Couple Dictionary Training for Image Super-resolution

Jianchao Yang, *Student Member, IEEE,* Zhaowen Wang, *Student Member, IEEE,* Zhe Lin, *Member, IEEE,* Scott Cohen, *Member, IEEE*, and Thomas Huang, *Life Fellow, IEEE*

Abstract

In this paper, we propose a novel coupled dictionary training method for single image super-resolution based on patch-wise sparse recovery, where the learned couple dictionaries relate the low- and high-resolution image patch spaces via sparse representation. The learning process enforces that the sparse representation of a low-resolution image patch in terms of the low-resolution dictionary can well reconstruct its underlying high-resolution image patch with the dictionary in the high-resolution image patch space. We model the learning problem as a bilevel optimization problem, where the optimization includes an ℓ^1-norm minimization problem in its constraints. Implicit differentiation is employed to calculate the desired gradient for stochastic gradient descent. We demonstrate that our coupled dictionary learning method can outperform the existing joint dictionary training method both quantitatively and qualitatively. Furthermore, for real applications, we speed up the algorithm approximately 10 times by learning a neural network model for fast sparse inference and selectively processing only those visually salient regions. Extensive experimental comparisons with state-of-the-art super-resolution algorithms validate the effectiveness of our proposed approach.

Jianchao Yang, Zhaowen Wang, and Thomas Huang are with Beckman Institute, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Illinois, IL 61801, USA (e-mail: jyang29@ifp.illinois.edu; wang308@illinois.edu; huang@ifp.illinois.edu).

Zhe Lin and Scott Cohen are with Adobe Systems Inc., San Jose, CA 95110, USA (e-mail: zlin@adobe.com; scohen@adobe.com)

I. INTRODUCTION

A. Dictionary Learning for Sparse Modeling

Signal processing and pattern recognition techniques commonly require meaningful data representations that capture the useful properties of the signal, *e.g.*, for compression, the representation should account for the essential content of the signal with only a few coefficients. Representations with orthogonal and biorthogonal dictionaries were prevalent for years in signal processing techniques due to their mathematical simplicity and computational efficiency, *e.g.*, wavelets for compression (JPEG2000) and denoising [1]. Despite their simplicity, these dictionaries are limited in their expressive power, leading to the recent development of over-complete dictionaries, which have more elementary signal atoms than the signal dimension and thus offer the flexibility to represent a much wider range of signal phenomena [2].

Sparse and redundant data modeling seeks the representation of signals as linear combinations of a small number of atoms from a pre-specific dictionary. Recently, there is a fast increasing interest in dictionary training — using machine learning techniques to obtain an over-complete dictionary adapted to the training data. Most of these algorithms employ ℓ^0- or ℓ^1-sparsity penalty measures, which give simple formulations and allow the use of recently developed efficient sparse coding techniques. Examples include the Method of Optimal Directions (MOD) with ℓ^0-sparsity measure proposed by Engan *et al.* [3], the K-SVD algorithm by Aharon *et al.* [4], an formulation with ℓ^1 sparsity measure by Lee *et al.*in [5], and an online large-scale learning algorithm by Mairal *et al.* [6]. The main advantage of the trained dictionaries is that they are adaptive to the signals of interest, which contributes to the state-of-the-art performance on many signal recovery tasks, *e.g.*, denoising [4], inpainting [7] and super-resolution [8], [9].

Current dictionary learning methods mainly focus on training an over-complete dictionary in a single feature space for various recovery or recognition tasks. In many applications and scenarios, we have coupled sparse feature spaces: high- and low-resolution signal (feature) spaces in patch-based super-resolution; source and target image patch spaces in texture transfer, *etc*. We denote the two spaces as the observation space and the latent space, which are tied by some mapping function (not necessarily linear and could be unknown). It is desirable to learn two coupled dictionaries, namely, the observation dictionary and the latent dictionary collaboratively such that the sparse representation of the signal in the observation space can be used to well reconstruct its paired signal in the latent feature space. Learning such coupled dictionaries have many potential applications in both signal processing and computer vision,

e.g., compressive sensing [10]. However, this problem has been little addressed in the literature. Yang *et al.* [8] proposed a joint dictionary training method to learn the dictionaries for high- and low-resolution image patch spaces. The method essentially concatenates the two feature spaces and converts the problem to the standard sparse coding in a single feature space. As such, the resulting dictionaries are not indeed trained for each of the feature spaces individually, and accurate recovery is not guaranteed. Yang *et al.* [11] proposed to train the dictionary together with the feature representation and prediction model for image classification in a bilevel formulation. With justified mathematic proof for the optimization, Mairal *et al.* [12] generalize the similar idea to a more general regression framework to produce dictionaries better suited to the specific tasks. However, they do not explicitly study the sparse modeling problem across different feature spaces.

In this paper, we propose a new dictionary learning method which explicitly enforces that the sparse representation of an observation signal in terms of the observation dictionary can well represent its underlying signal with the latent dictionary. The optimization employs a stochastic gradient descent procedure, where the gradient is computed via back-propagation and implicit differentiation. We then apply our new dictionary training method to patch-based single image super-resolution and demonstrate notable improvements over the previous approaches. As far as we know, this is the first work in the SR literature to optimize the dictionaries directly targeted at minimizing the recovery errors.

B. Image Super-resolution

Image super-resolution (SR) are techniques aiming estimation of a high-resolution (HR) image from one or several low-resolution (LR) observation images, which offer the promise of overcoming some of the inherent resolution limitations of low-cost imaging sensors (*e.g.*, cell phone cameras or surveillance cameras), and allow better utilization of the growing capability of HR displays (*e.g.*, HD LCDs). Conventional super-resolution approaches normally require multiple LR inputs of the same scene with sub-pixel motions. The SR task is thus cast as an inverse problem of recovering the original HR image by fusing the LR inputs, based on reasonable assumptions or prior knowledge about the observation model. However, SR image reconstruction is typically severely ill-conditioned because of the insufficient number of observations and the unknown registration parameters. Various regularization techniques are therefore proposed to stabilize the inversion of this ill-posed problem [13]–[15].

However, the performance of conventional approaches is only acceptable for small upscaling factors (usually less than 2) [16], leading to the development of later example-based learning approaches, which aim to learn the co-occurrence prior between the HR and LR image local structures from an external

training database [17]–[19]. This training database is usually required to contain millions of HR/LR image patch pairs in order to represent a generic image well, which makes the algorithms computationally intensive. Instead of relying on an external database, several recently proposed approaches exploit the self-similarity properties of local image patches within and across different spatial scales in the same image for super-resolution [20]–[24]. These approaches either need a separate deblurring process [20], [23], which is ill-posed and requires parameter tuning by itself, or relies too much on local image singularities, *e.g.*, edges and corners, thus generating super-resolution results which are not photo realistic [22], [24]. Motivated by the recent compressive sensing theories [10], Yang *et al.*in [8], [25] proposed to use sparse representation to recover the HR image patch. The method can generate both photo realistic textures and sharp edges from a single input image. However, the joint dictionary training method proposed in [8] does not guarantee that the sparse representation of a LR image patch can well reconstruct its underlying HR image patch. We will show that our coupled dictionary learning method can overcome this problem and demonstrate superior performance both qualitatively and quantitatively.

Despite its strength in sparse signal recovery, sparse representation cannot be calculated in an efficient way due to the ℓ^1-norm minimization, which hinders its application in many real time scenarios with constrained computational resources. For example, it typically takes more than one minute to magnify a LR image of size 128×128 by a factor of 2, which is intolerable to most users. In this paper, to enable the practical application of our sparse recovery based super-resolution in consumer photo editing, *e.g.*, in Photoshop^{TM}, we further propose an efficient implementation of our algorithm based on two strategies:

1) **Selective patch processing**. For image super-resolution or upscaling, image regions with textures, sharp edges and corners are more crucial to visual quality improvement. Therefore, we apply our high-accuracy sparse recovery method selectively on those salient regions, and process other less notable regions using more efficient methods, *e.g.*, bicubic interpolation, without compromise to overall visual quality.
2) **Learning a neural network model for fast sparse inference**. The bottle neck of our algorithm is the computation of the sparse code from the ℓ^1-norm minimization for each LR input image patch. Instead of solving the exact optimization, we train a feed-forward model for fast approximate inference [26].

The remainder of this paper is organized as follows. Section II reviews two related dictionary training methods for sparse representation. Section III presents our dictionary learning method for coupled feature spaces. In Section IV, we discuss how to apply our dictionary learning method to the single image super-

resolution. Section V proposes a fast and approximate version of our coupled dictionary learning based method for practical applications. Then in Section VI, we demonstrate the effectiveness of our approach by comparing it with state-of-the-art image super-resolution techniques. Finally, Section VII concludes our paper with discussions and future works.

II. Current Dictionary Learning Methods

In this section, we introduce two related dictionary learning methods with the ℓ^1-sparsity measure—sparse coding in a single feature space and joint sparse coding in coupled feature spaces for signal recovery.

A. Sparse Coding

The goal of sparse coding is to represent an input signal $\boldsymbol{x} \in \mathbb{R}^d$ approximately as a weighted linear combination of a few elementary signals called basis atoms, often chosen from an over-complete dictionary $\boldsymbol{D} \in \mathbb{R}^{d\times K}$ ($d < K$). Sparse coding is the method to automatically discover such a good set of basis atoms. Concretely, given the training data $\{\boldsymbol{x}_i\}_{i=1}^N$, the problem of learning a dictionary for sparse coding, in its most popular form, is solved by minimizing the energy function that combines squared reconstruction errors and the ℓ^1-sparsity penalties on the representations:

$$\begin{aligned} &\min_{\boldsymbol{D},\{\boldsymbol{\alpha}_i\}_{i=1}^N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \\ &\text{s.t.} \quad \|D(:,k)\|_2 \le 1, \quad \forall k \in \{1,2,...,K\}, \end{aligned} \tag{1}$$

where $D(:,k)$ is the k-th column of $\boldsymbol{D}$, $\boldsymbol{\alpha}_i$ is the sparse code of $\boldsymbol{x}_i$, and λ is a parameter controlling the sparsity penalty and representation fidelity. The above optimization problem is convex in either $\boldsymbol{D}$ or $\{\boldsymbol{\alpha}_i\}_{i=1}^N$ when the other is fixed, but not in both. When $\boldsymbol{D}$ is fixed, inference for $\{\boldsymbol{\alpha}_i\}_{i=1}^N$ is known as the Lasso problem in the statistic literature; when $\{\boldsymbol{\alpha}_i\}_{i=1}^N$ are fixed, solving $\boldsymbol{D}$ becomes a standard quadratically constrained quadratic programming (QCQP) problem. A practical solution to Eqn. (1) is to alternatively optimize over $\boldsymbol{D}$ and $\{\boldsymbol{\alpha}_i\}_{i=1}^N$, and the algorithm is guaranteed to converge to a local minimum [5].

B. Joint Sparse Coding

Unlike the standard sparse coding, joint sparse coding considers the problem of learning two dictionaries $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ for two coupled feature spaces, $\mathcal{X}$ and $\mathcal{Y}$, tied by a certain mapping function $\mathcal{F}$, such that the sparse representation of $\boldsymbol{x}_i \in \mathcal{X}$ in terms of $\boldsymbol{D}_x$ should be the same as that of $\boldsymbol{y}_i \in \mathcal{Y}$ in terms

of $\boldsymbol{D}_y$, where $\boldsymbol{y}_i = \mathcal{F}(\boldsymbol{x}_i)$. Accordingly, if $\boldsymbol{y}_i$ is our observation signal, we can recover its underlying latent signal $\boldsymbol{x}_i$ via its sparse representation in terms of $\boldsymbol{D}_y$. Yang *et al.* [8] addressed this problem by generalizing the basic sparse coding scheme as follows:

$$\begin{aligned} \min_{\boldsymbol{D}_x, \boldsymbol{D}_y, \{\boldsymbol{\alpha}_i^{x\setminus y}\}_{i=1}^N} & \sum_{i=1}^{N} \left\{ \frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{D}_x\boldsymbol{\alpha}_i^x\|_2^2 + \lambda\|\boldsymbol{\alpha}_i^x\|_1 \right\} + \left\{ \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{D}_y\boldsymbol{\alpha}_i^y\|_2^2 + \lambda\|\boldsymbol{\alpha}_i^y\|_1 \right\}, \\ \text{s.t.} \quad & \|D_x(:,k)\|_2 \leq 1, \quad \|D_y(:,k)\|_2 \leq 1, \quad \boldsymbol{\alpha}_i^x = \boldsymbol{\alpha}_i^y, \end{aligned} \tag{2}$$

which is equivalent to

$$\begin{aligned} \min_{\boldsymbol{D}_x, \boldsymbol{D}_y, \{\boldsymbol{\alpha}_i\}_{i=1}^N} & \sum_{i=1}^{N} \left(\|\boldsymbol{x}_i - \boldsymbol{D}_x\boldsymbol{\alpha}_i\|_2^2 + \|\boldsymbol{y}_i - \boldsymbol{D}_y\boldsymbol{\alpha}_i\|_2^2 \right) + \lambda\|\boldsymbol{\alpha}_i\|_1, \\ \text{s.t.} \quad & \|D_x(:,k)\|_2 \leq 1, \quad \|D_y(:,k)\|_2 \leq 1, \end{aligned} \tag{3}$$

The formulation above basically requires that the resulting common sparse representation $\boldsymbol{\alpha}_i$ should reconstruct both $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ well. Grouping the two reconstruction error terms together and denoting

$$\bar{\boldsymbol{x}}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}, \quad \bar{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{D}_x \\ \boldsymbol{D}_y \end{bmatrix}, \tag{4}$$

we can convert Eqn. (3) to the standard sparse coding problem in the concatenated feature space of $\mathcal{X}$ and $\mathcal{Y}$:

$$\begin{aligned} \min_{\bar{\boldsymbol{D}}, \{\boldsymbol{\alpha}_i\}_{i=1}^N} & \sum_{i=1}^{N} \|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{D}}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \\ \text{s.t.} \quad & \|\bar{\boldsymbol{D}}(:,k)\|_2 \leq 1. \end{aligned} \tag{5}$$

Therefore, such a joint sparse coding scheme can only be claimed to be optimal in the concatenated feature space of $\mathcal{X}$ and $\mathcal{Y}$, but not in each feature space individually.

In the testing phase, given an observed signal $\boldsymbol{y}$, we want to recover the corresponding latent signal $\boldsymbol{x}$ by inferring its sparse representation. Since $\boldsymbol{x}$ is unknown, there is no way to enforce the equivalence constraint on the sparse representations of $\boldsymbol{y}$ and $\boldsymbol{x}$, as has been done in the training phase. Instead, we can only infer the sparse representation of $\boldsymbol{y}$ in the feature space $\mathcal{Y}$ with respect to $\boldsymbol{D}_y$, and use it as an approximation to the joint sparse representation of $\boldsymbol{x}$ and $\boldsymbol{y}$, which is not guaranteed to be consistent with the sparse representation of $\boldsymbol{x}$ in terms of $\boldsymbol{D}_x$. Consequently, accurate recovery is not assured using the above jointly learned dictionaries.

III. Coupled Dictionary Learning for Sparse Recovery

A. Problem Statement

Suppose we have two coupled feature spaces: the latent space $\mathcal{X} \subseteq \mathbb{R}^{d_1}$ and the observation space $\mathcal{Y} \subseteq \mathbb{R}^{d_2}$, where the signals are sparse, *i.e.*, the signals have sparse representations in terms of certain dictionaries. Signals in $\mathcal{Y}$ are observable, and signals in $\mathcal{X}$ are what we want to recover or infer. There exists some mapping function $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ (not necessarily linear and probably unknown) that maps a signal $\boldsymbol{x}$ in $\mathcal{X}$ to its corresponding signal $\boldsymbol{y}$ in $\mathcal{Y}$: $\boldsymbol{y} = \mathcal{F}(\boldsymbol{x})$. We assume that the mapping function is nearly injective; otherwise, the inference for $\mathcal{X}$ from $\mathcal{Y}$ would be impossible. Our problem is to find a coupled dictionary pair $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ for space $\mathcal{X}$ and $\mathcal{Y}$ respectively, such that given any signal $\boldsymbol{y} \in \mathcal{Y}$, we can use its sparse representation in terms of $\boldsymbol{D}_y$ to recover the corresponding latent signal $\boldsymbol{x} \in \mathcal{X}$ in terms of $\boldsymbol{D}_x$. Formally, an ideal pair of coupled dictionaries $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ should satisfy the following equations for any coupled signal pair $\{\boldsymbol{y}_i, \boldsymbol{x}_i\}$:

$$\boldsymbol{z}_i = \arg\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{y}_i - \boldsymbol{D}_y\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1, \forall i = 1...N \tag{6}$$

$$\boldsymbol{z}_i = \arg\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{x}_i - \boldsymbol{D}_x\boldsymbol{\alpha}_i\|_2^2 \qquad , \forall i = 1...N^1 \tag{7}$$

where $\{\boldsymbol{x}_i\}_{i=1}^N$ are the training samples from $\mathcal{X}$, $\{\boldsymbol{y}_i\}_{i=1}^N$ are the training samples from $\mathcal{Y}$ with $\boldsymbol{y}_i = \mathcal{F}(\boldsymbol{x}_i)$, and $\{\boldsymbol{z}_i\}_{i=1}^N$ are the sparse representations.

Signal recovery from coupled spaces can be thought as a problem similar to compressive sensing [10]. In the context of compressive sensing, the observation and latent spaces are related through a linear random projection function $\mathcal{F}$. Dictionary $\boldsymbol{D}_x$ is usually chosen to be a mathematically defined basis (*e.g.*, wavelets), and $\boldsymbol{D}_y$ is obtained directly from $\boldsymbol{D}_x$ with the linear mapping $\mathcal{F}$. Under some moderate conditions, the sparse representation of $\boldsymbol{y}$ derived from Eqn. (6) can be used to recover $\boldsymbol{x}$ with performance guarantees. However, in more general scenarios where the mapping function $\mathcal{F}$ is unknown and may take non-linear forms,[2] the compressive sensing theory cannot be applied. Then it becomes more favorable to learn the coupled dictionaries from the training data with machine learning techniques.

[1]Alternatively, one can require that the sparse representation of $\boldsymbol{x}_i$ in terms of $\boldsymbol{D}_x$ is $\boldsymbol{z}_i$. However, since only the recovery accuracy of $\boldsymbol{x}_i$ is concerned, we directly impose $\boldsymbol{x}_i \approx \boldsymbol{D}_x\boldsymbol{z}_i$.

[2]In the example of patch-based super-resolution, the image degradation process $\mathcal{F}$ from HR space to LR space is no longer a simple linear transformation of blurring and downsampling if the signals in the LR space are represented as high frequency features of raw patches, which is typically employed for better visual effect. We will discuss this in more details in Section IV.

B. Formulation

Given an input signal $\boldsymbol{y}$, the recovery of its latent signal $\boldsymbol{x}$ consists of two consecutive steps: first find the sparse representation $\boldsymbol{z}$ of $\boldsymbol{y}$ in terms of $\boldsymbol{D}_y$ according to Eqn. (6), and then estimate the latent signal as $\boldsymbol{x} = \boldsymbol{D}_x\boldsymbol{z}$. Since the goal of our dictionary learning is to minimize the recovery error of $\boldsymbol{x}$, we define the following squared loss term:

$$L(\boldsymbol{D}_x, \boldsymbol{D}_y, \boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{D}_x\boldsymbol{z} - \boldsymbol{x}\|_2^2. \tag{8}$$

Then the optimal dictionary pair $\{\boldsymbol{D}_x^*, \boldsymbol{D}_y^*\}$ is found by minimizing the empirical expectation of Eqn. 8 over the training signal pairs,

$$\begin{aligned} &\min_{\boldsymbol{D}_x, \boldsymbol{D}_y} \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{D}_x, \boldsymbol{D}_y, \boldsymbol{x}_i, \boldsymbol{y}_i) \\ \text{s.t.} \quad &\boldsymbol{z}_i = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y}_i - \boldsymbol{D}_y\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \text{ for } i = 1, 2, ..., N, \\ &\|D_x(:,k)\|_2 \le 1, \quad \|D_y(:,k)\|_2 \le 1, \text{ for } k = 1, 2, ..., K. \end{aligned} \tag{9}$$

Simply minimizing the above empirical loss does not guarantee that $\boldsymbol{y}$ can be well represented by $\boldsymbol{D}_y$. Therefore, we can add one more reconstruction term to the loss function to ensure good representation of $\boldsymbol{y}$,

$$L(\boldsymbol{D}_x, \boldsymbol{D}_y, \boldsymbol{x}_i, \boldsymbol{y}_i) = \frac{1}{2}\left(\gamma\|\boldsymbol{D}_x\boldsymbol{z}_i - \boldsymbol{x}_i\|_2^2 + (1-\gamma)\|\boldsymbol{y}_i - \boldsymbol{D}_y\boldsymbol{z}_i\|_2^2\right), \tag{10}$$

where γ ($0 < \gamma \le 1$) balances the two reconstruction errors.

The objective function in Eqn. (9) is highly nonlinear and highly nonconvex. We propose to minimize it by alternatively optimizing over $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ while keeping the other fixed. When $\boldsymbol{D}_y$ is fixed, the sparse representation $\boldsymbol{z}_i$ can be determined for each $\boldsymbol{y}_i$ with $\boldsymbol{D}_y$, and the problem of Eqn. (9) reduces to

$$\begin{aligned} &\min_{\boldsymbol{D}_x} \sum_{i=1}^{N} \frac{1}{2}\|\boldsymbol{D}_x\boldsymbol{z}_i - \boldsymbol{x}_i\|_2^2 \\ \text{s.t.} \quad &\boldsymbol{z}_i = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y}_i - \boldsymbol{D}_y\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \text{ for } i = 1, 2, ..., N, \\ &\|\boldsymbol{D}_x(:,k)\|_2 \le 1, \text{ for } k = 1, 2, ..., K. \end{aligned} \tag{11}$$

which is a quadratically constrained quadratic programing that can be solved efficiently using conjugate gradient descent [5]. When $\boldsymbol{D}_x$ is fixed, the optimization over $\boldsymbol{D}_y$ is more complicated and is discussed in the following.

C. Optimization for Dictionary $\boldsymbol{D}_y$

Minimizing the loss function of Eqn. (9) over $\boldsymbol{D}_y$ is a highly nonconvex bilevel programming problem [27]. The upper-level optimization of Eqn. (9) depends on the variable $\boldsymbol{z}_i$, which is the optimum of a lower-level ℓ^1-minimization. To solve this bilevel problem, we employ the same descent method developed in our previous work [11]. For the descent method, we need to find a descent direction along which a feasible step in this direction will decrease the objective function value. For easy of presentation, we drop the subscripts of $\boldsymbol{x}_i$, $\boldsymbol{y}_i$, and $\boldsymbol{z}_i$ in the following. Applying the chain rule, we have

$$\frac{\partial L}{\partial \boldsymbol{D}_y} = \frac{1}{2}\left(\sum_{j\in\Omega}\frac{\partial(\gamma R_x + (1-\gamma)R_y)}{\partial z_j}\frac{dz_j}{d\boldsymbol{D}_y} + (1-\gamma)\frac{\partial R_y}{\partial \boldsymbol{D}_y}.\right), \tag{12}$$

Here, we denote $R_x = \|\boldsymbol{D}_x\boldsymbol{z} - \boldsymbol{x}\|_2^2$ and $R_y = \|D_y\boldsymbol{z} - \boldsymbol{y}\|_2^2$ as the reconstruction residuals with representation $\boldsymbol{z}$ for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. z_j is the j_{th} element of $\boldsymbol{z}$, and Ω denotes the index set for j, where the derivative $dz_j/d\boldsymbol{D}_y$ is well defined. Let $\tilde{\boldsymbol{z}}$ denote the vector built with the elements $\{z_j\}_{j\in\Omega}$, and $\tilde{\boldsymbol{D}_x}$ and $\tilde{\boldsymbol{D}_y}$ denote the dictionaries that consist of the columns in $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ with indices in Ω. It is easy to find that

$$\begin{aligned}
\frac{\partial R_x}{\partial \tilde{\boldsymbol{z}}} &= 2\tilde{\boldsymbol{D}}_x^T(\boldsymbol{D}_x\boldsymbol{z} - \boldsymbol{x}),\\
\frac{\partial R_y}{\partial \tilde{\boldsymbol{z}}} &= 2\tilde{\boldsymbol{D}}_y^T(\boldsymbol{D}_y\boldsymbol{z} - \boldsymbol{y}),\\
\frac{\partial R_y}{\partial \boldsymbol{D}_y} &= 2(\boldsymbol{D}_y\boldsymbol{z} - \boldsymbol{y})\boldsymbol{z}^T.
\end{aligned} \tag{13}$$

To evaluate the gradient in Eqn. (12), we still need to find the index set Ω and the derivative $d\,\tilde{\boldsymbol{z}}/d\boldsymbol{D}_y$. However, there is no analytical link between $\tilde{\boldsymbol{z}}$ and $\boldsymbol{D}_y$. We use the technique developed in [11] to find the derivative in the following, which turns out to work well in practice.

1) Sparse Derivative: For the Lasso problem in Eqn. (6), we have the following condition for the optimum $\boldsymbol{z}$ [28]:

$$\frac{\partial\|\boldsymbol{y} - \boldsymbol{D}_y\boldsymbol{z}\|_2^2}{\partial z_j} + \lambda\ \mathrm{sign}(z_j) = 0,\ \text{for } j \in \Lambda, \tag{14}$$

where $\Lambda = \{j|z_j \neq 0\}$. Define our index set $\Omega = \{j||z_j| > 0^+\}$, we have

$$\frac{\partial\|\boldsymbol{y} - \tilde{\boldsymbol{D}}_y\tilde{\boldsymbol{z}}\|_2^2}{\partial z_j} + \lambda\ \mathrm{sign}(z_j) = 0,\ \text{for } j \in \Omega. \tag{15}$$

Equivalently, we have

$$\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y\tilde{\boldsymbol{z}} - \tilde{\boldsymbol{D}}_y{}^T\boldsymbol{y} + \lambda\mathrm{sign}(\tilde{\boldsymbol{z}}) = 0. \tag{16}$$

It is easy to show that $\tilde{\boldsymbol{z}}$ is a continuous function of $\boldsymbol{D}_y$ [29]. Therefore, a small perturbation on $\boldsymbol{D}_y$ will not change the signs of the elements in $\tilde{\boldsymbol{z}}$. As a result, can apply the implicit differentiation [30] on

Eqn. (16) to obtain

$$\frac{\partial\{\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y\tilde{\boldsymbol{z}}-\tilde{\boldsymbol{D}}_y{}^T\boldsymbol{y}\}}{\partial\tilde{\boldsymbol{D}}_y}=\frac{\partial\{-\lambda\cdot\text{sign}(\tilde{\boldsymbol{z}})\}}{\partial\tilde{\boldsymbol{D}}_y}$$

$$\Rightarrow\quad\frac{\partial\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y}{\partial\tilde{\boldsymbol{D}}_y}\tilde{\boldsymbol{z}}+\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y\frac{\partial\tilde{\boldsymbol{z}}}{\partial\tilde{\boldsymbol{D}}_y}-\frac{\partial\tilde{\boldsymbol{D}}_y{}^T\boldsymbol{y}}{\partial\tilde{\boldsymbol{D}}_y}=0. \tag{17}$$

Then, we calculate the derivative as

$$\frac{\partial\tilde{\boldsymbol{z}}}{\partial\tilde{\boldsymbol{D}}_y}=\left(\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y\right)^{-1}\left(\frac{\partial\tilde{\boldsymbol{D}}_y{}^T\boldsymbol{y}}{\partial\tilde{\boldsymbol{D}}_y}-\frac{\partial\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y}{\partial\tilde{\boldsymbol{D}}_y}\tilde{\boldsymbol{z}}\right), \tag{18}$$

where we assume the solution to Eqn. (6) is unique and $\left(\tilde{\boldsymbol{D}}_y{}^T\tilde{\boldsymbol{D}}_y\right)^{-1}$ exists. Eqn. (18) only gives us the derivative function of $\tilde{\boldsymbol{z}}$ with respect to $\tilde{\boldsymbol{D}}_y$, which builds only on the index set Ω. To evaluate Eqn. (12), we can set the remaining gradient elements of $\partial\boldsymbol{z}/\partial\boldsymbol{D}_y$ to be zero. From a practical point of view, as long as the approximate derivative given by Eqn. (12) is a feasible descent direction for the optimization, the descent method guarantees that the objective function value will always decrease for a feasible step along that direction. Empirically, the above Eqn. (12) indeed serves as a descent direction for the optimization, as shown in our previous work [11] and the experiments in this work. Theoretically, it is easy to establish a much stronger argument for Eqn (12) based on Eqn. (18). It can be shown that for $\Omega=\Lambda$, this set and the corresponding solution signs will not change for a small perturbation of $\boldsymbol{D}_y$, as long as λ is not a transition point of $\boldsymbol{y}$ in terms of $\boldsymbol{D}_y$ [31]. Because the chance of λ being a transition point of $\boldsymbol{y}$ is low for a reasonable distribution assumption on $\boldsymbol{y}$, Eqn. (12) will approximate the true gradient accurately on expectation. Instead of looking into the mathematics in this work, we focus on our image super-resolution application. And we refer the reader to a recent work by Mairal *et al.* [12] for a mathematic analysis from a slightly different perspective.

D. Algorithm Summarization

With the gradient in Eqn. (12) calculated, we employ a projected stochastic gradient descent procedure for the optimization of $\boldsymbol{D}_y$ due to its fast convergence and good behavior in practice. Because of the high nonconvexity of the Bilevel optimization over $\boldsymbol{D}_y$ as well as the greedy nature of the alternative optimization over $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$, we can only expect our coupled dictionary learning algorithm to find a local minimum, which turns out to be sufficient for practical use as demonstrated in the experimental part.

Algorithm 1 summarizes the complete procedures for our coupled dictionary learning algorithm:

Algorithm 1 Coupled Dictionary Training

1: **input:** training patch pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, and dictionary size K.
2: **initial:** initialize $\boldsymbol{D}_x^{(0)}$ and $\boldsymbol{D}_y^{(0)}$, $n = 0$, $t = 1$.
3: **repeat**
4: **for** $i = 1, 2, ..., N$ **do**
5: Compute gradient $\boldsymbol{a} = dL(\boldsymbol{D}_x^{(n)}, \boldsymbol{D}_y^{(n)}, \boldsymbol{x}_i, \boldsymbol{y}_i)/d\boldsymbol{D}_y$ according to Eqn. (12);
6: Update $\boldsymbol{D}_y^{(n)} = \boldsymbol{D}_y^{(n)} - \eta(t) \cdot \boldsymbol{a}$;
7: Project the columns of $\boldsymbol{D}_y^{(n)}$ onto the unit ball;
8: $t = t + 1$;
9: **end for**
10: Update $\boldsymbol{D}_y^{(n+1)} = \boldsymbol{D}_y^{(n)}$;
11: Update $\boldsymbol{D}_x^{(n+1)}$ according to Eqn. (11) with $\boldsymbol{D}_y^{(n+1)}$;
12: $n = n + 1$;
13: **until** convergence
14: **output:** coupled dictionaries $\boldsymbol{D}_x^{(n)}$ and $\boldsymbol{D}_y^{(n)}$.

1) Line 2: We can initialize $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ with various methods: a) we can train $\boldsymbol{D}_x$ from $\{\boldsymbol{x}_i\}_{i=1}^{N}$ using standard sparse coding, or define it mathematically, and initialize $\boldsymbol{D}_y$ with a random matrix; b) or we can initialize $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ with those trained from joint sparse coding.
2) Line 6: $\eta(t)$ is the step size for stochastic gradient descent, which shrinks in the rate of $1/t$.
3) Line 7: To satisfy the dictionary norm constraint, we normalize each column of $\boldsymbol{D}_y$ to unit ℓ^2-norm.

The proposed coupled learning algorithm is generic, and hence can be potentially applied to many signal recovery and computer vision tasks, *e.g.*, image compression, texture transfer, and super-resolution. In the following, we will focus on its application to patch-based single image super-resolution.

IV. Image Super-resolution via Sparse Recovery

In this section, we discuss single image super-resolution via sparse recovery, where the signals of LR image patches constitute an observation space $\mathcal{Y}$ and the signals of HR image patches constitute a latent space $\mathcal{X}$. We propose to model the mapping between the two spaces by coupled sparse dictionary learning, and use the learnt dictionaries to recover HR patch $\boldsymbol{x}$ for any given LR patch $\boldsymbol{y}$.

A. Coupled Dictionary Learning

Instead of directly using raw pixel values, we extract simple features from HR/LR patches respectively as the signals in their coupled spaces. The DC component is first removed from each HR/LR patch because the mean value of a patch is always preserved well through the mapping from HR space to LR space. Also, we extract gradient features from LR image patch as in Yang *et al.* [25], since the median frequency band in LR patch is believed to be more relevant to the missing high frequency information. Finally, all the HR/LR patch signals (extracted features) are normalized to unit length so that we do not need to worry about the shrinkage effect of ℓ^1-norm minimization on the sparse representations. As can be seen, the resultant HR/LR image patch (feature) pairs are tied by a mapping function far more complex than the linear system considered in most conventional inverse problems such as compressive sensing.

To train the coupled dictionaries, we sample a large number of training HR/LR image patch pairs from a external database containing clean HR images $\{\boldsymbol{X}_i\}_{i=1}^n$. For each HR image $\boldsymbol{X}_i$, we first blur and down-sample it to get a LR image, and then upscale this LR image by "bicubic" interpolation back to its original size to get the interpolated LR image $\boldsymbol{Y}_i$. From these image pairs $\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^n$, we sample N pairs of HR/LR patches of size $p \times p$, and extract their patch features using the aforementioned procedures to get training data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$. To avoid sampling too many smooth patches which are less informative, we eliminate those patches with small variances. Once the training data are prepared, the coupled dictionaries $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ are learnt as in Algorithm 1.

B. Patch-wise Sparse Recovery

In testing phase, we perform patch-wise sparse SR recovery with the learnt couple dictionaries. The input LR image $\boldsymbol{Y}$ is first interpolated to the size of desired HR image, and the interpolated LR image $\boldsymbol{Y}'$ is divided into a set of overlapping patches of size $p \times p$. For each LR image patch $\boldsymbol{y}_p$, we extract its feature $\boldsymbol{y}$ as in the training phase, and compute its sparse representation with respect to learnt dictionary $\boldsymbol{D}_y$. This sparse representation is then used to predict the underlying HR image patch $\boldsymbol{x}_p$ (feature $\boldsymbol{x}$) with respect to $\boldsymbol{D}_x$. The predicted HR patches are tiled together to reconstruct the HR image $\boldsymbol{X}$, where the average of multiple predictions is taken for each pixel in overlapping region as its final recovery. Algorithm 2 describes the sparse recovery SR method in detail.

Since the norm information of a signal is lost in its sparse recovery (typically $\|\boldsymbol{D}_x\boldsymbol{z}\|_2 < \|\boldsymbol{x}\|_2$), extra consideration is given when we recover the norm of HR image patches. From the unnormalized training patches, we find that the norm of a de-meaned HR image patch is approximately proportional to that of

Algorithm 2 Super-Resolution via Sparse Recovery

input: coupled dictionaries $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$, LR image $\boldsymbol{Y}$.

initialize: set HR image $\boldsymbol{X} = \boldsymbol{0}$; upscale $\boldsymbol{Y}$ to $\boldsymbol{Y}'$ by bicubic interpolation.

for each $p \times p$ patch $\boldsymbol{y}_p$ in $\boldsymbol{Y}'$ **do**

  $m = \text{mean}(\boldsymbol{y}_p)$, $r = \|\boldsymbol{y}_p - m\|_2$;

  Extract normalized gradient feature $\boldsymbol{y}$ for $\boldsymbol{y}_p$;

  $\boldsymbol{z} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{D}_y\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$;

  Recover HR patch feature: $\boldsymbol{x} = \boldsymbol{D}_x\boldsymbol{z}/\|\boldsymbol{D}_x\boldsymbol{z}\|_2$;

  Recover HR image patch:

$$\boldsymbol{x}_p = (c \times r) \cdot \boldsymbol{x} + m, \tag{19}$$

  where c is a constant;

  Add $\boldsymbol{x}_p$ to the corresponding pixels in $\boldsymbol{X}$;

end for

Average multiple predictions on overlapping pixels of $\boldsymbol{X}$;

output: HR image $\boldsymbol{X}$.

its de-meaned and interpolated LR image patch. The proportional factor c is a constant greater than 1[3], depending on the magnification scale, *e.g.*, $c = 1.2$ for magnification factor of 2 is found from linear regression. Therefore, we linearly scale $\boldsymbol{x}$ and add the mean value back in Eqn. (19) to recover the actual HR image patch $\boldsymbol{x}_p$.

V. Fast Image Super-resolution

The patch-wise sparse recovery approach can produce SR images of superior quality. However, the high computational cost associated with this approach has limited its practical use in real applications, which is the reason why most commercial photo editing softwares still prefer simple bicubic interpolation for image upscaling. In our approach, to produce one SR image of moderate size, we need to do processing on thousands of patches and each one involves a time-consuming ℓ^1-norm minimization problem. Therefore, reducing the number of patches to process and finding fast solver for ℓ^1-norm minimization problem are the two directions for efficiency improvement. In this section, we significantly speed up our algorithm in both directions without much compromise to SR performance: 1) selectively process LR image patches

[3]HR image patches have better contrast, and consequently have larger norms compared to their LR counterparts.

based on natural image statistics; and 2) learn a neural network model for fast sparse representation inference.

A. Selective Patch Processing

Natural images typically contain large smooth regions as well as strong discontinuities, such as edges and corners. Although simple interpolation methods for image upscaling, *e.g.*, bilinear and bicubic interpolation, will result in noticeable artifacts along the edges and corners, such as ringing, jaggies and blurring effects, they perform reasonably well on smooth regions. Figure 1 illustrates this fact, where we upscale ($\times 2$) each image patch of "Lena" by bicubic interpolation and sparse recovery respectively for RMSE comparison—"red" color denotes regions where our sparse recovery beats bicubic interpolation in terms of RMSE; "blue" color denotes regions where bicubic interpolation is superior; and "gray" regions means bicubic interpolation and sparse recovery are comparable. From this figure, we can conclude that 1) our sparse recovery algorithm performs overwhelmingly better than bicubic interpolation; 2) spare recovery performs much better than bicubic interpolation in edge and highly textured regions; and 3) sparse recovery and bicubic interpolation are comparable on large smooth regions. This observation suggests that we can selectively process those highly textured regions using our "expensive" sparse recovery technique and simply apply "cheap" bicubic interpolation for the rest smooth regions, which will save a remarkable amount of computation. In this work, we select the edge and textured regions by measuring the variance of image patch: if the variance of an image patch is larger than a threshold, we process it using sparse recovery; otherwise, we simply apply bicubic interpolation as our HR image patch estimation.

B. Fast Sparse Inference for Super-resolution

Instead of seeking the exact solution to the computationally expensive ℓ^1-norm minimization problem, we find approximated solution can also yield very accurate SR recovery. Several recent works have proposed to find the fast approximation of the sparse code by learning feed-forward neural network models [26], [32], [33]. Given examples of input vectors (LR image patch features) paired with their corresponding optimal sparse codes obtained by conventional optimization methods, the main idea of these neural network models is to learn an efficient parameterized non-linear encoder function to predict the optimal sparse codes. Although the predicted sparse codes from these neural network models have shown to be both efficient and effective for various recognition tasks, they have not been explored for

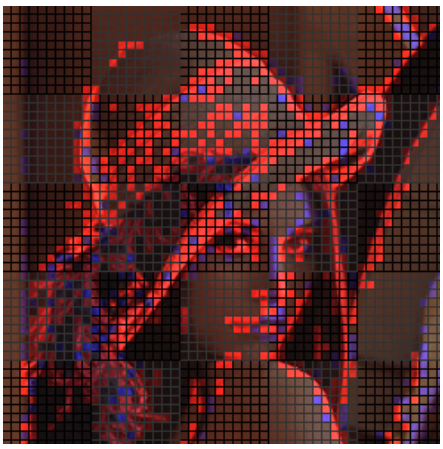


Fig. 1. Comparison of patch-wise SR reconstruction accuracy between sparse recovery and bicubic interpolation. "Red" color denotes the patches where sparse recovery beats bicubic interpolation; "Blue" color denotes patches where bicubic interpolation is superior; and "Gray" color indicates that the two perform on par with each other.

image restoration tasks. In this work, we employ the same network structure as in [26], while demanding the predicted sparse representations reconstruct the corresponding HR image patches well.

The particular form and parameterizations of the neural network encoder in [26] is inspired by the Iterative Shrinkage Thresholding Algorithm (ISTA) [34]. Given an input vector $\boldsymbol{y}$, the ISTA iterates the following recursive equation until convergence:

$$\boldsymbol{z}(k+1) = h_\theta(W_e\boldsymbol{y} + S\boldsymbol{z}(k)) \quad \boldsymbol{z}(0) = 0. \tag{20}$$

The elements of the above equation are defined as

$$\text{filter matrix}: \qquad W_e = \tfrac{1}{L}\boldsymbol{D}_y^T \tag{21}$$

$$\text{inhibition matrix}: \qquad S = I - \tfrac{1}{L}\boldsymbol{D}_y^T\boldsymbol{D}_y \tag{22}$$

$$\text{shrinkage function}: \quad h_\theta(\boldsymbol{z}) = sign(\boldsymbol{z})(|\boldsymbol{z}| - \theta)_+, \tag{23}$$

where L is a constant chosen to be larger than the largest eigenvalue of $\boldsymbol{D}_y^T\boldsymbol{D}_y$, function $h_\theta(\boldsymbol{z})$ is a component-wise shrinkage function with vector threshold θ with its all elements set to be λ/L. Depending on the overcompleteness of the dictionary, the ISTA may take tens of iterations to converge to the optimal solution, which is too slow for practical applications where typically thousands of such optimizations need to be solved. The basic idea of the algorithm in [26] is to employ a network structure which takes the form of Eqn. (20), with only a fixed number of iterations T (*e.g.*, $T = 2$). Instead of defining the equation elements in Eqn. (21) as in ISTA, the algorithm learns the parameters $W = (W_e, S, \theta)$ from

the training samples, which consists of input vectors and their paired optimal sparse codes, in order to minimize the divergence between the predicted sparse codes and the optimal sparse codes. Specifically, the neural network model is to minimize the following energy function:

$$\mathcal{L}(W) = \frac{1}{N}\sum_{i=1}^{N} C(W, \boldsymbol{y}_i), \tag{24}$$

with

$$C(W, \boldsymbol{y}_i) = \frac{1}{2}\|\boldsymbol{z}_i - f_e(W, \boldsymbol{y}_i)\|_2^2, \tag{25}$$

where $\boldsymbol{z}_i$ is the optimal sparse code of $\boldsymbol{y}_i$ in terms of $\boldsymbol{D}_y$, and $f_e(W, \boldsymbol{y}_i)$ is the transform defined by repeating Eqn. (20) T iterations with the learned parameter W. Tailored to our super-resolution algorithm, we can redefine the energy function as

$$C(W, \boldsymbol{y}_i, \boldsymbol{x}_i) = \frac{1}{2}\|\boldsymbol{z}_i - f_e(W, \boldsymbol{y}_i)\|_2^2 + \frac{\nu}{2}\|\boldsymbol{D}_x\boldsymbol{z}_i - \boldsymbol{x}_i\|_2^2 \tag{26}$$

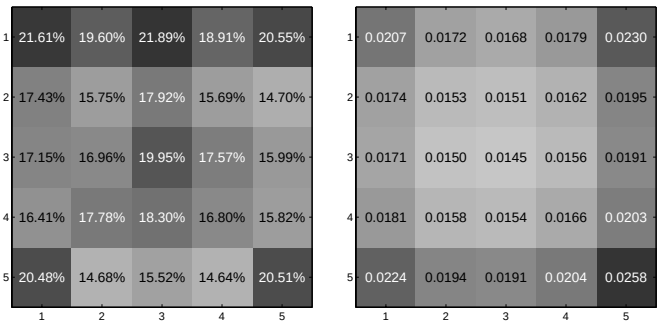
to achieve better predictions for HR image patches. As in conventional neural networks, the learning algorithm is again a simple stochastic gradient descent procedure, where the gradient can be derived from back-propagation by applying the simple chain rule.

VI. Experimental Evaluation

In this section, we apply our coupled dictionary learning method to single image super-resolution and also evaluate the efficiency of our implementation of the algorithm. For training, we sampled $100,000$ HR/LR 5×5 image patch (feature) pairs for magnification factor of 2, and learned the coupled dictionaries each of size $K = 512$ using stochastic gradient descent. With the learned coupled dictionary $\boldsymbol{D}_y$, we then train our neural network model for fast inference. All the experiments are performed on a PC running a single core of Intel Pentium 3.0GHz CPU. In the following, we first compare our results with the joint dictionary training method for sparse recovery in [8], which is one of the state-of-the-art super-resolution algorithms, both qualitatively and quantitatively. Then we discuss the computational efficiency of our algorithm, and make extensive comparisons in terms of visual quality between the results of our method and other recently proposed state-of-the-art approaches, *e.g.*, [22], [24].

A. Comparison with Joint Dictionary Training

We use the joint dictionary training approach by Yang *et al.* [8] as the baseline for comparison with our coupled training method. To ensure fair comparisons, we use the same training data for both methods, and employ exactly the same procedure to recover the HR image patches. Furthermore, to better manifest



(a) (b)

Fig. 2. (a). Percentages of pixel-wise MSE reduced by our coupled training method compared with joint dictionary training method. (b) Pixel-wise MSE of the recovered HR image patches (normalized and de-meaned) using our coupled dictionary training method.

the advantages of our coupled training, we use the same $\boldsymbol{D}_x$ learned by joint dictionary training as our pre-defined dictionary for HR image patches (which is usually not the optimal choice, since $\boldsymbol{D}_x$ can be updated along with the optimization of $\boldsymbol{D}_y$ as in Section III), and then optimize $\boldsymbol{D}_y$ to ensure good sparse recovery. The optimization converges very quickly, typically in less than 5 iterations.

1) Signal Recovery Accuracy: To validate the effectiveness of our coupled dictionary training, we first compare the recovery accuracy of both dictionary training methods on a validation set, which includes $100,000$ normalized image patch pairs sampled independently from the training set. Note that here we focus on evaluating the recovery accuracy for the de-meaned and normalized HR patch signals instead of the actual HR image pixels, thus isolating the affect of any application-specific technical details (*e.g.*, patch overlapping, contrast normalization, etc). Figure 2 (a) shows the pixel-wise mean square error (MSE) improvement using our coupled dictionary training method over the previous joint dictionary training method. It can be seen that our approach significantly reduces the recovery error in all pixel locations, which verifies the effectiveness of our training approach for sparse signal recovery.

2) Super-Resolution Performance: For patch-based super-resolution algorithms, the common practice is to recover each overlapping HR image patch independently, and then fuse the multiple pixel predictions in overlapping regions by simple averaging or other more sophisticated operation. Such a strategy is empirically supported by the error pattern observed in Figure 2 (b): large recovery errors are most likely to occur at the corner and boundary pixels in a patch. Therefore, even with only one pixel overlapping between adjacent patches, the inaccurate predictions or outliers are expected to be suppressed significantly. However, such an improvement in accuracy is obtained at the cost of computation time, which increases almost quadratically with the amount of overlapping. In Figure 4, the results of super-resolution by

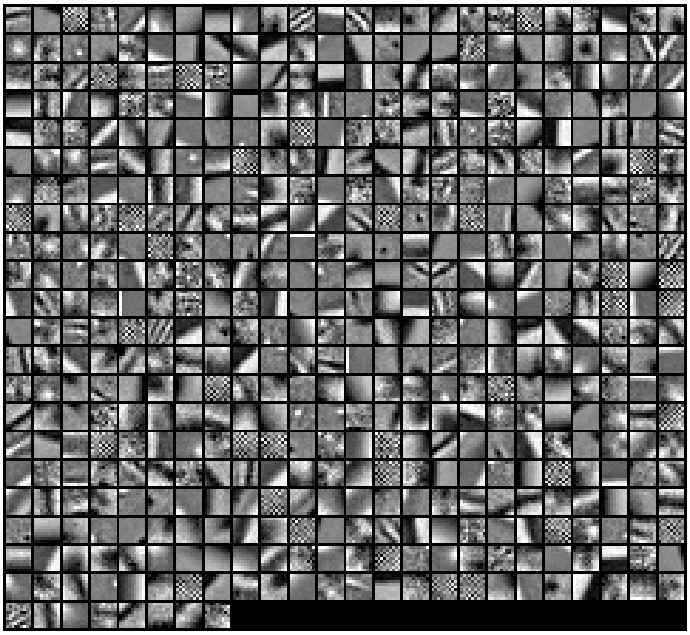


Fig. 3. The high resolution image patch dictionary trained.

magnification factor of 2 are compared on various real images between joint dictionary training and our coupled dictionary training with different amount of patch overlapping between adjacent patches. As shown, the results by our method are free of artifacts no matter how much overlapping is used; on the contrary, the artifacts introduced by joint dictionary training are always visible even up to 2-pixel overlapping. Actually, the artifacts of joint dictionary training will remain noticeable unless the overlapping increases to 3 pixels (note that the patch size is only 5×5).

Quantitatively, Figure 5 shows the changes of the recovery PSNRs on both "Lena" and "Flower" as we increase the overlapping pixels between adjacent patches. For reference, we also show the PSNRs of the bicubic interpolation for both images with horizontal dashed lines. Our method outperforms bicubic notably even with 0-pixel patch overlapping, and continues to improve as overlapping increases. The PSNRs for joint training are initially lower than bicubic, but increases substantially with more overlapping. One interesting observation is that our method does not benefit from pixel overlapping as much as joint dictionary training does; this is because our recovery is already close to the ground truth, and subsequently taking the average can not improve the accuracy too much. However, overlapping seems critical for the success of joint dictionary training for recovery. Another important observation is that the recovery using our training method with 0-pixel patch overlapping can achieve approximately the same level of

Fig. 4. Super-resolution results upscaled by factor 2, using joint dictionary training (top row) and our method (bottom row). 0/1/2-pixel overlapping between adjacent patches are used for the Flower/Lena/Street image, respectively.

performance as joint training with 3-pixel patch overlapping, with reduction in computation by more than 6 times. This advantage is crucial especially for real time applications and mobile devices.

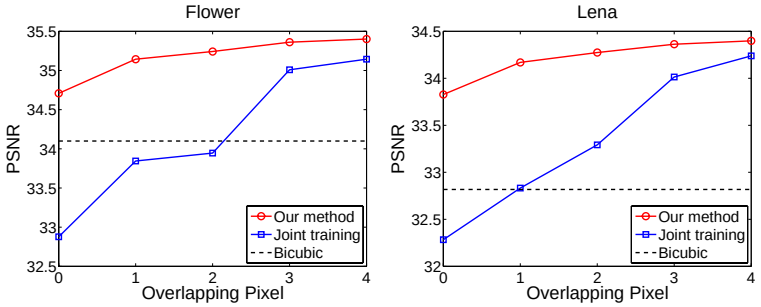


Fig. 5. The effects of pixel overlapping on PSNRs of the recovered images using different dictionary training methods. Our training method with 0-pixel overlapping can achieve the same level of performance as joint training with 3-pixel overlapping.

B. Algorithm Efficiency

In this subsection, we first evaluate our algorithm's efficiency with the two speeding up strategies proposed in Section V—selective patch processing and neural network for fast sparse inference—and then compare the visual qualities of our upscaled images with several recently proposed state-of-the-arts algorithms.

TABLE I

THRESHOLDING EFFECTS ON SR RECOVERY ACCURACY AND PROCESSING TIME WITH SPARSE CODES FOUND AS THE EXACT SOLUTION OF ℓ^1-NORM MINIMIZATION, TESTED ON THREE IMAGES WITH DIFFERENT SIZE.

Threshold on σ		0	10	20	30	40	50
Lena 128×128	RMSE	4.71	4.71	4.73	4.76	4.81	4.87
	Time(s)	62.0	42.4	28.9	20.8	17.2	13.5
Flower 106×99	RMSE	4.20	4.21	4.22	4.24	4.27	4.32
	Time(s)	39.2	29.9	20.4	15.8	11.5	8.9
Face 127×129	RMSE	4.32	4.33	4.36	4.42	4.46	4.50
	Time(s)	60.1	48.8	30.1	16.6	10.3	6.7

1) Selective Patch Processing: We apply thresholding on the standard deviation σ of LR input image patches to adaptively apply sparse recovery or bicubic interpolation: if the σ is larger than a pre-set threshold, we process the patch by sparse recovery; otherwise, we simply use bicubic interpolation. Table I shows the effects of the threshold on recovery accuracy and computation time for upscaling by factor of 2, where the sparse recovery is based on the exact sparse code solution of the optimization problem. For all the three images, the computation time drops significantly as the threshold increases, while the increase in recovery RMSEs is marginal (only $3\% \sim 4\%$), which is imperceivable from visual inspection. The results indicate that the test images contain a large portion of smooth regions that can be skipped in processing, which is a phenomena one can expect on most generic natural images.

2) Neural Network for Fast Sparse Inference: Table II shows the effects of the threshold on recovery accuracy and computation time for the neural network approximation. First, the same trends as in Table I are observed: when the threshold increases, the recovery RMSEs increase slowly, but the computation time drops remarkably. In addition, by collating the results in Table I and II, we can see the neural network model significantly reduces the computation time for processing the same number of patches (same threshold) compared with the original exact ℓ^1-norm minimization, only at the cost of marginal increase in RMSE, which again is hardly perceivable in the visual quality. In Figure 6, we show the visual quality of the zoomed "Face" images ($\times 2$) for both ℓ^1-norm minimization and neural network approximation with different thresholds on σ. The differences between ℓ^1-norm minimization and neural network approximation and between variant thresholds for both methods are visually unnoticeable, which validates our strategies of selective patch processing and neural network for fast sparse inference.

Table III shows the results on more test images which have also been used in [22], [24], and the results

TABLE II

THRESHOLDING EFFECTS ON SR RECOVERY ACCURACY AND PROCESSING TIME WITH SPARSE CODES FOUND VIA NEURAL NETWORK INFERENCE, TESTED ON THREE IMAGES WITH DIFFERENT SIZE.

Threshold on σ		0	10	20	30	40	50
Lena 128×128	RMSE	4.79	4.79	4.80	4.83	4.87	4.92
	Time(s)	6.5	5.2	4.4	3.8	3.5	3.3
Flower 106×99	RMSE	4.32	4.32	4.33	4.34	4.36	4.39
	Time(s)	4.1	3.5	2.9	2.5	2.3	2.1
Face 127×129	RMSE	4.38	4.39	4.42	4.46	4.49	4.52
	Time(s)	6.2	5.6	4.4	3.5	3.0	2.8

Fig. 6. Image quality comparisons for different standard deviation thresholds. Top: ℓ^1-norm minimization; bottom: neural network approximation. Thresholds from left to right: 0, 10, 20, 30.

are obtained using both speeding up strategies in our algorithm. These test images are magnified by factor of 4, which is achieved by upscaling the image twice with a magnification factor of 2 in each step. We set the variance threshold to be $20(30)$ and the patch overlap to be $3(2)$ pixels for the first(second) step. In Matlab, our algorithm can upscale an image of moderate size (200×200) in less than 30 seconds, and we expect it to be much faster with a dedicated C implementation. Compared to the exact ℓ^1-norm minimization without selective patch processing, our fast implementation is in general more than 20 times faster.

TABLE III

PROCESSING TIME OF SUPER-RESOLUTION BY MAGNIFICATION FACTOR OF 4 ON TEST IMAGES WITH DIFFERENT SIZES. RESULT IMAGES CAN BE ACCESSED AT HTTP://WWW.IFP.ILLINOIS.EDU/ JYANG29/IMAGES.HTM.

Image	Lena	Girl	Child	Chip	Wheel	Koala
Size	128×128	151×225	128×128	244×200	207×157	161×241
Time(s)	10.0	19.6	10.7	24.0	20.2	26.9
Image	Sculpture	Can	Street	Kitchen	Bird	-
Size	200×200	311×200	234×177	188×188	241×161	-
Time(s)	22.0	32.7	34.2	27.9	23.7	-

3) Visual Quality Comparisons with State-of-the-arts: In Figure 7, we compare the visual quality of our upscaled images with the results produced by several recent state-of-the-art methods. For the "Child" test image (first row, $\times 4$ magnification), the method of Glasner *et al.* [22] generates very sharp edges, but also a small amount of ringing and jaggies artifacts. Since their method aggressively enhances all edge-like structures, including the regions with smooth illumination transition (boundary between the face and the hat), the overall result does not look photorealistic. Freedman *et al.* [24] produces similar output as ours in this case, although occasionally some small "hard" and unnatural edges or corners are observed. For the "Statue" test image (second row, $\times 8$ magnification), both Freedman's and our results look much better than that of Sun *et al.* [19]. However, Freedman's approach generates again many "hard" edges that are not photorealistic, and also some false structures inconsistent with the observation (note the shape change between the upper lip and the moustache). For the "Wheel" test image (third row, $\times 4$ magnification), Freedman's algorithm can generate sharp edges, but loses the details that our approach can recover, *e.g.*, the texture details on the lower part of the dashboard behind the wheel. More results and comparisons can be found at http://www.ifp.illinois.edu/~jyang29/Images.htm.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we propose a novel coupled dictionary training method for single image super-resolution based on patch-wise sparse recovery. The coupled dictionary training enforces that the sparse representation derived from the low-resolution image patch in terms of the low-resolution dictionary can well reconstruct its high-resolution counterpart with the high-resolution patch dictionary. Compared with the previous joint dictionary training method, our algorithm improves the recovery accuracy notably, and at the same time removes the recovery artifacts. Furthermore, aiming at practical applications, we propose to adaptively process these image patches based on standard deviation thresholding and employ an neural

network model for fast approximate inference. Quantitatively and qualitatively, our fast implementation can achieve the same level of performance as the original exact algorithm, but at a much faster speed. For future work, we will investigate potential applications of our coupled dictionary training method in other recovery and vision problems, such as compressive sensing, texture transfer and intrinsic image estimation.

ACKNOWLEDGMENT

This work is supported by U.S. Army Research Laboratory and Army Research Office under grant number W911NF-09-1-0383. It is also supported in part by Adobe Systems, Inc.

REFERENCES

[1] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, pp. 613–627, 1995.

[2] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," in *Proceedings of IEEE*, 2010, vol. 98.

[3] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999, vol. 5, pp. 2443–2446.

[4] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Image Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[5] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, 2007, pp. 801–808.

[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *International Conference on Machine Learning*, 2009, pp. 689–696.

[7] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, 2008.

[8] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 1–8, 2010.

[9] A. Adler, Y. Hel-Or, and M. Elad, "A shrinkage learning approach for single image super-resolution with overcomplete representations," in *European Conference on Computer Vision*, 2010, pp. 622–635.

[10] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, 2006.

[11] J. Yang, K. Yu, and T. S. Huang, "Supervised translation-invariant sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3517–3524.

[12] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, (to appear).

[13] R. C. Hardie, K. J. Barnard, and E. A. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1621–1633, 1997.

[14] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super-resolution," *IEEE Transactions on Image Processing*, vol. 13, pp. 1327–1344, 2004.

[15] M. E. Tipping and C. M. Bishop, "Bayesian image super-resolution," in *Advances in Neural Information and Processing Systems 16*, 2003.

[16] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.

[17] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, pp. 56–65, 2002.

[18] H. Chang, D. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 275–282.

[19] J. Sun, Z. Xu, and H. Shum, "Image super-resolution using gradient profile prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[20] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the non-local means to super-resolution reconstruction," *IEEE Transactions on Image Processing*, pp. 36–51, 2009.

[21] G. Peyre, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," in *European Conference on Computer Vision*, 2008.

[22] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE International Conference on Computer Vision*, 2009, pp. 349–356.

[23] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang, "Non-local kernel regression for image and video restoration," in *European Conference on Computer Vision*, 2010.

[24] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–10, 2010.

[25] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[26] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, 2010, pp. 399–406.

[27] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals OR*, pp. 235–256, 2007.

[28] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annual of Statistics*, vol. 32, pp. 407–499, 2004.

[29] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, March 2010.

[30] D. Bradley and J. A. Bagnell, "Differentiable sparse coding," in *Advances in Neural Information Processing Systems*, 2008, pp. 113–120.

[31] H. Zou, T. Hastie, and R. Tibshirani, "On the "degree of freedom" of the lasso," *Annual of Statistics*, vol. 35, pp. 2173–2192, 2007.

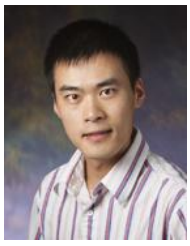
[32] K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "Fast inference in sparse coding algorithms with applications to object recognition," Tech. Rep., Computational and Biological Learning Lab, Courant Institute, NYU, 2008, Tech Report CBLL-TR-2008-12-01.

[33] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *IEEE International Conference on Computer Vision*, 2009, pp. 2146–2153.

[34] A. Beck and M. Teboulle, "A fast iterative shrinkage thresholding algorithm with application to wavelet based image deblurring," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 693–696.

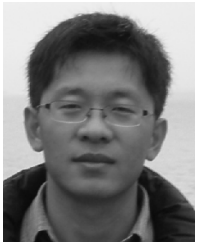
[35] Q. Shan, Z. Li, J. Jia, and C. Tang, “Fast image/video upsampling,” in *ACM Trans. Graph.*, 2008, vol. 27, pp. 153:1–153:7.

Jianchao Yang (S’08) received his B.E. degree in the Department of Electronics Engineering and Information Science, University of Science and Technology of China (USTC), China, in 2006; and his M.S. and Ph.D. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign, Illinois, in 2011. His research interests include computer vision, machine learning, sparse representation, compressive sensing, image and video processing.



Zhaowen Wang received the B.E. and M.S. degrees in electrical engineering from the Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2009, respectively. He is currently pursuing Ph.D. degree in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (UIUC), Urbana.

Since spring 2010, he has been with Department of Electrical and Computer Engineering, UIUC. His research interests include computer vision, statistical learning, and video events analysis.



Zhe Lin received the BEng degree in Automatic Control from the University of Science and Technology of China in 2002 and the MS degree in Electrical Engineering from the Korea Advanced Institute of Science and Technology in 2004, and the PhD degree in Electrical and Computer Engineering from the University of Maryland, College Park, in 2009. He has been a research intern at Microsoft Live Labs Research. He is currently working as a research scientist at the Advanced Technology Labs, Adobe Systems Incorporated, San Jose, California. His research interests include image and video restoration, object detection and recognition, content-based image and video retrieval. He is a member of the IEEE.

Scott Cohen received the B.S. degree in mathematics from Stanford University, Stanford, CA, in 1993, and the B.S., M.S., and Ph.D. degrees in computer science from Stanford University in 1993, 1996, and 1999, respectively.

During his PhD program, he did two summer internships in 1993 and 1994 at the Lawrence Livermore National Laboratory, Livermore, CA. He is currently a senior computer scientist in the Advanced Technology Labs of Adobe Systems, San Jose, CA. His research interests include interactive image and video segmentation, image and video matting, stereo, upsampling, and deblurring.

Thomas S. Huang (LF'01) received his B.S. Degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, China; and his M.S. and Sc.D. Degrees in Electrical Engineering from the Massachusetts Institute of Technology, Cambridge, Massachusetts. He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973; and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, and Research Professor at the Coordinated Science Laboratory, and at the Beckman Institute for Advanced Science he is Technology and Co-Chair of the Institute's major research theme Human Computer Intelligent Interaction.

Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 21 books, and over 600 papers in Network Theory, Digital Filtering, Image Processing, and Computer Vision. He is a Member of the National Academy of Engineering; a Member of the Academia Sinica, Republic of China; a Foreign Member of the Chinese Academies of Engineering and Sciences; and a Fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of America.

Among his many honors and awards: Honda Lifetime Achievement Award, IEEE Jack Kilby Signal Processing Medal, and the King-Sun Fu Prize of the International Association for Pattern Recognition.

Fig. 7. Super-resolution result comparisons. Top ($\times 4$): nearest neighbor, Glasner *et al.* [22], Freedman *et al.* [24], and ours. Middle ($\times 8$): nearest neighbor, Sun *et al.* [19], Freedman *et al.* [24], and ours. Bottom ($\times 4$): nearest neighbor, Shan *et al.* [35], Freedman *et al.* [24], and ours. Images courtesy of Glasner *et al.* [22], Freedman *et al.* [24], and Sun *et al.* [19].