**	**CSC**$\times 2$	**CSC**$\times 3$	**CSC**$\times 4$
Time(s)	7.65	2.02	570.93	276.75	170.85

The main computational burden of our CSC-SR model in the testing phase is solving the CSC problem. As analyzed in [23], the time complexity of CSC in each iteration is dominated by the FFTs, which is $\mathcal{O}(KNlogN)$ for an image with N pixels decomposed by K filters. Suppose that the LR image is of size 256×256, or 170×170, or 128×128, and we are going to enlarge it to 512×512, i.e., the zooming factor is 2, 3 and 4, respectively. Since CSC-SR directly decomposes the LR image for SR, its running time varies a lot for different zooming factors. In contrast, all the competing SR algorithms firstly enlarge the LR image to 512×512 by using the bicubic interpolator, and then post-process the enlarged image. Therefore, their running time is basically the same for the three zooming factors.

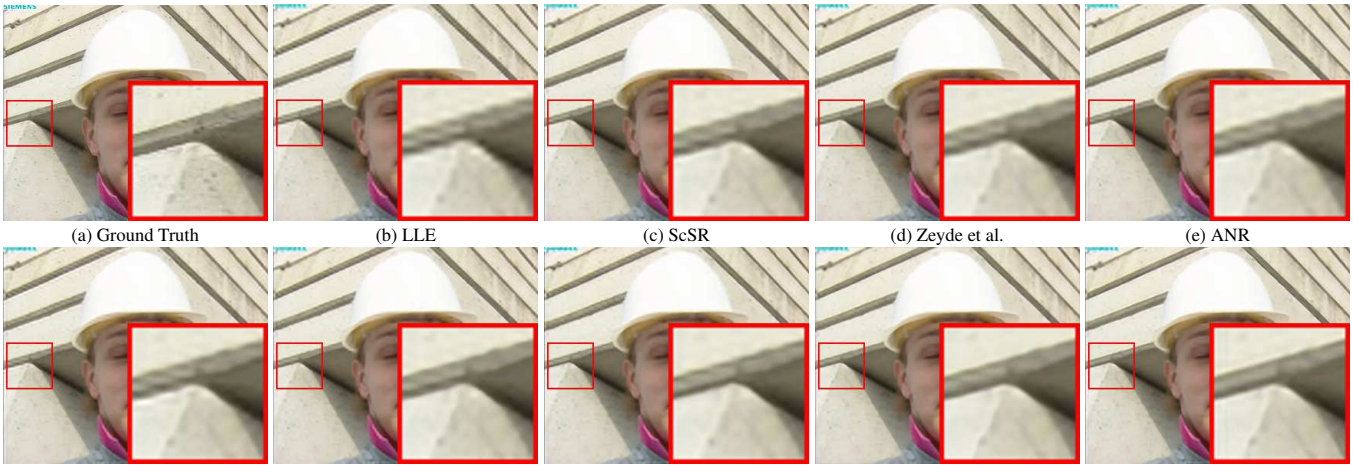
Table 3 lists the running time by different SR algorithms. The experiments are conducted on a PC with i7 3.5GHz CPU and 32Gb RAM. The Zeyde′s method [33], ANR [20], A+ [21] and CNN-SR [6] cost less than 10 seconds. The running time of the proposed CSC-SR is comparable to ScSR [29] and BPJDL [12]. DPSR [35] is the slowest method.

5. Conclusion

In this paper, we proposed a convolutional sparse coding based super resolution (CSC-SR) method. CSC directly decomposes the whole image by filtering, which naturally takes the consistency of pixels in overlapped patches into consideration. We introduced a mapping function between the LR and HR sparse coding feature maps for SR. Different from previous patch based sparse coding methods, the convolutional decomposition mechanism of CSC can keep the spatial information of input signal in the feature maps, and exploit the consistency of neighboring patches for better image reconstruction. Compared with other state-of-the-art SR methods, our algorithm achieves not only very competitive PSNR index, but also more pleasant visual quality of image texture and edge structures.

6. Acknowledgement

We gratefully acknowledge the support from NVIDIA Corporation for providing us the Tesla K40 GPU used in this research. This research is supported by the HK RGC GRF grant (PolyU5313/13E) and National Science Foundation of China (NSFC: 61271093).



(a) Ground Truth (b) LLE (c) ScSR (d) Zeyde et al. (e) ANR

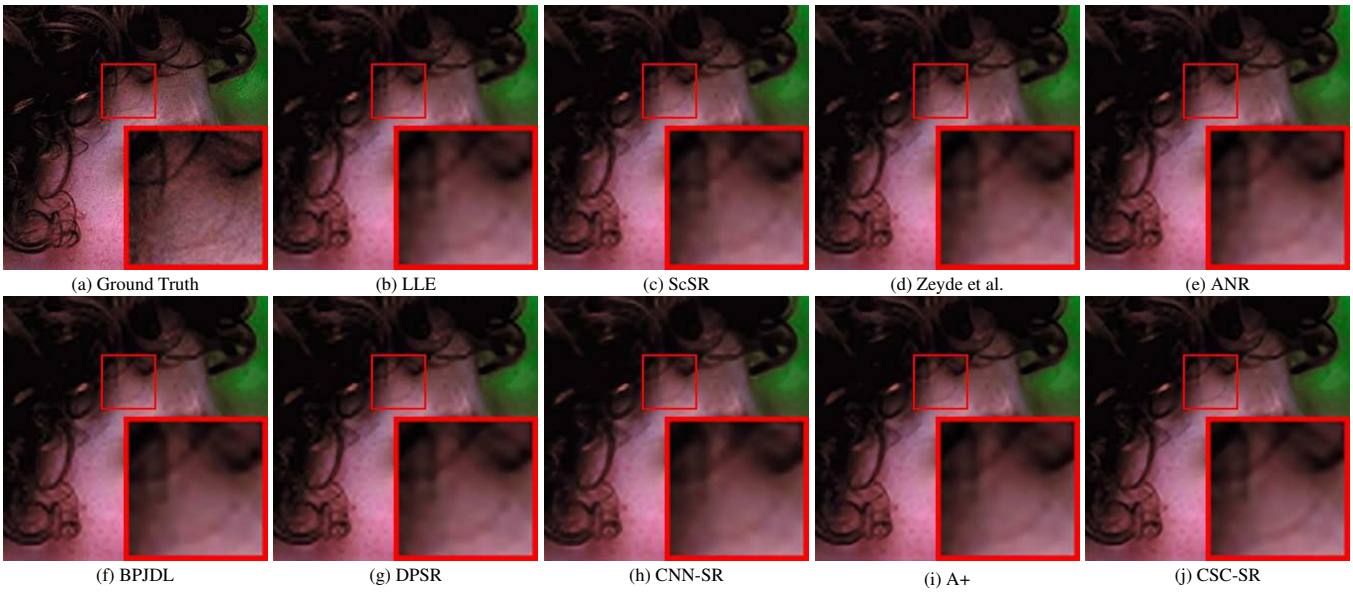
(f) BPJDL (g) DPSR (h) CNN-SR (i) A+ (j) CSC-SR

Figure 4. Super resolution results on image *Foreman* by different algorithms (zooming factor 3).

(a) Ground Truth (b) LLE (c) ScSR (d) Zeyde et al. (e) ANR

(f) BPJDL (g) DPSR (h) CNN-SR (i) A+ (j) CSC-SR

Figure 5. Super resolution results on image *Barbara* by different algorithms (zooming factor 3).



(a) Ground Truth (b) LLE (c) ScSR (d) Zeyde et al. (e) ANR

(f) BPJDL (g) DPSR (h) CNN-SR (i) A+ (j) CSC-SR

Figure 6. Super resolution results on image *Face* by different algorithms (zooming factor 4).

References

[1] H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding. In *CVPR*, 2013. 3

[2] H. Bristow and S. Lucey. Optimization methods for convolutional sparse coding. *arXiv preprint arXiv:1406.2407*, 2014. 2

[3] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004. 1, 4, 7

[4] Y. Chen, R. Ranftl, and T. Pock. Insights into analysis operator learning: A view from higher-order filter-based mrf model. *IEEE Trans. on Image Processing*, 2014. 1

[5] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *ECCV*. 2014. 1

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*. 2014. 1, 7

[7] R. Fattal. Image upsampling via imposed edge statistics. In *ACM Transactions on Graphics (TOG)*, volume 26, page 95. ACM, 2007. 1

[8] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000. 1, 2, 4

[9] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 1

[10] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007. 6

[11] S. Gu, N. Sang, and F. Ma. Fast image super resolution via local regression. In *ICPR*, 2012. 1

[12] L. He, H. Qi, and R. Zaretzki. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. In *CVPR*, 2013. 1, 3, 5, 7

[13] C. Kervrann. Pewa: Patch-based exponentially weighted aggregation for image denoising. In *NIPS*, 2014. 2

[14] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012. 4

[15] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. on Image Processing,*, 17(1):53–69, 2008. 2

[16] C. Osendorfer, H. Soyer, and P. van der Smagt. Image super-resolution with fast approximate convolutional sparse coding. In *Neural Information Processing*, pages 250–257. Springer, 2014. 2

[17] T. Peleg and M. Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE trans. on image processing*, 23(6):2569–2582, 2014. 3

[18] J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *CVPR*, 2008. 1

[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. 1

[20] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, 2013. 7

[21] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision–ACCV 2014*, pages 111–126. Springer, 2014. 7

[22] S. Wang, D. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *CVPR*, 2012. 1, 3

[23] B. Wohlberg. Efficient convolutional sparse coding. In *ICASSP*, 2014. 3, 4, 7

[24] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 2

[25] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-image super-resolution: A benchmark. In *ECCV*. 2014. 2

[26] C.-Y. Yang and M.-H. Yang. Fast direct super-resolution by simple functions. In *ICCV*, 2013. 1

[27] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *CVPR*, 2013. 1

[28] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang. Bilevel sparse coding for coupled feature spaces. In *CVPR*, 2012. 4

[29] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008. 1, 2, 3, 4, 5, 6, 7

[30] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 2

[31] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010. 2, 3

[32] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011. 2

[33] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2012. 4, 7

[34] L. W. Zhong and J. T. Kwok. Fast stochastic alternating direction method of multipliers. In *ICML*, 2013. 4

[35] Y. Zhu, Y. Zhang, and A. L. Yuille. Single image super-resolution using deformable patches. In *CVPR*, 2012. 3, 5, 7

[36] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 2